



EPiC Series in Computing

Volume 51, 2017, Pages 1–5

ARCADE 2017. 1st International Workshop
on Automated Reasoning: Challenges, Appli-
cations, Directions, Exemplary Achievements



A Report of ARCADE 2017

Giles Reger¹ and Dmitriy Traytel²

¹ University of Manchester, Manchester, UK

² Institute of Information Security, Department of Computer Science, ETH Zürich, Switzerland

Abstract

The ARCADE workshop took place on the 6th August 2017 in Gothenburg, colocated with CADE-26. ARCADE stands for Automated Reasoning: Challenges, Applications, Directions, Exemplary achievements. The goal of this workshop was to bring together key people from various sub-communities of automated reasoning—such as SAT/SMT, resolution, tableaux, theory-specific calculi (e.g. for description logic, arithmetic, set theory), interactive theorem proving—to discuss the present, past, and future of the field.

The workshop was attended by 45 people and consisted of 14 short presentations and 5 discussion sessions. The format was informal and highly interactive. This report gives a general overview of the presentations and summarizes the discussion sessions.

1 Introduction

This report gives a general overview of the positions presented at ARCADE and summarizes the discussions' outcomes from each session.

The main goal of the ARCADE workshop was to bring together key people from various sub-communities of automated reasoning to discuss the present, past, and future of the field. ARCADE invited non-technical position statements of 2-4 pages. Out of 19 submitted abstracts the program committee selected 14 for presentation and discussion at the workshop.

With its 45 attendees, the first installment of the ARCADE workshop was half as large as CADE itself. Several participants came to Gothenburg only or mainly because of ARCADE; in one instance this involved an expensive trip from the US just to give a 10 minute ARCADE presentation and to helpfully contribute to the discussion. (We make this comment to highlight the importance placed on the meeting by its participants.)

The workshop was divided into five sessions. Each session consisted of two or three short talks of either 5 or 10 minutes followed by a discussion of the presented topics driven by questions submitted by the authors. The accepted abstracts and proposed questions were published on the workshop's webpage in advance of the workshop to allow attendees to consider these points beforehand. Here, we present a high-level summary of the talks and associated discussions.

2 Session 1: Bridges Between Communities

The first session discussed how to bridge the gaps between the various sub-communities of automated reasoning (AR) as well as the even wider gaps between AR and other communities.

In the first talk of the workshop, James Davenport presented the SC² project about combining SMT solvers and computer algebra systems. The corresponding position statement reflects on the difficulties faced when bringing two communities together. The general conclusions from this presentation and associated discussion were: firstly, patience and perseverance are needed (it took 2-3 years to get things going in this case); secondly, a barrier is often a lack of common terminology and this needs to be acknowledged and addressed early on; and finally, a necessary starting point is a focus on shared applications. It was observed that we need large international efforts like SC² as it is virtually impossible for one person (or even a few people) to have enough expertise in two substantially different fields.

Next, Simon Cruanes presented his idea of *Making Automatic Theorem Provers more Versatile*. Simon considered the limits of today’s automatic theorem provers (including both SMT solvers and first-order superposition provers in this definition) and suggested three directions for further research: (1) combining SMT and superposition, (2) supporting user-defined theories, and (3) incorporating induction and higher-order reasoning. Once such combinations and extensions are considered, it was then asked whether we can hope to formalize all techniques in one system in an elegant (which some took to mean simple) way. Currently, combinations tend to be achieved merely by defining the interface between two approaches.

In the third talk *Towards Strong Higher-Order Automation for Fast Interactive Verification*, Jasmin Blanchette presented the challenges that his ERC project Matryoshka tackles. Matryoshka attempts to lift first-order provers to the higher-order setting and subsequently integrate those new higher-order automatic theorem provers in interactive theorem provers. The key observation is thereby that many “real-world” problems are mostly first-order in nature, but “tainted” by a small amount of higher-order features, making them inaccessible to today’s approaches. Matryoshka’s approach is centered around a pipeline of tools with good interfaces. This motivates the need for our community to agree on such interfaces and to work with them.

In a final round of discussion, it was observed that many of the examples given in this session referred to the software verification application area. One person mentioned that, whilst this application area has been a very good ‘poster child’ for the application of automated reasoning techniques, perhaps we could, as a community, give a little more focus to the many diverse alternative application areas that do exist. A reply to this comment was that more people from outside our community are interested in software *analysis* rather than *verification*.

