Report from Dagstuhl Seminar 11272

Decision Procedures in Soft, Hard and Bio-ware (Follow Up)

Edited by

Nikolaj Bjørner[1], Robert Nieuwenhuis[2], Helmut Veith[3], and Andrei Voronkov[4]

1 **Microsoft Research - Redmond, US,** nBjørner@microsoft.com
2 **UPC - Barcelona, ES,** roberto@lsi.upc.edu
3 **TU Wien, AT,** veith@forsyte.tuwien.ac.at
4 **University of Manchester, GB,** voronkov@cs.man.ac.uk

Abstract

This report documents the program and the outcomes of Dagstuhl Seminar 11272 *Decision Procedures in Soft, Hard and Bio-ware (Follow Up)*. It was held as a follow-on for a seminar 10161, of the same title, that took place in late April 2010 during the initial eruption of Eyjafjallajökull. In spite of the travel disruptions caused by the eruption of the volcano, the original seminar received a respectable turnout by European, mainly German and Italian participants. Unfortunately, the eruption hindered participation from overseas or even more distant parts of Europe. This caused the seminar to cover only part of the original objective. The follow-on seminar focused on the remaining objectives, in particular to *bio-ware* and *constraint solving methods.*

Seminar 03.–06. July, 2011 – www.dagstuhl.de/11272
1998 ACM Subject Classification B.1.4. Microprogram Design Aids/Verification, D.2.4. Software/Program Verification, F.4.1. Mechanical theorem proving, F.4.3. Decision Problems, J.3. Biology and genetics
Keywords and phrases Hardware and Software Verification, Bio-analysis, Satisfiability Modulo Theories, Dynamic Symbolic Execution, Interpolants
Digital Object Identifier 10.4230/DagRep.1.7.23

1 Executive Summary

Nikolaj Bjørner

The main goal of the seminar *Decision Procedures in Soft, Hard and Bio-ware (Follow Up)* was to bring together renowned as well as young aspiring researchers from two groups. The first group formed by researchers who develop both theory and efficient implementations of decision procedures. The second group comprising of researchers from application areas such as program analysis and testing, crypto-analysis, hardware verification, industrial planning and scheduling, and bio-informatics, who have worked with, and contributed to, high quality decision procedures. The purpose of the seminar was to heighten awareness between tool and theory developers for decision procedures with the array of applications found in software, hardware and biological systems analysis.

The seminar fell on two and a half days in the week of July 4–6, 2011. 25 researchers from 12 countries (Germany, Austria, Italy, France, USA, United Kingdom, China, Hungary, Spain, Sweden, Czech Republic, Ireland) participated.

2 Table of Contents

3 Overview of Talks

3.1 Combined First-Order and Separation Logic Reasoning

Joshua Berdine (Microsoft Research UK – Cambridge, GB)

We describe techniques for combining first-order and separation logic reasoning used in the SLAyer verification tool. SLAyer uses separation logic to reason about memory safety properties of low-level heap-manipulating code. The Z3 SMT solver is used internally in a number of ways:

- to discharge queries that fall into the first-order fragment of separation logic;
- to reason about equality between pointer expressions, using unSat core extraction to guide sequent calculus proof search for separation logic queries; and
- to direct sequent calculus case splits by unSAT cores.

These uses employ a first-order approximation of separation logic formulas that is linear in the size of the formula and constrains the variables as strongly as the separation logic formula, but makes weaker constraints on the heap.

3.2 Open Constraint Logic Programming with SMT

Nikolaj Bjørner (Microsoft Research – Redmond, US)

Joint work of Bjørner, Nikolaj; Ethan Jackson; Wolfram Schulte
URL http://research.microsoft.com/projects/formula

I will present work in progress on open constraint logic programming using the SMT solver Z3. Similar to Datalog satisfiability, open constraint logic programming solves satisfiability of constraint programs: the input is a constraint logic program and a query, the output is a set of pairs comprising of satisfying instances to queries and additional facts that are required to satisfy the query.

The Microsoft Research FORMULA system implements open constraint logic programming and uses it for model based design.

