Satisfiability Checking and Symbolic Computation (SC²)

E. Abraham¹, J. Abbott¹, B. Becker², A.M. Bigatti³, M. Brain⁴, B. Buchberger⁴, A. Cinaglia⁵, J.H. Davenport⁶, M. England⁶, P. Fontaine⁷, S. Forrest⁸, A. Griggio⁹, D. Kroening⁹, W.M. Selen₄ and T. Sturm⁴

¹RWTH Aachen University, Aachen, Germany; ²Albert-Ludwigs-Universität, Freiburg, Germany;
³Università degli studi di Genova, Italy; ⁴Johannes Kepler Universität, Linz, Austria; ⁵Fondazione Bruno Kessler, Trento, Italy;
⁶University of Bath, Bath, U.K.; ⁷Coventry University, Coventry, U.K.; ⁸LORIA, Inria, Université de Lorraine, Nancy, France;
⁹Maplesoft Europe Ltd. ¹⁰University of Oxford, Oxford, U.K.; ¹¹Universität Kassel, Kassel, Germany;
¹²CNRS, LORIA, Nancy, France and Max-Planck-Institut für Informatik, Saarbrücken, Germany.

Introduction

We describe a new project to bring together the communities of Symbolic Computation and Satisfiability Checking into a new joint community, SC², supported by a newly accepted EU project (H2020-FETOPEN-CSA 712689) of the same name. Both communities have long histories, as illustrated by the tool development timeline below, but there is currently little interaction. However, they now have the common interest of the development of tools. An early highlight was Schoenauer’s decision procedures for arithmetic theories. By working together the communities can resolve problems, academic and industrial, currently beyond the scope of either individually.

This poster gives introductions to the two separate communities, discusses some of the challenges for the new SC² community, and the project actions planned for addressing them. The reader is referred to [2] for more details and full references; and the SC² website [4] for new information on the project as it occurs, or to get involved.

Symbolic Computation and Computer Algebra Systems

The use of computers to do algebra rather than simply arithmetic is almost as old as computing itself. Initial work consisted of programs to do one thing, but this soon led to systems capable of a variety of tasks. Many examples exist in the development of Maple. An early highlight was Mösses’s algorithm for symbolic integration paired with Risch’s completeness theorem to prove un-integrability. Initially, SAT solvers can be seen as a multivariate non-linear generalization of both Euclid’s algorithm and Gaussian elimination, allowing the effective and in many cases efficient solution of problems involving polynomial ideals and their associated algebraic varieties. The main commercial general-purpose computer algebra systems (including MAGMA, Maple, Mathematica) all have independent implementations, and there are specialised (freely downloadable) systems such as Singular, Macaulay and CoCoA.

Satisfiability Checking

SAT: the problem of checking the satisfiability of logical statements over the Booleans. Notable work includes that of Davis, Putnam, Logemann and Loveland who used resolution for quantifier elimination, and a combination of enumeration and Boolean constraint propagation (BCP). Another major improvement was the conflict-driven clause-learning and non-chronological backtracking of Marques-Silva and Sakallah. While the SAT Problem is known to be NP-complete, SAT solvers have been developed which can handle inputs with millions of Boolean variables, and are used in many industrial applications, e.g. in verification and security.

Driven by this success, big efforts were made to enrich propositional SAT-solving for different extensions of the Boolean case: first order logic with uninterpreted functions, array theory, bit-vector arithmetic and quantifier-free linear real and integer arithmetic. See [3] for an introduction which discusses the highlights so far. However, the development for quantifier-free non-linear real and integer arithmetic is still in its infancy and progress is required for applications in the automotive and avionic industries [7]. We hereby note that the typical functioning of SMT solvers (as illustrated in Figure 2) is to check the consistency of constraints. To be SAT-solver-compliant the solvers should:

- work incrementally, i.e. accept additional constraints and recheck making use of previous results;
- support backtracking, i.e. the removal of previously added constraints;
- in case of unsatisfiability return an explanation, e.g. a small inconsistent subset of constraints.

Project Challenges and Opportunities

SAT solving has its strength in efficient techniques for exploring Boolean structures, learning, combining techniques, and developing dedicated heuristics. Symbolic Computation provides powerful procedures for systems of arithmetic constraints, and has expertise in simplification and preprocessing.

To allow further exploitation by the Satisfiability Checking community, Symbolic Computation tools must first be adapted to comply with SAT requirements as set out in the previous section. Cylindrical Algebraic Decomposition, Gröbner Bases and Virtual Substitution are algorithms of particular interest. However, this is a challenge that requires the expertise of computer algebra developers.

Conversely, Symbolic Computation could profit from exploiting successful SMT ideas, like dedicated data structures, dynamic heuristics, effective learning techniques, and approaches for constraint propagation and branch-and-bound. Even fewer SMT solvers are available for non-linear integer arithmetic.

The community is supported by conferences (e.g. CADE, ICAR, SMT) and journals (e.g. J. Automated Reasoning), while a role somewhat analogous to SIGSAM is played by the SatLive Forum.

References

[4] C. Barrett. Despite the common language, and being able to share challenging problems is an essential aspect for building a dynamic community.

Your Invitation

The project consists of not just the partner institutions but also associates from both EU and non-EU research institutions and industry. Associates will be regularly informed about project activities and invited to corresponding events. If you would like to participate please contact the Project Coordinator James Davenport (J.H.Davenport@bath.ac.uk).

