TOOLympics 2019: An Overview of Competitions in Formal Methods

https://tacas.info/toolympics.php

Ezio Bartocci[1], Dirk Beyer[2], Paul E. Black[3], Grigory Fedyukovich[4], Hubert Garavel[5], Arnd Hartmanns[6], Marieke Huisman[7], Fabrice Kordon[8], Julian Nagele[9], Mihaela Sighireanu[10], Bernhard Steffen[11], Martin Suda[12], Geoff Sutcliffe[13], Tjark Weber[14], and Akihisa Yamada[15]

[1] TU Wien, Austria, ezio.bartocci@tuwien.ac.at
[2] LMU Munich, Germany, dirk.beyer@sosy-lab.org
[3] NIST, USA, paul.black@nist.gov
[4] Princeton University, USA, grigoryf@cs.princeton.edu
[5] Univ. Grenoble Alpes, Inria, Cnrs, Grenoble Inp, Lig, France, hubert.garavel@inria.fr
[6] University of Twente, Netherlands, a.hartmanns@utwente.nl
[7] University of Twente, Netherlands, m.huisman@utwente.nl
[8] Sorbonne Université, France, Fabrice.Kordon@lib6.fr
[9] Queen Mary University of London, UK, j.nagele@qmul.ac.uk
[10] University Paris Diderot, France, mihaela.sighireanu@irif.fr
[11] TU Dortmund, Germany, steffen@cs.tu-dortmund.de
[12] Czech Technical University in Prague, Czech Republic, martin.suda@cvut.cz
[13] University of Miami, USA, geoff@cs.miami.edu
[14] Uppsala University, Sweden, tjark.weber@it.uu.se
[15] NII, Japan, akihisayamada@nii.ac.jp

Abstract. Evaluation of scientific contributions can be done in many different ways. For the various research communities working on the verification of systems (software, hardware, or the underlying involved mechanisms), it is important to bring together the community and to compare the state of the art, in order to identify progress of and new challenges in the research area. Competitions are a suitable way to do that.

The first verification competition was created in 1992 (SAT competition), shortly followed by the CASC competition in 1996. Since the year 2000, the number of dedicated verification competitions is steadily increasing. Many of these events now happen regularly, gathering researchers that would like to understand how well their research prototypes work in practice. Scientific results have to be reproducible, and powerful computers are becoming cheaper and cheaper, thus, these competitions are becoming an important means for advancing research in verification technology.

TOOLympics 2019 is an event to celebrate the achievements of the various competitions, and to understand their commonalities and differences. This volume is dedicated to the presentation of the 16 competitions that joined TOOLympics as part of the celebration of the 25^{th} anniversary of the TACAS conference.

1 Introduction

Over the last years, our society's dependency on digital systems has been steadily increasing. At the same time, we see that also the complexity of such systems is continuously growing, which increases the chances of such systems behaving unreliably, with many undesired consequences. In order to master this complexity, and to guarantee that digital systems behave as desired, software tools are designed that can be used to analyze and verify the behavior of digital systems. These tools are becoming more prominent, in academia as well as in industry. The range of these tools is enormous, and trying to understand which tool to use for which system is a major challenge. In order to get a better grip on this problem, many different competitions and challenges have been created, aiming in particular at better understanding the actual profile of the different tools that reason about systems in a given application domain.

The first competitions started in the 1990s (e.g., SAT and CASC). After the year 2000, the number of competitions has been steadily increasing, and currently we see that there is a wide range of different verification competitions. We believe there are several reasons for this increase in the number of competitions in the area of formal methods:

- increased computing power makes it feasible to apply tools to large benchmark sets,
- tools are becoming more mature,
- growing interest in the community to show practical applicability of theoretical results, in order to stimulate technology transfer,
- growing awareness that reproducibility and comparative evaluation of results is important, and
- organization and participation in verification competitions is a good way to get scientific recognition for tool development.

