



# Computation on Structures - The Challenge of Seamless Integration of Theory and Rigorous Scientific Practice

Klaus-Dieter Schewe\*

UIUC Institute, Zhejiang University, Haining, China

**Keywords:** theory of computation, behavioural theory, computation on structures, complexity theory, logic

In 1985 Egon Börger published his influential monograph on computation theory, logic, and complexity (Börger, 1989), which focused on the concept of *formal language* as a carrier of the precise expression of meaning, facts, and problems, and the concept of *algorithm* or calculus, i.e., a formally operating procedure for the solution of precisely described questions and problems. At that time, the text was at the forefront of a modern theory of these concepts, paving the way in which they developed first in mathematical logic and computability theory and later in automata theory, the theory of formal languages and complexity theory. Nonetheless, the state of the theory left many open problems. Computing started to stretch out into many new application areas. Distributed computing over networks became possible, database systems facilitated concurrent computation, artificial intelligence ventured from a niche area to a useful technology, enabling inferential problem solving in diagnosis, controlling machines through software became possible, etc.

Now, only 35 years later, the rapid progress in computing has led to a fascinating variety of interconnected systems that are used to support, manage, and control almost all aspects of our life. There is hardly an area that has not yet been penetrated by computing, and still there are many open challenges for the continuation of this success story. We are now dealing with systems of systems that are characterized by the following properties: 1) they operate in *parallel* either synchronously or asynchronously exploiting multiple processor cores and computing resources distributed over networks; 2) they are *hybrid* interacting with analogue systems with continuous behaviour; 3) they are *adaptive* changing their own behaviour; 4) they are *intelligent* reasoning about themselves and their environment and learning from data; 5) they are *interactive* communicating with their environment; and 6) they integrate *random* behaviour depending on probability distributions.

For instance, if we consider advanced manufacturing, all equipment will integrate electronic and mechanical parts with continuous behaviour, software with discrete behaviours, and sensors and actuators for their integration. Each piece of equipment constitutes an autonomous system of its own, and all the equipment of a shopfloor form a system of systems. This can be taken further to systems integrating all systems within a city. Any collection of systems that collaborate to achieve a common task can be considered a system of systems.

The key challenge for *theoretical computer science* is to provide the necessary scientific foundations centered around computation theory, complexity, and logic. It has to be ensured that modern computing, in practice with all its facets, is grounded in rigorous scientific methods. There is a huge gap between the abstraction level of an algorithmic system and the level of Turing machines. It is not sufficient to know that, deep inside the core of systems, we deal with computations that given a proper string encoding can be represented by Turing machines. Instead, we need a computation theory over arbitrary structures rather than strings. Structures in the sense of Tarski are omnipresent in all mathematical theories, and any extension should be conservative in the sense that the classical theory is preserved as a representation on the lowest level of abstraction.

## OPEN ACCESS

### Edited and reviewed by:

Kaleem Siddiqi,  
McGill University, Canada

### \*Correspondence:

Klaus-Dieter Schewe  
kdschewe@acm.org

### Specialty section:

This article was submitted to  
Theoretical Computer Science,  
a section of the journal  
Frontiers in Computer Science

**Received:** 22 February 2021

**Accepted:** 13 April 2021

**Published:** 07 May 2021

### Citation:

Schewe K-D (2021) Computation on  
Structures - The Challenge of  
Seamless Integration of Theory and  
Rigorous Scientific Practice.  
Front. Comput. Sci. 3:670602.  
doi: 10.3389/fcomp.2021.670602

- Is there a theory of computation that faithfully covers all the aspects of systems of systems that occur in practice?
- Is there a methodology grounded in such a theory of computation that permits the definition and classification of complex systems and the provision of means for specification, systematic development, validation, and verification?
- Is there a methodology that permits reasoning about problems and their solutions in terms of correctness and complexity?

A first answer was given in 1985 by Gurevich's "new thesis" (Gurevich, 1985), which was further elaborated in the 1995 Lipari guide (Gurevich, 1995). The new theory emphasizes universal algebras to capture abstract states of systems and evolving algebras (now known as ASMs) as the abstract machines capturing the algorithms on arbitrary levels of abstraction. These ideas do not only create a new paradigm for the foundations of computing subsuming the classical theory, but at the same time can be exploited for rigorous systems engineering in practice. A remarkable success story started: 1) the compiler correctness for the Warren Abstract Machine for Prolog was proven (Börger and Rosenzweig, 1995); 2) the correctness of the translation from Occam to transputers was shown (Börger and Durdanovic, 1996); 3) the compilation of Java and the proof of the correctness of the bytecode verifier was realized (Stärk et al., 2001); 4) a sophisticated and flexible theory of refinements was developed (Börger, 2003); and much more. The state of the theory and practice of ASMs is well summarized in Börger's and Stärk's monograph on ASMs (Börger and Stärk, 2003).