Figure 2: The typical functioning of SAT solvers

Solvers that are able to cope with linear arithmetic problems include Alt-Ergo, CVC4, ISAT3, MathSAT, OpenSMT, SAT-RAT, verifly, Yices2, and Z3. Far fewer tools exist for non-linear arithmetic: ISAT3 uses interval constraint propagation; Yices3 tries to reduce problems to linear real arithmetic; Z3 uses an adaptation of the CAD solver modules for SAT, with additional constraint propagation and branch-and-bound. Even fewer SMT solvers are available for non-linear integer arithmetic.

The community is supported by conferences (e.g. CADE, ICAR, SMT) and journals (e.g. J. Automated Reasoning), while a role somewhat analogous to SIGSAM is played by the SatLive Forum.

Project Challenges and Opportunities

SAT solving has its strength in efficient techniques for exploring Boolean structures, learning, combining techniques, and developing dedicated heuristics. Symbolic Computation provides powerful procedures for systems of arithmetic constraints, and has expertise in simplification and preprocessing.

To allow further exploitation by the Satisfiability Checking community, Symbolic Computation tools must first be adapted to comply with SAT requirements as set out in the previous section. Cylindrical Algebraic Decomposition, Gröbner Bases and Virtual Substitution are algorithms of particular interest. However, this is a challenge that requires the expertise of computer algebra developers.

Conversely, Symbolic Computation could profit from exploiting successful SMT ideas, like dedicated data structures, dynamic heuristics, effective learning techniques, and approaches for constraint propagation and branch-and-bound. Even fewer SMT solvers are available for non-linear integer arithmetic.

The community is supported by conferences (e.g. CADE, ICAR, SMT) and journals (e.g. J. Automated Reasoning), while a role somewhat analogous to SIGSAM is played by the SatLive Forum.

Project Challenges and Opportunities

SAT solving has its strength in efficient techniques for exploring Boolean structures, learning, combining techniques, and developing dedicated heuristics. Symbolic Computation provides powerful procedures for systems of arithmetic constraints, and has expertise in simplification and preprocessing.

To allow further exploitation by the Satisfiability Checking community, Symbolic Computation tools must first be adapted to comply with SAT requirements as set out in the previous section. Cylindrical Algebraic Decomposition, Gröbner Bases and Virtual Substitution are algorithms of particular interest. However, this is a challenge that requires the expertise of computer algebra developers.

Conversely, Symbolic Computation could profit from exploiting successful SMT ideas, like dedicated data structures, dynamic heuristics, effective learning techniques, and approaches for constraint propagation and branch-and-bound. Even fewer SMT solvers are available for non-linear integer arithmetic.

The community is supported by conferences (e.g. CADE, ICAR, SMT) and journals (e.g. J. Automated Reasoning), while a role somewhat analogous to SIGSAM is played by the SatLive Forum.

Project Challenges and Opportunities

SAT solving has its strength in efficient techniques for exploring Boolean structures, learning, combining techniques, and developing dedicated heuristics. Symbolic Computation provides powerful procedures for systems of arithmetic constraints, and has expertise in simplification and preprocessing.

To allow further exploitation by the Satisfiability Checking community, Symbolic Computation tools must first be adapted to comply with SAT requirements as set out in the previous section. Cylindrical Algebraic Decomposition, Gröbner Bases and Virtual Substitution are algorithms of particular interest. However, this is a challenge that requires the expertise of computer algebra developers.

Conversely, Symbolic Computation could profit from exploiting successful SMT ideas, like dedicated data structures, dynamic heuristics, effective learning techniques, and approaches for constraint propagation and branch-and-bound. Even fewer SMT solvers are available for non-linear integer arithmetic.

The community is supported by conferences (e.g. CADE, ICAR, SMT) and journals (e.g. J. Automated Reasoning), while a role somewhat analogous to SIGSAM is played by the SatLive Forum.

Project Challenges and Opportunities

SAT solving has its strength in efficient techniques for exploring Boolean structures, learning, combining techniques, and developing dedicated heuristics. Symbolic Computation provides powerful procedures for systems of arithmetic constraints, and has expertise in simplification and preprocessing.

To allow further exploitation by the Satisfiability Checking community, Symbolic Computation tools must first be adapted to comply with SAT requirements as set out in the previous section. Cylindrical Algebraic Decomposition, Gröbner Bases and Virtual Substitution are algorithms of particular interest. However, this is a challenge that requires the expertise of computer algebra developers.

Conversely, Symbolic Computation could profit from exploiting successful SMT ideas, like dedicated data structures, dynamic heuristics, effective learning techniques, and approaches for constraint propagation and branch-and-bound. Even fewer SMT solvers are available for non-linear integer arithmetic.

The community is supported by conferences (e.g. CADE, ICAR, SMT) and journals (e.g. J. Automated Reasoning), while a role somewhat analogous to SIGSAM is played by the SatLive Forum.

Project Challenges and Opportunities

SAT solving has its strength in efficient techniques for exploring Boolean structures, learning, combining techniques, and developing dedicated heuristics. Symbolic Computation provides powerful procedures for systems of arithmetic constraints, and has expertise in simplification and preprocessing.

To allow further exploitation by the Satisfiability Checking community, Symbolic Computation tools must first be adapted to comply with SAT requirements as set out in the previous section. Cylindrical Algebraic Decomposition, Gröbner Bases and Virtual Substitution are algorithms of particular interest. However, this is a challenge that requires the expertise of computer algebra developers.

Conversely, Symbolic Computation could profit from exploiting successful SMT ideas, like dedicated data structures, dynamic heuristics, effective learning techniques, and approaches for constraint propagation and branch-and-bound. Even fewer SMT solvers are available for non-linear integer arithmetic.

The community is supported by conferences (e.g. CADE, ICAR, SMT) and journals (e.g. J. Automated Reasoning), while a role somewhat analogous to SIGSAM is played by the SatLive Forum.