I will describe the abstract machine that combines forward chaining and SMT solving in FORMULA and the accompanying type system that is important to constrain how additional facts can be used.

3.3 Computational Problems in Biology: Introduction and Challenges

Christoph Flamm (Universität Wien, AT)

Main reference Jakob L. Andersen, Christoph Flamm, Daniel Merkle, Peter F. Stadler, "Maximizing Output and Recognizing Autocatalysis in Chemical Reaction Networks is NP-Complete," submitted to J Sys Chem 2011.
URL http://arxiv.org/abs/1110.6051

In my presentation I will give a brief overview of computational problems in Biology with a special focus on metabolic networks. The formalization of a chemical reaction network as a stoichiometric matrix allows to derive the important concept of elementary pathways. These pathways form a convex basis which spans the space of all feasible mass flux distributions through the reaction network under steady state conditions. These flux distributions can not be measured directly but must be inferred computationally from isotope labeling experiments. The NP-hard problem of finding the chemically correct atom to atom mapping between the reaction partners in the network constitutes the core problem of all flux reconstruction algorithm. In the main part of the talk I will present our current research on a graph grammar based approach for chemical transformations which allows for an explicitly construction of the "chemical space" over a set of chemical graphs and a set of graph rewrite rules (reactions). I will clarify the notion of chemical transformation motif an will explain how chemical transformation motifs can be found in arbitrary chemical reaction networks using an integer linear programming approach. I will close my talk by posing the unsolved inverse reaction mechanism problem which seems interesting for the verification community. The challenge of the inverse reaction mechanism problem is to find a suitable set of molecules which "implement" a given abstract reaction mechanism using only chemical transformations from a pre-specified input reactions set.

3.4 Solvers for Theories of Strings

Vijay Ganesh (MIT – Cambridge, US)

Many automatic testing, analysis, and verification techniques for programs can be effectively reduced to a constraint-generation phase followed by a constraint-solving phase. This separation of concerns often leads to more effective and maintainable tools. The increasing efficiency of off-the-shelf constraint solvers makes this approach even more compelling.

However, there are few, if any, effective and sufficiently expressive off-the-shelf solvers for string constraints generated by analysis techniques for string manipulating programs. In order to fulfill this need we designed and implemented Hampi, a solver for string constraints over bounded string variables.

Hampi constraints express membership in regular languages and bounded context-free languages. Hampi constraints may contain context-free- language definitions, regular-language definitions and operations, the membership predicate and equations over string terms (word equations). String terms are constructed out of string constants, variables, concatenation and extraction functions.

Given a set of constraints, Hampi outputs a string that satisfies all the constraints, or reports that the constraints are unsatisfiable. Hampi is expressive and efficient, and can be successfully applied to testing and analysis of real programs. Our experiments use Hampi in: static and dynamic analyses for finding SQL injection vulnerabilities in Web applications; automated bug finding in C programs using systematic testing; and compare Hampi with another string solver.

3.5 Robust Formulas over Reals and Delta-Complete Decision Procedures

Sicun Gao (Carnegie Mellon University – Pittsburgh, US)

I will present a framework for the reliable use of numerically-driven procedures for deciding nonlinear SMT problems over reals.

I will first show decidability and reasonably low complexity of decision problems in the robust fragment of SMT, which are formulas whose satisfiability remains invariant under controllable numerical perturbations, in a very rich first-order theory over reals. I will then propose the notion of delta-complete decision procedures to capture the ideal behavior of numerically-driven procedures, which should decide robust formulas correctly and also return informative answers on non-robust formulas. I argue that delta-complete decision procedures, apart from scalability, can be more suitable than the usual precise procedures for some verification problems such as bounded model checking and invariant checking of hybrid systems.