We notice that despite the many differences between the different competitions and challenges, there are also many similar concerns, in particular from an organizational point of view:

- How to assess adequacy of benchmark sets, and how to establish suitable input formats? And what is a suitable license for a benchmark collection?
- How to execute the challenges (on-site vs. off-site, on controlled resources vs. on individual hardware, automatic vs. interactive, etc.)?
- How to evaluate the results, e.g., in order to obtain a ranking?
- How to ensure fairness in the evaluation, e.g., how to avoid bias in the benchmark sets, how to reliably measure execution times, and how to handle incorrect or incomplete results?
- How to guarantee reproducibility of the results?
- How to achieve and measure progress of the state of the art?
- How to make the results and competing tools available so that they can be leveraged in subsequent events?

Therefore, as part of the celebration of 25 years of TACAS we organized TOOLympics, as an occasion to bring together researchers involved in competition organization. It is a goal of TOOLympics to discuss similarities and differences between the participating competitions, to facilitate cross-community communication to exchange experiences, and to discuss possible cooperation concerning benchmark libraries, competition infrastructures, publication formats, etc. We hope that the organization of TOOLympics will put forward the best practices to support competitions and challenges as useful and successful events.

In the remainder of this paper, we give an overview of all competitions participating in TOOLympics, as well as an outlook on the future of competitions. Table 1 provides references to other papers (also in this volume) providing additional perspective, context, and details about the various competitions. There are more competitions in the field, e.g., ARCH-COMP [1], ICLP Comp, MaxSAT Evaluation, Reactive Synthesis Competition [56], QBFGallery [72], and SyGuS-Competition.

2 Overview of all Participating Competitions

A competition is an event that is dedicated to fair comparative evaluation of a set of participating contributions at a given time. This section shows that such participating contributions can be of different forms: tools, result compilations, counterexamples, proofs, reasoning approaches, solutions to a problem, etc.

Table 1 categorizes the TOOLympics competitions. The first column names the competition (and the digital version of this article provides a link to the competition web site). The second column states the year of the first edition of the competition, and the third column the number of editions of the competition. The next two columns characterize the way the participating contributions are evaluated: Most of the competitions are evaluating automated tools that do not require user interaction and the experiments are executed by benchmarking environments, such as BenchExec [28], BenchKit [67], or StarExec [91]. However, some competitions require a manual evaluation, due to the nature of the competition and its evaluation criteria. The next two columns show where and when the results of the competition is determined: on-site during the event or off-site before the event takes place. Finally, the last column provides references to the reader to look up more details about each of the competitions.

The remainder of this section introduces the various competitions of TOOLympics 2019.

2.1 CASC: The CADE ATP System Competition

Organizer: Geoff Sutcliffe (Univ. of Miami, USA)
Webpage: http://www.tptp.org

The CADE ATP System Competition (CASC) [106] is held at each CADE and IJCAR conference. CASC evaluates the performance of sound, fully automatic,

Competition	Year first competition	Number editions	Automated evaluation	Interactive evaluation	On-site evaluation	Off-site evaluation	Competition reports
CASC	1996	23	●		●		[96–108, 115] [77, 78, 92–95, 109–114, 116]
CHC-COMP	2018	2	●		●		
CoCo	2012	8	●		●		[2, 3, 75]
CRV	2014	4	●			●	[11–13, 40, 80, 81]
MCC	2011	9	●			●	[62–66, 68–71]
QComp	2019	1	●			●	[46]
REC	2006	5	●			●	[35–38, 41]
RERS	2010	9	●	●		●	[42, 43, 47–49, 57–59]
SAT	1992	12	●			●	[4, 5, 14, 15, 85]
SL-COMP	2014	3	●			●	[83, 84]
SMT-COMP	2005	13	●			●	[6–10, 32–34]
SV-COMP	2012	8	●			●	[16–22]
termCOMP	2004	16	●			●	[44, 45, 73, 117]
Test-Comp	2019	1	●			●	[23]
VerifyThis	2011	8		●	●		[26, 31, 39, 50–55]

