SC-Square: Overview to 2021.

England, M

1613-0073

Publisher: CEUR Workshop Proceedings

SC-Square: Overview to 2021

Matthew England

Coventry University, Coventry, UK

Abstract

This extended abstract was written to accompany an invited talk at the 2021 SC-Square Workshop, where the author was asked to give an overview of SC-Square progress to date. The author first reminds the reader of the definition of SC-Square, then briefly outlines some of the history, before picking out some (personal) scientific highlights.

Keywords

symbolic computation, computer algebra systems, satisfiability checking, SMT solvers

1. SC-Square Definition

SC-Square, or SC^2, refers to the intersection of two fields of Computer Science which share that abbreviation: Symbolic Computation and Satisfiability Checking. The SC-Square community refers to people with an interest in *both* fields.

Satisfiability Checking refers to algorithms and solvers dedicated to ascertaining whether a system of logical constraints admits a solution (allocation of values to variables which satisfies the system). This grew out of the SAT-community and the success of SAT-solvers in answering many very large instances of the Boolean SAT Problem, despite the problem being NP-complete. It now encompasses logic problems with variables from a variety of mathematical domains. One popular paradigm for attacking such problems is to tackle the Boolean logic separately to the constraints in the domain by viewing the atoms of the formula as Boolean variables and employing a SAT solver; then using domain specific algorithms / software to see if the domain constraints assigned to be true can be mutually satisfied. This is often referred to as Satisfiability Modulo Theories (SMT) and the accompanying software as SMT-solvers.

Symbolic Computation refers to algorithms that perform symbolic mathematics efficiently, such as polynomial computations. Historic achievements in symbolic computation include algorithms for symbolic integration, polynomial factorisation, Gröbner bases for the effective solution of many problems concerning multivariate polynomials over algebraically-closed fields, and algorithms for addressing quantifier elimination and other problems involving a mixed system of equalities and inequalities on non-linear multivariate polynomials. Symbolic Computation algorithms are usually implemented in Computer Algebra Systems: large software products designed for use in mathematics research and education.

SC^2 2021: 6th International Workshop on Satisfiability Checking and Symbolic Computation, August 19–20, 2021

Matthew.England@coventry.ac.uk (M. England)

https://matthewengland.coventry.domains (M. England)

0000-0001-5729-3420 (M. England)

Traditionally, the two communities have been largely disjoint and unaware of the achievements of one another, despite there being significant areas of overlapping interest. In many domains of interest for SMT the theory reasoning would naturally use algorithms of symbolic computation. In the opposite direction; the integration of SAT solvers into computer algebra systems allows for more powerful logical reasoning; and the model-driven search algorithms of satisfiability provide inspiration for whole new algorithmic approaches in computer algebra.

2. SC-Square History

For more information on the separate histories of the two SCs we refer the reader, for example, to Section 2 of [3]. That paper was written to announce the start of the EU funded SC-Square Project[1]. This ran from 2016−2018 with the aim of bridging the gap between the communities to produce individuals whom can combine the knowledge and techniques of both fields to resolve problems currently beyond the scope of either. The project consortium consisted of a variety of EU universities, institutions and companies, and the project included a wider group of partners from all over the world. It was formed following the invited talk of Erika Ábrahaám at ISSAC 2015 [2] and a 2015 Dagstuhl Seminar[2].

The SC-Square Project funded new collaborations, new tool integrations, proposals on extensions to the SMT-LIB language standards, new collections of benchmarks, two summer schools in 2017[3] and 2018[4], a special issue of the Journal of Symbolic Computation (volume 100) [14], and the SC-Square Workshop Series[5].

Although the project finished in 2018, the collaborations it instigated have continued, as has the workshop series which bears its name. There have been six editions of the workshop to date, with two more planned:

2016 Timisoara, Romainia (as part of SYNASC 2016).

2017 Kaiserslautern, Germany (alongside ISSAC 2017).

2018 Oxford, UK (as part of FLoC 2018).

2019 Bern, Switzlerland (as part of SIAM AG19)

2020 Paris, France (online) (alongside IJCAR 2020)

2021 Texas, USA (online) (as part of SIAM AG21)

2022 Haifa, Israel (planned, as part of FLoC 2022)

2023 Tromsø Norway (planned, alongside ISSAC 2023)

The workshops take place as part of, or alongside, established conferences, alternating between those in computational algebra and logic. Each year there are two chairs, one from each SC.