3.6 On Quantifier-free Interpolation for Arrays

Silvio Ghilardi (Università di Milano, IT)

Joint work of Bruttomesso, Roberto; Ghilardi, Silvio; Ranise, Silvio
Main reference R. Bruttomesso, S. Ghilardi, S. Ranise, "Rewriting-based Quantifier-free Interpolation for a Theory of Arrays," Proc. 22nd International Conference on Rewriting Techniques and Applications (RTA'11), pp. 171–186, LIPIcs, Vol. 10.
URL http://dx.doi.org/10.4230/LIPIcs.RTA.2011.171

The use of interpolants in model checking [4] is becoming an enabling technology to allow fast and robust verification of hardware and software. The application of encodings based on the theory of arrays, however, is limited by the impossibility of deriving quantifier-free interpolants in general [5]. In this contribution, we first show that, with a minor extension to the theory of arrays, it is possible to obtain quantifier-free interpolants [3],[1]. We prove this by designing an interpolating procedure, based on solving equations between array updates. Rewriting techniques are used in the key steps of the solver and its proof of correctness.

Arrays are usually combined with fragments of arithmetic over indexes in applications, especially those related to software verification. For example, it is known that being able to handle integer indexes with constant increment or decrement operations is important when verifying a large class of programs with loops. As a further contribution [2], we combine the above quantifier-free interpolation solver for our variant of the theory of arrays with integer difference logic over indexes.

References

1 R. Bruttomesso, S. Ghilardi, and S. Ranise. Rewriting-based Quantifier-free Interpolation for a Theory of Arrays. Technical Report RI 334-10, Dip. Scienze dell'Informazione, Univ. di Milano, 2010.

2 R. Bruttomesso, S. Ghilardi, and S. Ranise. A Combination of Rewriting and Constraint Solving for the Quantifier-free Interpolation of Arrays with Integer Difference Constraints. In *FroCoS*, 2011.

3 R. Bruttomesso, S. Ghilardi, and S. Ranise. Rewriting-based Quantifier-free Interpolation for a Theory of Arrays. In *RTA*, LIPIcs, Vol. 10, 2011.

4 T. Henzinger and K. L. McMillan R. Jhala, R. Majumdar. Abstractions from Proofs. In *POPL*, 2004.

5 D. Kapur, R. Majumdar, and C. Zarba. Interpolation for Data Structures. In *SIGSOFT'06/FSE-14*, pages 105–116, 2006.

3.7 μZ, Fixed Point Engine in Z3

Krystof Hoder (University of Manchester, GB)

Joint work of Hoder, Krystof; Nikolaj Bjørner; Leonardo de Moura
Main reference Krystof Hoder, Nikolaj Bjørner, Leonardo de Moura, "μZ – An Efficient Engine for Fixed Points with Constraints," CAV 2011: 457–462.
URL http://dx.doi.org/10.1007/978-3-642-22110-1_36

The μZ tool is a scalable, efficient engine for fixed points with constraints.

It supports high-level declarative fixed point constraints over a combination of built-in and plugin domains. The built-in domains include formulas presented to the SMT solver Z3 and domains known from abstract interpretation. We present the interface to μZ, a number of the domains, and a set of examples illustrating the use of μZ.

3.8 Modular Theorem Proving

Christopher Lynch (Clarkson University – Potsdam, US)

Joint work of Lynch, Christopher; Ralph Eric McGregor

We show how to combine theorem proving techniques. A set of first order clauses to determine satisfiability is divided into two sets S and T, not necessarily disjoint. There are two theorem provers I and J. The theorem provers must be sound and refutationally complete, and I must be able to produce an (over-approximation of) a model. I is run on S, and J is run on T. If I determines that S is satisfiable, then I passes a candidate model M to J. If J determines that M union J is unsatisfiable then J passes back a learned clause witnessing the unsatisfiability to I. The process is repeated until I reports unsatisfiability or J had nothing new to learn. The process is sound and refutationally complete.

This is an abstract results, which can be instantiated with different theorem provers, possibly more than two. SMT is an instance of this, where S contains propositional clauses, T represents a theory, I is DPLL and J is a theory solver for T. However, our results allow for full first order and overlaps between the theories. We also show how Resolution/Superposition can construct an over-approximation of a model, and be used as the theorem prover I.

