

1. Record Nr.	UNINA9910143631003321
Titolo	Algebraic Methodology and Software Technology : 8th International Conference, AMAST 2000 Iowa City, Iowa, USA, May 20-27, 2000 Proceedings // edited by Teodor Rus
Pubbl/distr/stampa	Berlin, Heidelberg : , : Springer Berlin Heidelberg : , : Imprint : Springer, , 2000
ISBN	3-540-45499-3
Edizione	[1st ed. 2000.]
Descrizione fisica	1 online resource (XI, 545 p.)
Collana	Lecture Notes in Computer Science, , 0302-9743 ; ; 1816
Disciplina	005.1/01/512
Soggetti	Computers Data structures (Computer science) Algebra Computer logic Software engineering Logic, Symbolic and mathematical Theory of Computation Data Structures and Information Theory Logics and Meanings of Programs Software Engineering Mathematical Logic and Formal Languages
Lingua di pubblicazione	Inglese
Formato	Materiale a stampa
Livello bibliografico	Monografia
Note generali	Bibliographic Level Mode of Issuance: Monograph
Nota di bibliografia	Includes bibliographical references at the end of each chapters and index.
Nota di contenuto	Education Day -- Invited Talk: A Software Engineering Program of Lasting Value (Abstract) -- Invited Talk: Weaving Formal Methods into the Undergraduate Computer Science Curriculum (Extended Abstract) -- Technical Meetings -- Invited Talk: Making Mathematical Methods More Practical for Software Developers (Abstract) -- Step by Step to Histories -- Distance Functions for Defaults in Reactive Systems -- Generalizing the Modal and Temporal Logic of Linear Time -- Process Algebra versus Axiomatic Specification of a Real-Time Protocol -- Practical Application of Functional and Relational Methods for the Specification and Verification of Safety Critical Software -- Algebraic

State Machines -- Meta Languages in Algebraic Compilers -- Random Access to Abstract Data Types -- A Monad for Basic Java Semantics -- A Global Semantics for Views -- Analysis of Downward Closed Properties of Logic Programs -- Invited Talk: ASM Formalware in the Software Engineering Cycle -- Process Calculi for Coordination: From Linda to JavaSpaces -- The Algebra of Multi-tasking -- A Causal Semantics for Timed Default Concurrent Constraint Programming -- Casl-Chart: A Combination of Statecharts and of the Algebraic Specification Language Casl -- Message Authentication through Non Interference -- Plugging Data Constructs into Paradigm-Specific Languages: Towards an Application to UML -- An ASM Semantics for UML Activity Diagrams -- Approximate Bisimilarity -- Time and Probability in Process Algebra -- A Modal Logic for Klaim -- Kleene under a Demonic Star -- Pointwise Relational Programming -- Towards a Toolkit for Actor System Specification -- Maude Action Tool: Using Reflection to Map Action Semantics to Rewriting Logic -- The Extensibility of Maude's Module Algebra -- A Reuse-Oriented Framework for Hierarchical Specifications -- MIX(FL): A Kernel Language of Mixin Modules -- Behavioural Subtyping Relations for Object-Oriented Formalisms -- Applying Category Theory to Derive Engineering Software from Encoded Knowledge -- A New Logic for Electronic Commerce Protocols -- Extended Institutions for Testing -- Testing from Structured Algebraic Specifications.

Sommario/riassunto

The AMAST movement was initiated in 1989 with the First International Conference on Algebraic Methodology and Software Technology (AMAST), held on May 21-23 in Iowa City, Iowa, and aimed at setting the development of software technology on a mathematical basis. The virtue of the software technology envisioned by AMAST is the capability to produce software that has the following properties: (a) it is correct and its correctness can be proved mathematically, (b) it is safe, such that it can be used in the implementation of critical systems, (c) it is portable, i. e. , it is independent of computing platforms and language generations, and (d) it is evolutionary, i. e. , it is self-adaptable and evolves with the problem domain. Ten years later a myriad of workshops, conferences, and research programs that share the goals of the AMAST movement have occurred. This can be taken as proof that the AMAST vision is right. However, often the myriad of workshops, conferences, and research programs lack the clear objectives and the coordination of their goals towards the software technology envisioned by AMAST. This can be taken as a proof that AMAST is still necessary.