Table 1. Categorization of the competitions participating in TOOLympics 2019; planned competition Rodeo not contained in the table; CHC-COMP report not yet published (slides available: https://chc-comp.github.io/2018/chc-comp18.pdf)

classical logic Automated Theorem Proving (ATP) systems. The evaluation is in terms of: the number of problems solved, the number of problems solved with a solution output, and the average runtime for problems solved; in the context of: a bounded number of eligible problems, chosen from the TPTP Problem Library, and specified time limits on solution attempts. CASC is the longest running of the various logic solver competitions, with the 25th event to be held in 2020. This longevity has allowed the design of CASC to evolve into a sophisticated and stable state. Each year's experiences lead to ideas for changes and improvements, so that CASC remains a vibrant competition. CASC provides an effective public evaluation of the relative capabilities of ATP systems. Additionally, the organization of CASC is designed to stimulate ATP research, motivate development and implementation of robust ATP systems that are useful and easily deployed in applications, provide an inspiring environment for personal interaction between ATP researchers, and expose ATP systems within and beyond the ATP community.

2.2 CHC-COMP: Competition on Constrained Horn Clauses

Organizers: Grigory Fedyukovich (Princeton Univ., USA), Arie Gurfinkel (Univ. of Waterloo, Canada), and Philipp Rümmer (Uppsala Univ., Sweden)
Webpage: https://chc-comp.github.io/

Constrained Horn Clauses (CHC) is a fragment of First Order Logic (FOL) that is sufficiently expressive to describe many verification, inference, and synthesis problems including inductive invariant inference, model checking of safety properties, inference of procedure summaries, regression verification, and sequential equivalence. The CHC competition (CHC-COMP) compares state-of-the-art tools for CHC solving with respect to performance and effectiveness on a set of publicly available benchmarks. The winners among participating solvers are recognized by measuring the number of correctly solved benchmarks as well as the runtime. The results of CHC-COMP 2019 will be announced in the HCVS workshop affiliated with ETAPS.

2.3 CoCo: Confluence Competition

Organizers: Aart Middeldorp (Univ. of Innsbruck, Austria), Julian Nagele (Queen Mary Univ. of London, UK), and Kiraku Shintani (JAIST, Japan)
Webpage: http://project-coco.uibk.ac.at/

The Confluence Competition (CoCo) exists since 2012. It is an annual competition of software tools that aim to (dis)prove confluence and related (undecidable) properties of a variety of rewrite formalisms automatically. CoCo runs live in a single slot at a conference or workshop and is executed on the cross-community competition platform StarExec. For each category, 100 suitable problems are randomly selected from the online database of confluence problems (COPS). Participating tools must answer YES or NO within 60 seconds, followed by a justification that is understandable by a human expert; any other output signals that the tool could not determine the status of the problem. CoCo 2019 features new categories on commutation, confluence of string rewrite systems, and infeasibility problems.

2.4 CRV: Competition on Runtime Verification

Organizers: Ezio Bartocci (TU Wien, Austria), Yliès Falcone (Univ. Grenoble Alpes/CNRS/INRIA, France), and Giles Reger (Univ. of Manchester, UK)
Webpage: https://www.rv-competition.org/

Runtime verification (RV) is a class of lightweight scalable techniques for the analysis of system executions. We consider here specification-based analysis, where executions are checked against a property expressed in a formal specification language.

The core idea of RV is to instrument a software/hardware system so that it can emit events during its execution. These events are then processed by a monitor that is automatically generated from the specification. During the last decade, many important tools and techniques have been developed. The growing number of RV tools developed in the last decade and the lack of standard benchmark suites as well as scientific evaluation methods to validate and test new techniques have motivated the creation of a venue dedicated to comparing and evaluating RV tools in the form of a competition.