While these developments provided evidence that ASMs can take over the role of the formal languages in computation theory, it took until 2000 to develop the sequential ASM thesis (Gurevich, 2000): "if an abstraction level is fixed (...) and the states of an algorithm reflect all the relevant information, then a particular small instruction set suffices to model any algorithm, never mind how abstract, by a generalized machine very closely and faithfully". On the one hand the thesis provided a language-independent definition of the notion of *sequential algorithm* giving for the first time in history a precise axiomatic definition of the notion of "algorithm" (though restricted to sequential algorithms). On the other hand, it contained the proof that all algorithms as stipulated by the defining postulates are faithfully captured by sequential ASMs. This justified the establishment of another new notion: a *behavioural theory* comprises a machine-independent axiomatic definition of a class of algorithmic systems, an abstract machine model, and a proof that the machine model captures the class of computations. Further behavioural theories followed capturing parallel algorithms (Blass and Gurevich, 2003; Dershowitz and Falkovich-Derzhavetz, 2016; Ferrarotti et al., 2016), recursion (Börger and Schewe, 2020), reflection (Schewe and Ferrarotti, 2020a), concurrency, and distribution (Börger and Schewe, 2016), but these are only the first steps.

The behavioural theories yield variants of ASMs that can be used for rigorous systems development. Furthermore, logics for the reasoning about deterministic, non-deterministic and

reflective ASMs followed (Stärk and Nanchen, 2001; Ferrarotti et al., 2017; Ferrarotti et al., 2018; Schewe and Ferrarotti, 2020b). On the one hand this demonstrates the tight connections between the classes of algorithmic systems handled in the behavioural theories. On the other hand, this also shows that the development of the logical counterpart of the theories has not yet reached the same development state. This applies even more so to complexity theory. One of the few studies trying to bring complexity theory to the theory of computations on structures, is the theory of *choiceless polynomial time* (CPT) (Blass et al., 1999), which studies the choiceless fragment of PTIME. Though it was possible to show that CPT subsumes other models of computation on structures, it is strictly included in PTIME. If the hope had been to exhaust PTIME, this failed and Gurevich posited his conjecture that there is no logic capturing PTIME (Gurevich, 1988). If true, it would doom all further attempts in this direction. This would further imply that complexity theory as a whole could not be based on more abstract models of computations on structures, and in particular, it would not be possible to avoid dealing with string encodings using Turing Machines. However, this consequence appears to be less evident in view of the ASM success stories; only recently has a refutation of Gurevich's conjecture been achieved (Schewe, 2020).

In summary, what is needed is an enhanced theory of computation on structures comprising behavioural theories, logic, and complexity. The theory must provide the backbone for the systems of systems we deal with in practice in the sense that it reflects precisely on arbitrary levels of abstraction how systems in practice operate. Furthermore, the theory must guide the development of methods for systems engineering in practice as well as methods for the analysis of system behaviour.

Therefore, based on the challenges that I described briefly above my ambition for the development of theoretical computer science is driven by the vision of a complete scientific penetration of computing in practice, and the journal section should contribute to this. That is, I aim to obtain as much as possible an understanding of the challenges arising in practice. So, the goal is to have journal contributions that are grounded in the scientific abstraction process transferring practical problems into abstract problems for theoretical computer science. Then I aim to obtain solutions of these abstract problems including classification by expressiveness, complexity, feasibility, etc., so the journal section should contribute decent advanced computer science theory. Finally, I aim to see how the theoretical developments feed back into practice in the form of concrete languages and methodologies that can provably solve the problems that inspired the development of the theories.

The important first step, recruiting a board of Associate Editors sharing the overall vision, has been made. The next steps, launching special research topics on behavioural theory, logic, and complexity theory, will follow.

## AUTHOR CONTRIBUTIONS

The author confirms being the sole contributor of this work and has approved it for publication.

