ELSEVIER
Theoretical Computer Science 291 (2003) 219–221
Theoretical Computer Science
www.elsevier.com/locate/tcs

# Preface

The AMAST movement was initiated in 1989 with the First International Conference on Algebraic Methodology and Software Technology held in Iowa City, Iowa, and aimed at putting software development technology on a firm, mathematical foundation based on algebraic and logical methods. Devising and refining algebraic and logical methodologies for software development remains the major objective of the AMAST movement. The ultimate goal is to make such methodologies both viable and attractive for common software engineering practice.

During the years, AMAST has attracted an international following among researchers and practitioners interested in software technology, programming methodology and their algebraic and logical foundations. At the same time, there has been a proliferation of workshops, conferences, and initiatives at both the industrial and the academic level which share AMAST's main goals of applying mathematical methods to software development. We see this as a clear testimony of the validity and vitality of AMAST's original vision.

AMAST 2000, the 8th International Conference of the AMAST series, was held again in Iowa City, Iowa, from May 20 through May 27, 2000. It was intended as an anniversary meeting to celebrate the achievements of the first decade of the AMAST movement, examine current trends in the use of formal methods for software development, and discuss the opportunity of adapting AMAST's goals to the challenges raised by new developments in software technology.

This special issue of Theoretical Computer Science contains revised versions of selected papers from AMAST 2000. The works collected here were chosen for their quality, level of original contribution, and consistency with the overall AMAST goals. Each selected paper underwent a thorough revision process carried out by the author(s) and was reviewed by at least three referees selected worldwide among experts in the field. Only papers with positive reviews were accepted for final publication. Here follows a brief description of the accepted papers.

1. *A New Logic for Electronic Commerce Protocols* by K. Adi, M. Debbabi and M. Mejri describes a modal logic in the dynamic logic family that is particularly well-suited for specifying and proving properties of protocols for electronic commerce. These include traditional security properties such as authentication, secrecy and integrity, and also e-commerce specific properties, such as non-repudiation, anonymity, money atomicity, certified delivery, and so on. The main features of the logic are the presence of modalities, its linearity, and its ability to formalize recursive specifications. The logic's linearity is of particular interest for the specification of e-commerce properties, as it allows one to model resource consumption.


PII: S0304-3975(02)00363-8

The logic constructs are interpreted over a trace-based model. Traces reflect valid protocol executions in the presence of a malicious smart intruder. The logic is endowed with a sound and complete tableau-based inference system for verifying whether a given protocol trace satisfies a formula.

2. *A comparison of three authentication properties* by R. Focardi, R. Gorrieri and F. Martinelli formalizes and illustrates a number of connections between three notions of authenticity: Abadi and Gordon's *spi-authentication* property, Focardi and Gorrieri's *NDC* property, and Lowe's *agreement* property. Process algebraic approaches have been used to specify authentication properties in a variety of frameworks. The paper establishes a unified approach to compare such properties, based of the idea of non-interference. This is done within the framework of *Cryptographic Security Process Algebra* (*CrytpoSPA*), a process algebra in the style of value-passing CCS but with special facilities for handling crypto-mechanisms on messages. The authors show that, under mild assumptions, spi-authentication can be recast into CrytpoSPA and proved equivalent to NDC. From this they then show that Lowe's agreement is a stronger property than spi-authentication.
3. *Coalgebras and monads in the semantics of Java* by B. Jacobs and E. Poll describes the basic structures in the denotational and axiomatic semantics of a conventional programming language—sequential Java—considered in its entirety, "warts and all." The authors investigate the semantics of sequential Java from both a monadic and a coalgebraic perspective. Java statements and expressions have different termination options: normal termination (yielding a successor state and possibly a result value), non-termination, and abrupt termination (caused by exceptions or by statements such as `continue` or `break`). This makes the formalization of a denotational semantics for Java particularly challenging. The monadic view presented in the paper organizes and describes in a mathematically clean way the complications arising in defining the semantics of Java's major constructs (composition, extension, and repetition). The coalgebraic view yields an associated program logic with proper definitions of invariance, bisimulation and modalities. The modal operators can be used to define axiomatic semantics for Java that take the various termination options into account.
4. *Meta Languages in Algebraic Compilers* by E. Van Wyk contributes to the field of compiler design. The paper proposes an approach to dissociate the target language from the meta language used to describe the translation process. The operations provided by a given target language may not be expressive enough to correctly specify the translation from the source language; alternatively, they may be at such a low level of abstraction that the specification is excessively difficult to read and write. This paper illustrates how different specification languages can be used in conjunction with a target language to specify translators without extending the target language. As an example of this approach, and of its generality, the paper shows how model checking can be characterized as a language translation problem. The paper defines a model checker for Computation Tree Logic as an algebraic compiler mapping the CLT language into a target language of satisfiability sets. The operations in the chosen target language are not powerful enough to specify general computations. A couple of alternative specification languages are then described for

providing a more computationally expressive environment in which to specify the translation.

We take this opportunity to thank all the authors who submitted their work to AMAST 2000 and commend them on their efforts in carrying forward the goals of the AMAST movement. We are grateful to the anonymous referees of this issue for their great job and sharp reviews. Finally, we send our special thanks to Maurice Nivat, founding Editor of this journal and co-founder of AMAST, for making this special issue possible and for his invaluable contribution in promoting AMAST's ideas within the field of theoretical computer science.

*Guest Editors*
Cesare Tinelli
Teodor Rus

*Department of Computer Science*
*University of Iowa*
*Iowa City, IA 52242*
USA

tinelli@cs.uiowa.edu (C. Tinelli)
rus@cs.uiowa.edu (T. Rus)