The Competition on Runtime Verification (CRV) is an annual event, held since 2014, and organized as a satellite event of the main RV conference. The competition is in general organized in different tracks: (1) offline monitoring, (2) online monitoring of C programs, and (3) online monitoring of Java programs. Over the first three years of the competition 14 different runtime verification tools competed on over 100 different benchmarks[16].

In 2017 the competition was replaced by a workshop aimed at reflecting on the experiences of the last three years and discussing future directions. A suggestion of the workshop was to held a benchmark challenge focussing on collecting new relevant benchmarks. Therefore, in 2018 a benchmark challenge was held with a track for Metric Temporal Logic (MTL) properties and an Open track. In 2019 CRV will return to a competition comparing tools, using the benchmarks from the 2018 challenge.

2.5 MCC: The Model Checking Contest

Organizers: Fabrice Kordon (Sorbonne Univ., CNRS, France), Hubert Garavel (Univ. Grenoble Alpes/INRIA/CNRS, Grenoble INP/LIG, France), Lom Messan Hillah (Univ. Paris Nanterre, CNRS, France), Francis Hulin-Hubard (CNRS, Sorbonne Univ., France), Loïg Jezequel (Univ. de Nantes, CNRS, France), and Emmanuel Paviot-Adet (Univ. de Paris, CNRS, France)
Webpage: https://mcc.lip6.fr/

Since 2011, the Model Checking Contest (MCC) is an annual competition of software tools for model checking. Tools are confronted to an increasing benchmark set gathered from the whole community (currently, 88 parameterized models totalling 951 instances) and may participate in various examinations: state space generation, computation of global properties, computation of 16 queries with regards to upper bounds in the model, evaluation of 16 reachability formulas, evaluation of 16 CTL formulas, and evaluation of 16 LTL formulas.

For each examination and each model instance, participating tools are provided with up to 3600 seconds of runtime and 16 GB of memory. Tool answers are analyzed and confronted to the results produced by other competing tools to detect diverging answers (which are quite rare at this stage of the competition, and lead to penalties).

[16] https://gitlab.inria.fr/crv14/benchmarks

For each examination, golden, silver, and bronze medals are attributed to the three best tools. CPU usage and memory consumption are reported, which is also valuable information for tool developers. Finally, numerous charts to compare pair of tools' performances, or quantile plots stating global performances are computed. Performances of tools on models (useful when they contain scaling parameters) are also provided.

2.6 QComp: The Comparison of Tools for the Analysis of Quantitative Formal Models

Organizers: Arnd Hartmanns (Univ. of Twente, Netherlands) and Tim Quatmann (RWTH Aachen Univ., Germany),
Webpage: http://qcomp.org

Quantitative formal models capture probabilistic behaviour, real-time aspects, or general continuous dynamics. A number of tools support their automatic analysis with respect to dependability or performance properties. QComp 2019 is the first competition among such tools. It focuses on stochastic formalisms from Markov chains to probabilistic timed automata specified in the JANI model exchange format, and on probabilistic reachability, expected-reward, and steady-state properties. QComp draws its benchmarks from the new Quantitative Verification Benchmark Set. Participating tools, which include probabilistic model checkers and planners as well as simulation-based tools, are evaluated in terms of performance, versatility, and usability.

2.7 REC: The Rewrite Engines Competition

Organizers: Francisco Durán (Univ. of Malaga, Spain) and Hubert Garavel (Univ. Grenoble Alpes/INRIA/CNRS, Grenoble INP/LIG, France)
Webpage: http://rec.gforge.inria.fr/

Term rewriting is a simple, yet expressive model of computation, which finds direct applications in specification and programming languages (many of which embody rewrite rules, pattern matching, and abstract data types), but also indirect applications, e.g., to express the semantics of data types or concurrent processes, to specify program transformations, to perform computer-aided verification. The Rewrite Engines Competition (REC) was created under the aegis of the Workshop on Rewriting Logic and its Applications (WRLA) to serve three main goals:

1. being a forum in which tool developers and potential users of term rewrite engines can share experience;
2. bringing together the various language features and implementation techniques used for term rewriting; and
3. comparing the available term rewriting languages and tools in their common features.