## REFERENCES

- Blass, A., and Gurevich, Y. (2003). Abstract State Machines Capture Parallel Algorithms. *ACM Trans. Comput. Logic* 4, 578–651. doi:10.1145/937555.937561
- Blass, A., Gurevich, Y., and Shelah, S. (1999). Choiceless Polynomial Time. *Ann. Pure Appl. Logic* 100, 141–187. doi:10.1016/s0168-0072(99)00005-6
- Börger, E., and Durdanovic, I. (1996). Correctness of Compiling Occam to Transputer Code. *Comput. J.* 39, 52–92. doi:10.1093/comjnl/39.1.52
- Börger, E. (1989). *Computability, Complexity, Logic, Studies in Logic and the Foundations of Mathematics*. Amsterdam: North-Holland.
- Börger, E. (2003). The ASM Refinement Method. *Form. Aspec. Comput.* 15, 237–257. doi:10.1007/s00165-003-0012-7
- Börger, E., and Stärk, R. (2003). *Abstract State Machines*. Berlin Heidelberg New York: Springer-Verlag.
- Börger, E., and Rosenzweig, D. (1995). A Mathematical Definition of Full Prolog. *Sci. Comput. Programm.* 24, 249–286. doi:10.1016/0167-6423(95)00006-E
- Börger, E., and Schewe, K.-D. (2020). A Behavioural Theory of Recursive Algorithms. *Fundamenta Informati.* 177, 1–37. doi:10.3233/fi-2020-1978
- Börger, E., and Schewe, K.-D. (2016). Concurrent Abstract State Machines. *Acta Informatica* 53, 469–492. doi:10.1007/s00236-015-0249-7
- Dershowitz, N., and Falkovich-Derzhavetz, E. (2016). On the Parallel Computation Thesis. *Logic J. IGPL* 24, 346–374. doi:10.1093/jigpal/jzw008
- Ferrarotti, F., Schewe, K.-D., Tec, L., and Wang, Q. (2016). A New Thesis Concerning Synchronised Parallel Computing - Simplified Parallel ASM Thesis. *Theor. Comput. Sci.* 649, 25–53. doi:10.1016/j.tcs.2016.08.013
- Ferrarotti, F., Schewe, K.-D., Tec, L., and Wang, Q. (2017). A Complete Logic for Database Abstract State Machines. *Logic J. IGPL* 25, 700–740. doi:10.1093/jigpal/jzx021
- Ferrarotti, F., Schewe, K.-D., Tec, L., and Wang, Q. (2018). A Unifying Logic for Non-deterministic, Parallel and Concurrent Abstract State Machines. *Ann. Math. Artif. Intell.* 83, 321–349. doi:10.1007/s10472-017-9569-3
- Gurevich, Y. (1985). American Mathematical Society. *A New Thesis (Abstract)*. 6, 317.
- Gurevich, Y. (1988). “Logic and the Challenge of Computer Science,” in *Current Trends in Theoretical Computer Science* Editor E. Börger (Ann Arbor, Michigan: Computer Science Press), 1–57.
- Gurevich, Y. (1995). “Evolving Algebras 1993: Lipari Guide,” in *Specification and Validation Methods* Editor E. Börger (Oxford, United Kingdom: Oxford University Press), 9–36.
- Gurevich, Y. (2000). Sequential Abstract-State Machines Capture Sequential Algorithms. *ACM Trans. Comput. Logic* 1, 77–111. doi:10.1145/343369.343384
- Schewe, K. D. (2020). Insignificant Choice Polynomial Time Preprint: arxiv:2005.04598. 69.
- Schewe, K. D., and Ferrarotti, F. (2020a). Behavioural Theory of Reflective Algorithms I: Reflective Sequential Algorithms Preprint: arxiv:2001.01873. 32.
- Schewe, K.-D., and Ferrarotti, F. (2020b). “A Logic for Reflective ASMs,” in *Rigorous State-Based Methods - 7th International Conference (ABZ 2020)* Lecture Notes in Computer Science. Editors A. Raschke, D. Méry, and F. Houdek, (Ulm, Germany: Springer). doi:10.1007/978-3-030-48077-6\_7
- Stärk, R., and Nanchen, S. (2001). A Logic for Abstract State Machines. *J. Universal Comput. Sci.* 7, 217–231. doi:10.3217/jucs-007-11-0980
- Stärk, R. F., Schmid, J., and Börger, E. (2001). *Java and the Java Virtual Machine: Definition, Verification, Validation*. Berlin, Germany: Springer. doi:10.1007/978-3-642-59495-3 CrossRef Full Text

**Conflict of Interest:** The author declares that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.

Copyright © 2021 Schewe. This is an open-access article distributed under the terms of the Creative Commons Attribution License (CC BY). The use, distribution or reproduction in other forums is permitted, provided the original author(s) and the copyright owner(s) are credited and that the original publication in this journal is cited, in accordance with accepted academic practice. No use, distribution or reproduction is permitted which does not comply with these terms.