[1] http://www.sc-square.org/EU-CSA.html
[2] # 15471: https://www.dagstuhl.de/en/program/calendar/semhp/?semnr=15471
[3] http://www.sc-square.org/CSA/school/
[4] http://ssa-school-2018.cs.manchester.ac.uk/
[5] http://www.sc-square.org/workshops.html

3. SC-Square Scientific Highlights

3.1. Non-linear Real Arithmetic

Algorithms for working with systems of non-linear polynomials are a core topic for symbolic computation, but also an important domain for SMT where it is referred to as Non-linear Real Arithmetic (NRA). Such algorithms are notoriously complex but also offer limitless applications, so it is no surprise that this is one area where there has been much activity[6].

Arguably the first algorithmic development in the scope of SC-Square was the NLSAT Algorithm of [20] which pre-dated and inspired the project. The authors re-purposed the symbolic computation theory of cylindrical algebraic decomposition from [13], for use in their proof framework (now known as MCSAT [15]) to solve satisfiability problems in NRA.

At a similar time, the SMT-RAT solver and toolbox [24] started developing implementations of a variety of computer algebra tools for use in SMT, including cylindrical algebraic decomposition, Gröbner Bases, Virtual Term Substitution, and more. This was preferable to using computer algebra systems directly as SMT theory solvers, since the algorithms needed adaption to suit the SMT requirements of efficient incrementality by constraint, backtracking and explanations for unsatisfiability. For examples of such adaptions see e.g., [25] for cylindrical algebraic decomposition and [21] for Gröbner bases.

The success of NLSAT and SMT-RAT inspired new algorithmic approaches. The Conflict Driven Cylindrical Algebraic Coverings of [4] gives an alternative repurposing of CAD technology for SMT, compatible with the traditional SMT proof framework. Meanwhile the NuCAD algorithm of [10] was inspired by NLSAT to use an incremental local cell construction to build a decomposition suitable for the more general quantifier elimination [9] application. This is an example of a new symbolic computation algorithm development inspired by algorithmic ideas from satisfiability checking.

Other SC-Square work in NRA includes the combination of computer algebra system Reduce/Redlog into SMT solver VeriT [16]; the combination of computer algebra with heuristics based on interval constraint propagation and subtropical satisfiability [16]; and the Incremental Linearlization techniques of [12].

3.2. Other Highlights

We note a few other SC-Square highlights the author is aware of:

- Popular commercial computer algebra system Maple can now read to and from SMT-LIB [17] and ships with the both the Z3 and MapleSAT solvers.
- The computer algebra system CoCoA now releases the open-source CoCoALib: a C++ Library that underpins many of its routines [1], more suitable for use by SMT solvers.
- The MathCheck project has used a combination of SAT-solvers and computer algebra to make progress on a variety of combinatorics problems, enumerating new cases and verifying conjectures. E.g., Williamson Matrices [7]; Golay Pairs [8]; Good Matrices [6].
- Algebraic techniques were key for the circuit verification work [22], and in combination with SAT-solvers [23].

[6]The author also acknowledges that this is his field of interest which likely led this to be forefront of his highlights!

- The Boolean SAT problem has also benefited from symbolic computation via Boolean Gröbner Bases and parallel computation on both the conjunctive and algebraic normal forms of a problem [18].
- An emerging direction in both SCs is the production of proofs and some initial ideas here have emerged from SC-Square: [19], [5].
- The SC-Square workshops have been home to application descriptions that are new to both SCs, including in the fields of Economics [28], Dynamic Geometry [29], and knot theory [26], [27].

4. SC-Square Future

The past successes of SC-Square warrant a promising future for the community. The workshop series will continue in 2022 and 2023 at the least. There will also be a new Dagstuhl Seminar on the topic[7]. However, it is still the case that the bulk of both communities are working independently from each other, and greater integration would surely bring further successes.

Acknowledgements

The author is supported by the EPSRC project, *Pushing Back the Doubly-Exponential Wall of Cylindrical Algebraic Decomposition* (EP/T015748/1). Thanks to the reviewer for suggestions that improved the text.

References

[1] Abbott, J., Bigatti, A.M.: What is new in CoCoA? In: Hong, H., Yap, C. (eds.) Mathematical Software – ICMS 2014. Lecture Notes in Computer Science, vol. 8592, pp. 352–358. Springer Heidelberg (2014), https://doi.org/10.1007/978-3-662-44199-2_55

[2] Ábrahám, E.: Building bridges between symbolic computation and satisfiability checking. In: Proceedings of the 2015 International Symposium on Symbolic and Algebraic Computation. pp. 1–6. ISSAC '15, ACM (2015), https://doi.org/10.1145/2755996.2756636

[3] Ábrahám, E., Abbott, J., Becker, B., Bigatti, A.M., Brain, M., Buchberger, B., Cimatti, A., Davenport, J.H., England, M., Fontaine, P., Forrest, S., Griggio, A., Kroening, D., Seiler, W.M., Sturm, T.: SC^2: Satisfiability checking meets symbolic computation. In: Kohlhase, M., Johansson, M., Miller, B., de Moura, L., Tompa, F. (eds.) Intelligent Computer Mathematics: Proceedings CICM 2016, Lecture Notes in Computer Science, vol. 9791, pp. 28–43. Springer International Publishing (2016), https://doi.org/10.1007/978-3-319-42547-4_3