Earlier editions of the Rewrite Engines Competition have been held in 2006, 2008, 2010, and 2018.

2.8 RERS: Rigorous Examination of Reactive System

Organizers: Falk Howar (TU Dortmund, Germany), Markus Schordan (LLNL, USA), Bernhard Steffen (TU Dortmund, Germany), and Jaco van de Pol (Univ. of Aarhus, Denmark)
Webpage: http://rers-challenge.org/

Reactive systems appear everywhere, e.g., as Web services, decision support systems, or logical controllers. Their validation techniques are as diverse as their appearance and structure. They comprise various forms of static analysis, model checking, symbolic execution, and (model-based) testing, often tailored to quite extreme frame conditions. Thus it is almost impossible to compare these techniques, let alone to establish clear application profiles as a means for recommendation. Since 2010, the RERS Challenge aims at overcoming this situation by providing a forum for experimental profile evaluation based on specifically designed benchmark suites.

These benchmarks are automatically synthesized to exhibit chosen properties, and then enhanced to include dedicated dimensions of difficulty, ranging from conceptual complexity of the properties (e.g., reachability, full safety, liveness), over size of the reactive systems (a few hundred lines to millions of them), to exploited language features (arrays, arithmetic at index pointer, and parallelism). The general approach has been described in [88, 89], while variants to introduce highly parallel benchmarks are discussed in [86, 87, 90]. RERS benchmarks have been used also by other competitions, like MCC or SV-COMP, and referenced in a number of research papers as a means of evaluation not only in the context of RERS [30, 60, 74, 76, 79, 82].

In contrast to the other competitions described in this paper, RERS is problem oriented and does not evaluate the power of specific tools but rather tool usage that ideally makes use of a number of tools and methods. This is meant to help leveraging the synergy potential also between seemingly quite separate technologies like, e.g., source-code-based (white-box) approaches and purely observation/testing-based (black-box) approaches. The most convincing heterogeneous approach is awarded the RERS Methods Combination Award.

2.9 Rodeo for Production Software Verification Tools Based on Formal Methods

Organizer: Paul E. Black (NIST, USA)
Webpage: https://samate.nist.gov/FMSwVRodeo/

Formal methods are not widely used in the United States. The US government is now more interested because of the wide variety of FM-based tools that can handle production-sized software and because algorithms are orders of magnitude faster. NIST proposes to select production software for a test suite and to hold a periodic Rodeo to assess the effectiveness of tools based on formal methods that can verify large, complex software. To select software, we will develop

tools to measure structural characteristics, like depth of recursion or number of states, and calibrate them on others' benchmarks. We can then scan thousands of applications to select software for the Rodeo.

2.10 SAT

Organizer: Marijn Heule (Univ. of Texas at Austin, USA), Matti Järvisalo (Univ. of Helsinki, Finland), and Martin Suda (Czech Technical Univ., Czechia)
Webpage: https://www.satcompetition.org/

SAT Competition 2018 is the twelfth edition of the SAT Competition series, continuing the almost two decades of tradition in SAT competitions and related competitive events for Boolean Satisfiability (SAT) solvers. It was organized as part of the 2018 FLoC Olympic Games in conjunction with the 21^{th} International Conference on Theory and Applications of Satisfiability Testing (SAT 2018), which took place in Oxford, UK, as part of the 2018 Federated Logic Conference (FLoC). The competition consisted of four tracks, including a main track, a "no-limits" track with very few requirements for participation, and special tracks focusing on random SAT and parallel solving. In addition to the actual solvers, each participant was required to also submit a collection of previously unseen benchmark instances, which allowed the competition to only use new benchmarks for evaluation. Where applicable, verifiable certificates were required both for the "satisfiable" and "unsatisfiable" answers; the general time limit was 5000 s per benchmark instance and the solvers were ranked using the PAR-2 scheme, which encourages solving many benchmarks but also rewards solving the benchmarks fast. A detailed overview of the competition, including summary of the results, will appear in the JSAT special issue on SAT 2018 Competitions and Evaluations.

