Jose Meseguer

Formal Methods and Delarative Languages Laboratory
Department of Computer Science
University of Illinois, Urbana-Champaign

TITLE:
Flexible Formal Methods for High Assurance: The Rewriting Logic/Maude Experience

ABSTRACT:
High assurance is a comprehensive goal that should not be sought using a single method in isolation, or focusing on a single level of system abstraction. We need a flexible wide spectrum of formal methods that can be used in tandem and can be applied to both designs and code at different phases in a a system's lifecycle. This talk will describe our experience using the Maude formal specification language and its support for a wide range of such formal methods. The Maude project is a joint effort at SRI International, UIUC, and the Universities of Madrid, Malaga, and Oslo. The Maude system and its tool environment and documentation are freely available on the web (at http://maude.cs.uiuc.edu) and are widely used all over the world.

Maude is based on rewriting logic, a very simple computational logic easy to understand by engineers and very well suited to specify distributed systems. This logic is executable in Maude with very high performance. This means that it is relatively easy to: (i) formally specify a system in Maude, and (ii) symbolically simulate such a system by executing its Maude specification. This way we can debug the specifications themselves, easily explore alternative designs, and analyze system behaviors in different scenarios before a system is built. Since distributed systems are notoriously hard to get right, because of their nondeterminism, Maude also offers the possibility of (iii) exhaustively exploring all system behaviors from an initial state to find violations of safety properties. This is accomplished by a breadth first search command that provides a semidecision procedure for finding such failures. An even stronger analysis method is (iv) model checking of temporal logic properties, supported by Maude's built-in LTL model checker. This requires the set of reachable states to be finite, and provides then a decision procedure for any temporal logic property. Even when a system is infinite-state, it is often possible to model check a finite-state abstraction, thus verifying the desired properties for the original system. Using abstractions requires discharging some proof obligations; also, for some systems an abstraction may not be available. In such cases, (v) inductive theorem proving support is needed, which in Maude is provided by its ITP Theorem Prover. Not all methods (i)-(v) need to be used for all systems or at all levels of abstraction: a judicious combination of methods will best fit each particular application. In general, it is most cost-effective to first apply lighter methods to get many bugs out, reserving heavier methods for later as needed.

The talk will describe our experience applying Maude in several areas and at different levels of system abstraction, including: network security, cryptographic protocols, active networks, wireless communication, real-time systems, Maude-based formal tool development, and formal analysis of concurrent code in languages such as Java and the JVM.

◆ABSTRACTS OF TALKS

Munehiro Iwami

Department of Mathematics and Computer Science
Shimane University

TITLE:
Persistence of Termination for Overlay Term Rewriting Systems

ABSTRACT:
A property P is called persistent if for any many-sorted term rewriting system R, R has the property P if and only if term rewriting system $¥Theta (R)$, which results from R by omitting its sort information, has the property P. In this talk, we show that termination is persistent for locally confluent overlay term rewriting systems and we give the example as application of this result. Furthermore we show that termination is persistent for right-linear overlay term rewriting systems.

****

Hitoshi Ohsaki

Research Center for Verification and Semantics (CVS),
National Institute of Advanced Industrial Science and Technology (AIST)

TITLE:
Reachability Analysis Based on AC-tree Automata Technique

ABSTRACT:
In this talk a verification tool called ACTAS, e.g for security protocols, based on AC-tree automata theory is introduced. ACTAS is an integrated system for manipulating associative and commutative tree automata (AC-tree automata for short), that has various functions such as for Boolean operations of AC-tree automata, computing rewrite descendants, and solving emptiness and membership problems. In order to deal with high-complexity problems in reasonable time, over-and under-approximation algorithms are also equipped.

Such functionality enables us automated verification of safety property in infinite state models, that is helpful in the domain of, e.g. network security, in particular, for security problems of cryptographic protocols allowing an equational property. In runtime of model construction, a tool support for analysis of state space expansion is provided. The intermediate status of the computation is displayed in numerical data table, and also the line graphs are generated. Besides, a graphical user interface of the system provides us a user-friendly environment for handy use.

****

◆ABSTRACTS OF TALKS

Bruno Buchberger

Professor of Computer Mathematics Research Institute for Symbolic Computation
Johannes Kepler University, A4232 Castle of Hagenberg, Austria

TITLE:
The Theorema Project: An Overview

ABSTRACT:
The Theorema project aims at supporting, by algorithms, the process of mathematical theory exploration. We study all phases of this process from concept invention to proposition invention and proof, problem invention, and algorithm invention and verification. The final goal is the flexible build-up of well structured, formal, verified mathematical knowledge bases including algorithm libraries and a tool box for working with such knowledge bases. The project is based on previous research experience in the area of computer algebra, notably the theory of Groebner bases.

In the talk I will describe the basic ideas and the current state of the project. A particular emphasis will be laid on recent results on algorithm-supported algorithm synthesis. I will also give some examples of the currently available tools in the Theorema system.

****

Makoto Takeyama

Research Center for Verification and Semantics (CVS),
National Institute of Advaced Industrial Science and Technology (AIST)

TITLE:
An Integrated Verification Environment

ABSTRACT:
We outline our plan for an integrated developing envrionment for systems verification. Its aim is to integrate variety of verification methods and tools into a coherent whole: high-level model description, abstraction, model-checking, and interactive theorem proving. We explain aspects of the methodology supported by such an environment.

****

Masahiko Sato

Graduate School of Informatics, Kyoto University

TITLE:
A Simple Theory of Expressions, Judgments and Derivations

◆ABSTRACTS OF TALKS

ABSTRACT:

We propose a simple theory of expressions which is intended to be used as a foundational syntactic structure for the Natural Framework (NF). We define expression formally and give a simple proof of the decidability of $¥alpha$-equivalence. We use this new theory of expressions

to define judgments and derivations formally, and we give concrete examples of derivation games to show a flavor of NF.

****

Yoshinori Tanabe

Japan Science and Technology Agency (JST)
Research Center for Verification and Semantics (CVS),
National Institute of Advaced Industrial Science and Technology (AIST)

TITLE:

Satisfiability checking of temporal logics for verification of pointer systems

ABSTRACT:

Hagiya and Takahashi has proposed a method of abstraction for graph rewriting systems, which uses temporal logics and is suitable for describing properties of pointer systems. We have started designing a verification tool based on the method. Satisfiability checking plays an important role for verification using abstraction and therefore should be a key part in our tool. In this talk algorithms for deciding satisfiability in variants of CTL, their implementation with BDD, and relationship to pointer systems are presented.

****