3 Session 2: Artificial Intelligence and Machine Learning

The topic of the second session was machine learning (ML) and more generally artificial intelligence (AI). The session consisted of two talks: *Automated Reasoning for Explainable Artificial Intelligence* by Maria Paola Bonacina and *We know (nearly) nothing! But can we learn?* by Stephan Schulz. A third thought-provoking position statement *AI/ML at CADE* by Josef Urban is part of these proceedings. It was not presented on site as Josef could not attend ARCADE. Yet, the discussion had also touched upon the questions Josef had contributed.

Maria Paola observed that although historically AR was a part of AI, AI has become synonymous with ML, while AR has become a second choice technology in cases where ML fails. As a community we cannot be satisfied with this status, but the immediate countermeasures remained unclear. As a possible way to benefit from AR technology in the ML world, Maria Paola formulated the challenge of using AR to explain ML-based predictions. In the follow-up discussion, one participant stated that ML currently enjoys many “cheap successes”, which mainly rely on the availability of better hardware, better software, better (and more) data, and better education, and not so much on a conceptual revolution. The idea of a joint workshop

bringing AR and ML together was proposed.

Stephan Schulz presented the other possible way of combining AR and ML: namely using ML to steer the search in AR. To demonstrate his point, Stephan divided a first-order prover into three parts: calculus, implementation, and search control. He then empirically showed on the example of the E prover that changes to the calculus and implementation parts have led to modest improvements in performance. In contrast, search heuristics were the greatest source of improvements despite often being ad hoc and still unsatisfactory. And the prime successes of ML-powered approaches to Chess and Go, show that ML can be the way to go when selecting heuristics. In the domain of theorem proving, “What to learn from?” remains the key question.

Josef Urban’s standpoint is quite similar to Stephan’s. Additionally, in his statement, Josef points out (quite directly) that a considerable portion of the CADE community lacks the understanding of ML-based techniques and seems therefore to be averse to them.

4 Session 3: User Interfaces and Industrial Usage

The third session dealt with the user friendliness of automated reasoning tools and their applicability in industry.

In his presentation *24 Challenges in Deductive Software Verification*, Reiner Hähnle considered the challenges to the deduction community both in general and more specifically to the setting of deductive software verification. Two main points arose in discussions. First, the discussion considered the *infrastructure* required to carry out research in our community. It was pointed out that to obtain sustainable support for infrastructure from funding agencies, computer science in general might need to be reclassified as an engineering science as opposed to a theoretical science. Second, the need for user-driven research was discussed along with the incentives that could make it happen, such as the support (through calls for papers etc.) for experimental user studies and for industry-oriented problems in competitions.

The second talk of this session was *Industrial Use of ACL2: Applications, Achievements, Challenges, and Directions*, in which Marijn Heule presented the industry success story of the ACL2 proof assistant. The key to success was that “success in industry” was an original design goal of ACL2. It was pointed out that the needs of industry were not necessarily aligned with the research directions in automated reasoning and more should be done to acknowledge the concerns of industry. For example, it was noted that typical industrial users care more about usability and efficiency, rather than about issues such as completeness.

5 Session 4: Proof Formats

The fourth session focussed on methods to make proofs produced by automated reasoners more easily checkable. It consisted of three talks.

In his presentation *Beyond DRAT: Challenges in Certifying UNSAT*, Bertram Felgenhauer proposed to extend the DRAT format for propositional unsatisfiability proofs to cover advanced features used by SAT solvers, such as symmetry breaking. The proposal was countered by the desire to retain the simplicity of DRAT to ensure both the efficiency and trustworthiness of a checker’s implementation. While the feature of being efficiently checkable must be part of the concrete extension proposal, the existence of formally verified DRAT checkers (two of which were presented at CADE-26) is a good starting point to attack the trustworthiness concern.

In the second talk, Marijn Heule discussed *The Potential of Interference-Based Proof Systems*. He defined interferences as those reasoning steps that preserve satisfiability but not logical

equivalence. Clause deletion as used in DRAT is an example for an interference rule. Marijn outlined the usefulness of interferences when checking unsatisfiability proofs for SAT (DRAT) and QBF (QRAT) and urged the community to study interference-based calculi in the first-order setting. It was pointed out that Skolemization and introduction of new symbols via definitions are examples of interferences, which are already used in first-order provers.