2.11 SL-COMP: Competition of Solvers for Separation Logic

Organizer: Mihaela Sighireanu (Univ. of Paris Diderot, France)
Webpage: https://sl-comp.github.io/

SL-COMP aims at bringing together researchers interested in improving the state of the art of automated deduction methods for Separation Logic (SL). The event took place twice until now and collected more than 1K problems for different fragments of SL. The input format of problems is based on the SMT-LIB format and therefore fully typed; only one new command is added to SMT-LIB's list, the command for the declaration of the heap's type. The SMT-LIB theory of SL comes with ten logics, some of them being combinations of SL with linear arithmetic. The competition's divisions are defined by the logic fragment, the kind of decision problem (satisfiability or entailment), and the presence of quantifiers. Until now, SL-COMP has been run on the StarExec platform, where the benchmark set and the binaries of participant solvers are freely available. The benchmark set is also available with the competition's documentation on a public repository in GitHub.

2.12 SMT-COMP

Organizer: Matthias Heizmann (Univ. of Freiburg, Germany), Aina Niemetz (Stanford Univ., USA), Giles Reger (Univ. of Manchester, UK), and Tjark Weber (Uppsala Univ., Sweden)
Webpage: http://www.smtcomp.org

Satisfiability Modulo Theories (SMT) is a generalization of the satisfiability decision problem for propositional logic. In place of Boolean variables, SMT formulas may contain terms that are built from function and predicate symbols drawn from a number of background theories, such as arrays, integer and real arithmetic, or bit-vectors. With its rich input language, SMT has applications in software engineering, optimization, and many other areas.

The International Satisfiability Modulo Theories Competition (SMT-COMP) is an annual competition between SMT solvers. It was instituted in 2005, and is affiliated with the International Workshop on Satisfiability Modulo Theories. Solvers are submitted to the competition by their developers, and compete against each other in a number of tracks and divisions. The main goals of the competition are to promote the community-designed SMT-LIB format, to spark further advances in SMT, and to provide a useful yardstick of performance for users and developers of SMT solvers.

2.13 SV-COMP: Competition on Software Verification

Organizer: Dirk Beyer (LMU Munich, Germany)
Webpage: https://sv-comp.sosy-lab.org/

The 2019 International Competition on Software Verification (SV-COMP) is the 8^{th} edition in a series of annual comparative evaluations of fully-automatic tools for software verification. The competition was established and first executed in 2011 and the first results were presented and published at TACAS 2012 [16]. The most important goals of the competition are the following:

1. Provide an overview of the state of the art in software-verification technology and increase visibility of the most recent software verifiers.
2. Establish a repository of software-verification tasks that is publicly available for free as standard benchmark suite for evaluating verification software [17].
3. Establish standards that make it possible to compare different verification tools, including a property language and formats for the results, especially witnesses.
4. Accelerate the transfer of new verification technology to industrial practice.

The benchmark suite for SV-COMP 2019 [22] consists of nine categories with a total of 10 522 verification tasks in C and 368 verification tasks in Java. A verification task (benchmark instance) in SV-COMP is a pair of a program M and

[17] https://github.com/sosy-lab/sv-benchmarks

a property ϕ, and the task for the solver (here: verifier) is to verify the statement $M \models \phi$, that is, the benchmarked verifier should return FALSE and a violation witness that describes a property violation [25, 29], or TRUE and a correctness witness that contains invariants to re-establish the correctness proof [24]. The ranking is computed according to a scoring schema that assigns a positive score (1 and 2) to correct results and a negative score (-16 and -32) to incorrect results, for tasks with and without property violations, respectively. The sum of CPU time of the successfully solved verification tasks is the tie-breaker if two verifiers have the same score. The results are also illustrated using quantile plots. [18]

The 2019 competition attracted 31 participating teams from 14 countries. This competition included Java verification for the first time, and this track had four participating verifiers. As before, the large jury (one representative of each participating team) and the organizer made sure that the competition follows high quality standards and is driven by the four important principles of (1) *fairness*, (2) *community support*, (3) *transparency*, and (4) *technical accuracy*.