3.9 Haplotype Inference with Boolean Optimization

João Marques-Silva (University College – Dublin, IE)

Joint work of Marques-Silva, João; Lynce, Ines ; Graca, Ana ; Morgado, Antonio; Oliveira, Arlindo

Motivated by the success of Boolean Satisfiability (SAT) solvers, there has been recent work on solving combinatorial optimization problems in Bioinformatics with SAT and SMT-based solutions. This talk overviews the successful use of SAT-based approaches in solving a concrete combinatorial optimization problem in Bioinformatics, namely haplotype inference.

The talk will details the models used the haplotype inference problem, and also overviews the algorithms used for implementing SAT and SMT- based optimization.

3.10 Computing the Size of the Solution Space

Feifei Ma (Chinese Academy of Sciences, CN)

Joint work of Ma, Feifei; Liu, Sheng; Zhang, Jian
URL http://dx.doi.org/10.1007/978-3-642-02959-2_33

Most constraint solvers and decision procedures try to decide whether a given set of formulas (constraints) are satisfiable, and try to find a solution in case they are indeed satisfiable. In this talk, we discuss a different but related class of problems, i.e., how to compute the number of solutions or the size of the solution space. Such a problem can be regarded as the counting version of the decision problem. We describe the motivation for this work, a prototype tool for solving a special version of the problem (i.e., SMT instances on linear arithmetic), and application of the technique and tool to program analysis. In addition, we are also investigating optimization problems that are constrained by SMT formulas. We use similar techniques for computing the solution space size and for solving the generalized optimization problem. (The details are available in a technical report.)

References

1 Feifei Ma, Sheng Liu and Jian Zhang, Volume Computation for Boolean Combination of Linear Arithmetic Constraints. CADE 2009: 453-468.
2 Feifei Ma, Jun Yan and Jian Zhang, Solving Generalized Optimization Prolems Subject to SMT Constraints. Technical Report, ISCAS-SKLCS-11-12, 2011.

3.11 Using Bounded Model Checking to Focus Fixpoint Iteration

David Monniaux (Verimag, FR)

Joint work of Monniaux, David and Gonnord, Laure
Main reference David Monniaux und Laure Gonnord, "Using Bounded Model Checking to Focus Fixpoint Iterations", SAS 2011, pp. 369–385, LNCS Vol. 6887, Springer-Verlag.
URL http://dx.doi.org/10.1007/978-3-642-23702-7_27

Two classical sources of imprecision in static analysis by abstract interpretation are widening and merge operations. Merge operations can be done away by distinguishing paths, as in trace partitioning, at the expense of enumerating an exponential number of paths. In this talk, we describe how to avoid such systematic exploration by focusing on a single path at a time, designated by SMT-solving. Our method combines well with acceleration techniques, thus doing away with widenings as well in some cases. We illustrate it over the well-known domain of convex polyhedra.

3.12 SAT Modulo Theories and Scheduling Applications

Robert Nieuwenhuis (UPC – Barcelona, ES)

Here we first give an overview of SMT, the DPLL(T) approach to SMT (Nieuwenhuis et al, JACM, November 2006), and its implementation in our Barcelogic SMT tool.

Then we discuss current work on the development of SMT technology for hard combinatorial (optimization) problems outside the usual verification applications. The aim is to obtain the best of several worlds, combining the advantages inherited from SAT: efficiency, robustness and automation (no need for "tuning") and CP features such as rich modeling languages, special-purpose filtering algorithms (for, e.g., planning, scheduling or timetabling constraints), and sophisticated optimization techniques. We give several examples and discuss the impact of aspects such as first-fail heuristics vs. activity-based ones, realistic structured problems vs. random or handcrafted ones, and lemma learning.

3.13 SAT/SMT Techniques for Scheduling Problems with Sequence-Dependent Setup Times

Albert Oliveras (TU of Catalonia – Barcelona, ES)

The well-known success of SAT/SMT techniques in verification applications has motivated researchers to also focus on other application areas. In this talk, we will focus on scheduling problems with sequence-dependent setup times, that is, problems where a certain time is required to prepare the necessary resources to perform a given task. Since setup times are sequence-dependent, they depend both on the current task and the one immediately preceding it.