As of today, for first-order provers there is no common proof format, except for the very general standardized but semantic-less TSTP syntax. The next presentation *Checkable Proofs for First-Order Theorem Proving* by Martin Suda aimed at improving this situation. Martin outlined several desiderata for such a format, with the most important one being widespread adoption by the community. The competition was identified as the right instrument to enforce adoption. However, the subsequent discussion showed that the community is not yet at the point to leave TSTP behind, as there is no agreement on the “prime” first-order calculus (unlike in the propositional case, where CDCL takes this role).

6 Session 5: Provers

The last session outlined some current challenges for actual reasoners. All three talks covered quite orthogonal topics for a diverse selection of automated reasoning tools.

Gopal Gupta made *A Case for Predicate Answer Set Programming* (ASP). ASP can be viewed as logic programming extended with a “negation as failure” semantics. Thereby it mimics the human non-monotonic way of reasoning by jumping to conclusions and withdrawing earlier deductions. Gopal pointed out that existing ASP systems are usually based on finite *model-finding* and can only handle those logic programs that are finitely groundable, which for example excludes programming with lists and other infinite types. Inspired by this and several other limitations of model-finding approaches, Gopal called for investigating *proof-finding* (also called *query-driven*) approaches, which could support programs that are not finitely groundable. The follow-up discussion revolved around several success-stories of lifting ground calculi to first-order calculi in the automated reasoning community.

Next, Andrew Reynolds explored the *Challenges for Fast Synthesis Procedures in SMT*. Andrew started by summarizing his earlier work on using an SMT solver as a stand-alone synthesis tool for so called *single invocation* specifications—second-order formulas that can be converted to equivalent first-order formulas using anti-Skolemization. He then formulated the challenge of going beyond this fragment to truly second- or higher-order formulas.

The last talk of the session and the workshop was given by Christoph Weidenbach. The presentation was only loosely related to his position statement *Do Portfolio Solvers Harm?* that can be found in this proceedings. In the presentation, Christoph has viewed CDCL as an “Ordering Portfolio Order Resolution” solver. Similarly, he challenged the community to extract common principles for different existing decision procedures for the Bernays-Schönfinkel subclass of first-order logic in order to view those approaches as a semantic “Model Portfolio”.

In the follow-up discussion, it was observed that the unifying principle of this session’s challenges was the identification of commonalities and weak spots in existing approaches. A resulting consideration was to focus the evaluation of reasoners around the number of uniquely solved problems, as this helps rewarding solvers which attack exactly the weak spots. Geoff Sutcliffe confirmed that the CASC competition measures this, but does not reward tools that perform well according to this metric.

7 Concluding Discussion

In a final round of discussion, we considered various constellations of N authors and M pages for a document that would summarize the state-of-the-art in AR and upcoming challenges. For $N > 40$ and $M < 5$, an idea of a CACM article on the challenges in AR was discussed. Another proposal was a survey of AR, with $N < 20$ and $M > 40$. For $N = 1$ and $M > 300$, an more up-to-date textbook than the “Handbook of Automated Reasoning” to attract new PhD students was considered helpful. In this space, John Harrison’s “Handbook of Practical Logic and Automated Reasoning” is a good hands-on introduction to many topics, but it is deliberately selective in what it covers to keep things “elementary”. Rumors tell that Christoph Weidenbach is working on another textbook draft.

Acknowledgments. The organization of ARCADE was very smooth and easy going, mainly due to the great support that we received from the CADE community. We thank Christoph Weidenbach, who initiated the workshop and jointly with Jasmin Blanchette helped recruiting most of the program committee. We are very grateful to Wolfgang Ahrendt and the local organization team for taking great care of everything on-site and reserving the venue for the ARCADE dinner. The latter was suggested by Reiner Hähnle and turned out to be a great choice. We thank the program committee (and one subreviewer) for detailed, careful, and constructive reviews and the pleasant discussion. This report benefitted from notes taken by James Davenport during the workshop and from Jasmin Blanchette’s comments. Finally, we thank the authors and the attendees for creating such a diverse and timely program and discussion that will certainly have an impact on the future of automated reasoning.