[4] Ábrahám, E., Davenport, J.H., England, M., Kremer, G.: Deciding the consistency of non-linear real arithmetic constraints with a conflict driven search using cylindrical algebraic coverings. Journal of Logical and Algebraic Methods in Programming **119**, 100633 (2021), https://doi.org/10.1016/j.jlamp.2020.100633

[5] Ábrahám, E., Davenport, J.H., England, M., Kremer, G., Tonks, Z.: New opportunities for the formal proof of computational real geometry? In: Fontaine, P., Korovin, K., Kotsireas,

[7] # 22072: https://www.dagstuhl.de/en/program/calendar/semhp/?semnr=22072

I.S., Rümmer, P., Tourret, S. (eds.) Proceedings of the 5th Workshop on Satisfiability Checking and Symbolic Computation (SC^2 2020). pp. 178–188. No. 2752 in CEUR Workshop Proceedings (2020), http://ceur-ws.org/Vol-2752/

[6] Bright, C., Doković, D.Z., Kotsireas, I., Ganesh, V.: A SAT+CAS approach to finding good matrices: New examples and counterexamples. In: Proceedings of the Thirty-Third AAAI Conference on Artificial Intelligence (2019), https://doi.org/10.1609/aaai.v33i01.33011435

[7] Bright, C., Kotsireas, I., Ganesh, V.: Applying computer algebra systems with SAT solvers to the Williamson conjecture. Journal of Symbolic Computation **100**, 187–209 (2020), https://doi.org/10.1016/j.jsc.2019.07.024

[8] Bright, C., Kotsireas, I., Heinle, A., Ganesh, V.: Enumeration of complex Golay pairs via programmatic SAT. In: Proceedings of the 2018 ACM on International Symposium on Symbolic and Algebraic Computation. pp. 111–118. ISSAC '18, ACM (2018), https://doi.org/10.1145/3208976.3209006

[9] Brown, C.: Projection and quantifier elimination using non-uniform cylindrical algebraic decomposition. In: Proceedings of the 2017 ACM on International Symposium on Symbolic and Algebraic Computation. pp. 53–60. ISSAC '17, ACM (2017), https://doi.org/10.1145/3087604.3087651

[10] Brown, C.W.: Open non-uniform cylindrical algebraic decompositions. In: Proceedings of the 2015 International Symposium on Symbolic and Algebraic Computation. pp. 85–92. ISSAC '15, ACM (2015), https://doi.org/10.1145/2755996.2756654

[11] Caviness, B., Johnson, J.: Quantifier Elimination and Cylindrical Algebraic Decomposition. Texts & Monographs in Symbolic Computation, Springer-Verlag (1998), https://doi.org/10.1007/978-3-7091-9459-1

[12] Cimatti, A., Griggio, A., Irfan, A., Roveri, M., Sebastiani, R.: Incremental linearization: A practical approach to satisfiability modulo nonlinear arithmetic and transcendental functions. In: 20th Intl. Symposium on Symbolic and Numeric Algorithms for Scientific Computing (SYNASC). pp. 19–26 (2018), http://doi.org/10.1109/SYNASC.2018.00016

[13] Collins, G.E.: Quantifier elimination for real closed fields by cylindrical algebraic decomposition. In: Proceedings of the 2nd GI Conference on Automata Theory and Formal Languages. pp. 134–183. Springer-Verlag (reprinted in the collection [11]) (1975), https://doi.org/10.1007/3-540-07407-4_17

[14] Davenport, J.H., England, M., Griggio, A., Sturm, T., Tinelli, C.: Symbolic computation and satisfiability checking: Editorial. Journal of Symbolic Computation **100**, 1–10 (2020), https://doi.org/10.1016/j.jsc.2019.07.017

[15] de Moura, L., Jovanović, D.: A model-constructing satisfiability calculus. In: Giacobazzi, R., Berdine, J., Mastroeni, I. (eds.) Verification, Model Checking, and Abstract Interpretation (Proc. VMCAI 2013), pp. 1–12. Springer Berlin Heidelberg (2013), https://doi.org/10.1007/978-3-642-35873-9_1

[16] Fontaine, P., Ogawa, M., Sturm, T., Khanh To, V., Tung Vu, X.: Wrapping computer algebra is surprisingly successful for non-linear SMT. In: Bigatti, A.M., Brain, M. (eds.) Proceedings of the 3rd Workshop on Satisfiability Checking and Symbolic Computation (SC^2 2018). pp. 110–117. CEUR Workshop Proc. 2189 (2018), http://ceur-ws.org/Vol-2189/