We will present two versions of this problem depending on the function to minimize: (i) makespan and (ii) earliness plus tardiness. In both cases, will explain how we can build upon SAT/SMT technology.

3.14 SAT Modulo Non-Linear Integer Arithmetic and Linear Invariant Generation

Albert Rubio (UPC – Barcelona, ES)

Joint work of Borralleras, Cristina; Larraz, Daniel; Lucas, Salvador; Oliveras, Albert; Rodríguez-Carbonell, Enric; Rubio, Albert

Polynomial constraint solving plays a prominent role in several areas of hardware and software analysis and verification. In this talk we propose a new method for solving non-linear constraints over the integers based on encoding the problem into an SMT problem considering only linear arithmetic. Unlike other existing methods, our method focuses on proving satisfiability of the constraints rather than on proving unsatisfiability, which is more relevant in several applications. In particular, we show how our solver can be used inside the so-called constraint-based invariant generation approach, first described in Colon et al. 2003, to obtain linear invariants of imperative programs automatically. Implementation issues are described and future extension addressed.

The talk is based on the work described in [1] and [2].

References

1 Cristina Borralleras Salvador Lucas Albert Oliveras Enric Rodríguez-Carbonell Albert Rubio. *SAT Modulo Linear Arithmetic for Solving Polynomial Constraints*. Journal of Automated Reasoning, 2011. DOI 10.1007/s10817-010-9196-8.

2 Daniel Larraz. *Automatic Generation of Loop Invariants*. Master Thesis, 2011. Universitat Politecnica de Catalunya.

3.15 Decidability and complexity for the verification of safety properties of reasonable linear hybrid automata

Viorica Sofronie-Stokkermans (MPI für Informatik - Saarbrücken, DE)

Joint work of Sofronie-Stokkermans, Viorica; Damm, Werner; Ihlemann, Carsten

We identify an industrially relevant class of linear hybrid automata (LHA) called reasonable LHA for which parametric verification of convex safety properties with exhaustive entry states can be verified in polynomial time and time-bounded reachability can be decided in nondeterministic polynomial time for non-parametric verification and in exponential time for parametric verification.

Properties with exhaustive entry states are restricted to runs originating in a (specified) inner envelope of some mode invariant.

Deciding whether an LHA is reasonable is shown to be decidable in polynomial time.

The results are presented in a paper published in the proceedings of HSCC 2011.

3.16 Solving Systems of Linear Inequalities by Bound Propagation.

Andrei Voronkov (University of Manchester, GB)

Joint work of Voronkov, Andrei; Korovin, Konstantin

In this talk we introduce a new method for solving systems of linear inequalities. The algorithm incorporates many state-of-the-art techniques from DPLL-style reasoning.

We prove soundness, completeness and termination of the method.

3.17 An Efficient Decision Procedure for Imperative Tree Data Structures

Thomas Wies (IST Austria – Klosterneuburg, AT)

Joint work of Wies, Thomas; Muñiz, Marco; Kuncak, Viktor
Main reference Thomas Wies, Marco Muñiz, Viktor Kuncak, "An Efficient Decision Procedure for Imperative Tree Data Structures," CADE 2011: 476–491
URL http://dx.doi.org/10.1007/978-3-642-22438-6_36

We present a new decidable logic called TREX for expressing constraints about imperative tree data structures. In particular, TREX supports a transitive closure operator that can express reachability constraints, which often appear in data structure invariants. We show that our logic is closed under weakest precondition computation, which enables its use for automated software verification. We further show that satisfiability of formulas in TREX is decidable in NP. The low complexity makes it an attractive alternative to more expensive logics such as monadic second-order logic (MSOL) over trees, which have been traditionally used for reasoning about tree data structures.

3.18 Lazy Decomposition for Distributed Decision Procedures

Christoph Wintersteiger (Microsoft Research UK – Cambridge, GB)

Joint work of Hamadi, Youssef; Marques-Silva, Joao; Wintersteiger, Christoph
Main reference Hamadi, Marques-Silva, Wintersteiger, "Lazy Decomposition for Distributed Decision Procedures," Workshop on Parallel and Distributed Methods in Verification, 2011.
URL http://dx.doi.org/10.4204/EPTCS.72.5

The increasing popularity of automated tools for software and hardware verification puts ever increasing demands on the underlying decision procedures. In this seminar talk, we present a framework for distributed decision procedures (for first-order problems) based on Craig interpolation.

