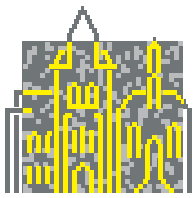
D. Kapur (Univ. of New Mexico, US), A. Podelski (MPI Saarbrücken, D), A. Voronkov (Univ. of Manchester, GB)
(Editors)

Deduction and Infinite-state Model Checking

Dagstuhl Seminar 03171 – April 21 to April 25, 2003
Dagstuhl-Seminar-Report No. 376



SCHLOSS DAGSTUHL

INTERNATIONALES
BEGEGNUNGS-
UND FORSCHUNGSZENTRUM
FÜR INFORMATIK

ISSN 0940-1121

Herausgegeben von IBFI gem. GmbH, Schloss Dagstuhl, 66687 Wadern, Germany.

Das Internationale Begegnungs- und Forschungszentrum für Informatik (IBFI) Schloss Dagstuhl ist eine gemeinnützige GmbH. Sie veranstaltet regelmäßig wissenschaftliche Seminare, welche nach Antrag der Tagungsleiter und Begutachtung durch das wissenschaftliche Direktorium mit persönlich eingeladenen Gästen durchgeführt werden.

Gesellschafter:

- Gesellschaft für Informatik e.V. – Bonn
- TH Darmstadt
- Universität Frankfurt
- Universität Kaiserslautern
- Universität Karlsruhe
- Universität Stuttgart
- Universität Trier
- Universität des Saarlandes

Motivation

Q: In 'infinite model checking', what is infinite, the model or the checking?

A: Both.

Model checking is an automated method to verify runtime properties of programs. Finite model checking applies to finite abstractions of software systems. Often, the task of constructing appropriate finite abstractions manually is hard, if not impossible. Therefore, a recent and promising research direction aims at infinite model checking. Here, deduction takes the central role in accounting for the infiniteness that arises from the direct modelling of software systems.

So far, the deduction problems arising in this context have been addressed in an adhoc manner by the model checking community. It is interesting to explore where existing techniques can be applied and where new kinds of research questions are raised.

For finite systems, model checking is based on Boolean logic. For many of the classes of systems with specific characteristics for infinite data and infinite control, the question for the right logic is still open (right in terms appropriate expressiveness and computational cost). It will be useful to classify the deduction problems corresponding to the different classes of systems.

Data: What classes of formulas are best used to account for classes of operations over classical domains such as integers? What are the new domains to model pointer structures, lists, queues, abstract data types in general?

Control: Advanced control(recursion, concurrency, threads, dynamic objects with changing communication patterns, mobility of computational agents,) requires models of process terms with specific algebraic laws (for stack concatenation, parallel composition,); which ones exactly?

For safety properties, model checking amounts to automatically synthesising inductive invariants, by fixpoint iteration. For infinite model checking, the application of the fixpoint operator, the fixpoint test and the extrapolation of intermediate results each are theorem proving tasks. What are the demands, the functionality, and the evaluation criteria for theorem provers that are called during fixpoint iteration?

For example, the performance of a possibly incomplete decision tool for the validity of implication (used for the fixpoint termination test) determines a tradeoff where the fixpoint iteration terminates after either few but possibly expensive steps or cheap but possibly numerous steps.

Extrapolation of intermediate results during fixpoint iteration is required for accelerating or enforcing termination. The abstract interpretation framework of Cousot and Cousot formulates abstraction techniques at a semantic level. Their instantiation to syntax-based theorem provers is still not obvious.

There are many more possible topics to be discussed at our workshop...

Participants

- Armando, Alessandro (University of Genova)
- Baader, Franz (TU Dresden)
- Baumgartner, Peter (MPI für Informatik – Saarbrücken)
- Benzmüller, Christoph (Universität des Saarlandes)
- Bibel, Wolfgang (TU Darmstadt)
- Boigelot, Bernard (University of Liège)
- Bonacina, Maria Paola (Università degli Studi di Verona)
- Bouajjani, Ahmed (University Paris-Diderot)
- Bultan, Tevfik (University of California – Santa Barbara)
- de Nivelle, Hans (MPI für Informatik – Saarbrücken)
- Degtyarev, Anatoli (King's College – London)
- Felty, Amy (University of Ottawa)
- Furbach, Ulrich (Universität Koblenz-Landau)
- Ganzinger, Harald (-1)
- Giese, Martin (Chalmers UT – Göteborg)
- Giesl, Jürgen (RWTH Aachen)
- Hillenbrand, Thomas (MPI für Informatik – Saarbrücken)
- Kapur, Deepak (University of New Mexico – Albuquerque)
- Kazakov, Yevgeny (MPI für Informatik – Saarbrücken)
- Korovin, Konstantin (University of Manchester)
- Legay, Axel (University of Liège)
- Leitsch, Alexander (TU Wien)
- Letz, Reinhold (TU München)
- McAllester, David (Toyota Technological Institute – Chicago)
- McCune, William (Argonne National Laboratory)
- Middeldorp, Aart (University of Tsukuba)
- Minea, Marius (Polytechnical University – Timisoara)
- Nieuwenhuis, Robert (UPC – Barcelona)
- Plaisted, David A. (University of North Carolina – Chapel Hill)
- Podelski, Andreas (MPI für Informatik – Saarbrücken)
- Ranise, Silvio (INRIA Lorraine)
- Raskin, Jean-Francois (Université Libre de Bruxelles)

– Richardson, Julian (NASA / RIACS – Moffett Field)
– Rubio, Albert (UPC – Barcelona)
– Rusinowitch, Michaël (INRIA Lorraine)
– Schmidt, Renate (University of Manchester)
– Schmidt-Schauss, Manfred (Universität Frankfurt)
– Schnoebelen, Philippe (ENS – Cachan)
– Schulz, Stephan (TU München)
– Seshia, Sanjit A. (Carnegie Mellon University – Pittsburgh)
– Slaney, John (Australian National University – Canberra)
– Stenz, Gernot (TU München)
– Stickel, Mark (SRI – Menlo Park)
– Subramaniam, Mahadevan (University of Nebraska)
– Thomas, Wolfgang (RWTH Aachen)
– Tiwari, Ashish (SRI – Menlo Park)
– Veith, Helmut (TU München)
– Veroff, Robert (University of New Mexico – Albuquerque)
– Voronkov, Andrei (University of Manchester)