[17] Forrest, S.A.: Integration of SMT-LIB support into maple. In: England, M., Ganesh, V. (eds.) Proceedings of the 2nd Intl. Workshop on Satisfiability Checking and Symbolic

Computation (SC^2 2017). CEUR Workshop Proc. 1974 (2017), http://ceur-ws.org/Vol-1974/
[18] Horáček, J., Kreuzer, M.: On conversions from CNF to ANF. Journal of Symbolic Computation **100**, 164–186 (2020), https://doi.org/10.1016/j.jsc.2019.07.023
[19] Jebelean, T.: Techniques for natural-style proofs in elementary analysis (work in progress). In: Bigatti, A.M., Brain, M. (eds.) Proceedings of the 3rd Workshop on Satisfiability Checking and Symbolic Computation (SC^2 2018). pp. 122–131. CEUR Workshop Proc. 2189 (2018), http://ceur-ws.org/Vol-2189/
[20] Jovanovic, D., de Moura, L.: Solving non-linear arithmetic. In: Gramlich, B., Miller, D., Sattler, U. (eds.) Automated Reasoning: 6th Intl. Joint Conference (IJCAR), Lecture Notes in Computer Science, vol. 7364, pp. 339–354. Springer (2012), https://doi.org/10.1007/978-3-642-31365-3_27
[21] Junges, S., Loup, U., Corzilius, F., Ábrahám, E.: On Gröbner bases in the context of Satisfiability-Modulo-Theories solving over the real numbers. In: Muntean, T., Poulakis, D., Rolland, R. (eds.) Algebraic Informatics, Lecture Notes in Computer Science, vol. 8080, pp. 186–198. Springer Berlin Heidelberg (2013), https://doi.org/10.1007/978-3-642-40663-8_18
[22] Kaufmann, D.: Formal verification of integer multiplier circuits using algebraic reasoning: A survey. In: Drechsler, R., Große, D. (eds.) Recent Findings in Boolean Techniques: Selected Papers from the 14th Intl. Workshop on Boolean Problems, pp. 1–27. Springer International Publishing (2021), https://doi.org/10.1007/978-3-030-68071-8_1
[23] Kaufmann, D., Biere, A., Kauers, M.: Verifying large multipliers by combining SAT and computer algebra. In: Formal Methods in Computer Aided Design (FMCAD 2019). pp. 28–36. IEEE (2019), https://doi.org/10.23919/FMCAD.2019.8894250
[24] Kremer, G., Ábrahám, E.: Modular strategic SMT solving with SMT-RAT. Acta Universitatis Sapientiae, Informatica **10**(1), 5–25 (2018), http://dx.doi.org/10.2478/ausi-2018-0001
[25] Kremer, G., Ábrahám, E.: Fully incremental CAD. Journal of Symbolic Computation **100**, 11–37 (2020), https://doi.org/10.1016/j.jsc.2019.07.018
[26] Lisitsa, A., Vernitski, A.: Automated reasoning for knot semigroups and π-orbifold groups of knots. In: Blömer, J., Kotsireas, I.S., Kutsia, T., Simos, D.E. (eds.) Mathematical Aspects of Computer and Information Sciences (Proc. MACIS '17), Lecture Notes in Computer Science, vol. 10693, pp. 3–18. Springer International Publishing (2017), https://doi.org/10.1007/978-3-319-72453-9_1
[27] Meesum, S.M., Prathamesh, T.V.H.: Unknot recognition through quantifier elimination. In: Bigatti, A.M., Brain, M. (eds.) Proceedings of the 3rd Workshop on Satisfiability Checking and Symbolic Computation (SC^2 2018). pp. 77–87. CEUR Workshop Proc. 2189 (2018), http://ceur-ws.org/Vol-2189/
[28] Mulligan, C., Bradford, R., Davenport, J.H., England, M., Tonks, Z.: Non-linear real arithmetic benchmarks derived from automated reasoning in economics. In: Bigatti, A.M., Brain, M. (eds.) Proceedings of the 3rd Workshop on Satisfiability Checking and Symbolic Computation (SC^2 2018). pp. 48–60. CEUR Workshop Proc. 2189 (2018), http://ceur-ws.org/Vol-2189/
[29] Vajda, R., Kovács, Z.: GeoGebra and the realgeom reasoning tool. In: Fontaine, P., Korovin, K., Kotsireas, I.S., Rümmer, P., Tourret, S. (eds.) Proceedings of the 5th Workshop on Satisfiability Checking and Symbolic Computation (SC^2 2020). pp. 204–219. No. 2752 in CEUR Workshop Proceedings (2020), http://ceur-ws.org/Vol-2752/