Formulas are distributed in a lazy fashion, i.e., without the use of costly decomposition algorithms. Potential models which are shown to be incorrect are reconciled through the use of Craig interpolants. Experimental results on challenging propositional satisfiability problems indicate that our method is able to outperform traditional solving techniques even without the use of additional resources.

4 Working Groups

4.1 An SMT format for Strings, Sequences and Regular Languages

We arranged a discussion session around the topic of creating an interchange format for logical formulas using strings, regular expressions and grammars. This is increasingly relevant as decision procedures are being developed and used for analyzing string-manipulating programs. There are several applications. One important application area is for sanitizer programs that remove potentially malicious content from strings so that they can be safely used when performing data-base queries or used as parameters to Java-script code run inside a browser. We formed a working group on strings led by Vijay Ganesh and Nikolaj Bjørner. The discussion forum strings-smtization@googlegroups.com, which now has a few dozen subscribers. We summarize the objectives below.

4.2 Objectives

The objective is for a design for an SMT-LIB2 format for strings, regular expressions and context free grammars. The aim is to develop a set of core operations capturing the capabilities of main string solvers and the needs of main applications that use string constraints.

Strings can be viewed as an instance of the theory of monoids (sequences) where the main operations are creating the empty string, the singleton string and concatentation of strings. Unification algorithms for this theory has been subject to extensive theoretical advances over several decades. In contrast modern programming environments support libraries that contain a large set of string operations. Applications arising from programming analysis tools use the additional vocabulary available in libraries. A realistic interchange format should therefore support operations that are encountered in applications.

Note that SMT-LIB distinguishes between theories, which define sort and function symbols and their semantics, and logics which define the fragment of the language of one or more theories (see the reference document or the tutorial at http://www.smt-lib.org/) that one wants to work with.

4.3 Working Group Participants (from the seminar and afterwards)

Nikolaj Bjørner, David Cok, Vijay Ganesh, Tim Hinrichs, Pieter Hooimeijer, Ruzica Piskac, Prateek Saxena, Cesare Tinelli, Margus Veanes, Andrei Voronkov and Ting Zhang.

Participants

- Joshua Berdine
Microsoft Res. UK – Cambridge, GB
- Nikolaj Bjørner
Microsoft Res. – Redmond, US
- Christoph Flamm
Universität Wien, AT
- Vijay Ganesh
MIT – Cambridge, US
- Sicun Gao
Carnegie Mellon University – Pittsburgh, US
- Silvio Ghilardi
Università di Milano, IT
- Krystof Hoder
University of Manchester, GB
- Deepak Kapur
University of New Mexico – Albuquerque, US
- Laura Kovacs
TU Wien, AT
- Christopher Lynch
Clarkson Univ. – Potsdam, US
- Feifei Ma
Chinese Academy of Sciences, CN
- Joao Marques-Silva
University College – Dublin, IE
- David Monniaux
VERIMAG – Gières, FR
- Robert Nieuwenhuis
UPC – Barcelona, ES
- Albert Oliveras
TU of Catalonia – Barcelona, ES
- Ruzica Piskac
EPFL – Lausanne, CH
- Enric Rodriguez-Carbonell
UPC – Barcelona, ES
- Albert Rubio
UPC – Barcelona, ES
- Viorica Sofronie-Stokkermans
MPI für Informatik – Saarbrücken, DE
- Helmut Veith
TU Wien, AT
- Andrei Voronkov
University of Manchester, GB
- Thomas Wies
IST Austria – Klosterneuburg, AT
- Christoph Wintersteiger
Microsoft Research UK – Cambridge, GB
- Jian Zhang
Chinese Academy of Sciences, CN
- Ting Zhang
Iowa State Univ. – Ames, US