2.14 termComp: The Termination and Complexity Competition

Organizer: Akihisa Yamada (National Institute of Informatics, Japan)
Steering Committee: Jürgen Giesl (RWTH Aachen Univ., Germany), Albert Rubio (Univ. Politècnica de Catalunya, Spain), Christian Sternagel (Univ. of Innsbruck, Austria), Johannes Waldmann (HTWK Leipzig, Germany), and Akihisa Yamada (National Institute of Informatics, Japan)
Webpage: http://termination-portal.org/wiki/Termination_Competition

The termination and complexity competition (termCOMP) focuses on automated termination and complexity analysis for various kinds of programming paradigms, including categories for term rewriting, integer transition systems, imperative programming, logic programming, and functional programming. It has been organized annually after a tool demonstration in 2003. In all categories, the competition also welcomes the participation of tools providing certifiable output. The goal of the competition is to demonstrate the power and advances of the state-of-the-art tools in each of these areas.

2.15 Test-Comp: Competition on Software Testing

Organizer: Dirk Beyer (LMU Munich, Germany)
Webpage: https://test-comp.sosy-lab.org/

The 2019 International Competition on Software Testing (Test-Comp) is the 1^{st} edition of a series of annual comparative evaluations of fully-automatic tools for software testing. The design of Test-Comp is very similar to the design of SV-COMP, with the major difference that the task for the solver (here: tester)

[18] https://sv-comp.sosy-lab.org/2019/results/

is to generate a test suite, which is validated against a coverage property, that is, the ranking is based on the coverage that the resulting test-suites achieve.

There are several new and powerful tools for automatic software testing around, but they were difficult to compare before the competition [27]. The reason had been that so far no established benchmark suite of test tasks was available and many concepts were only validated in research prototypes. Now the test-case generators support a standardized input format (for C programs as well as for coverage properties). The overall goals of the competition are:

- Provide a snapshot of the state-of-the-art in software testing to the community. This means to compare, independently from particular paper projects and specific techniques, different test-generation tools in terms of precision and performance.
- Increase the visibility and credits that tool developers receive. This means to provide a forum for presentation of tools and discussion of the latest technologies, and to give the students the opportunity to publish about the development work that they have done.
- Establish a set of benchmarks for software testing in the community. This means to create and maintain a set of programs together with coverage criteria, and to make those publicly available for researchers to be used free of charge in performance comparisons when evaluating a new technique.

2.16 VerifyThis

Organizers 2019: Carlo A. Furia (Univ. della Svizzera Italiana, Switzerland) and Claire Dross (AdaCore, France)
Steering Committee: Marieke Huisman (Univ. of Twente, Netherlands), Rosemary Monahan (National Univ. of Ireland at Maynooth, Ireland), and Peter Müller (ETH Zurich, Switzerland)
Webpage: http://www.pm.inf.ethz.ch/research/verifythis.html

The aims of the VerifyThis competition are:

- to bring together those interested in formal verification,
- to provide an engaging, hands-on, and fun opportunity for discussion, and
- to evaluate the usability of logic-based program verification tools in a controlled experiment that could be easily repeated by others.

The competition offers a number of challenges presented in natural language and pseudo code. Participants have to formalize the requirements, implement a solution, and formally verify the implementation for adherence to the specification.

There are no restrictions on the programming language and verification technology used. The correctness properties posed in problems will have the input-output behaviour of programs in focus. Solutions will be judged for correctness, completeness, and elegance.

VerifyThis is an annual event. Earlier editions were held at FoVeOos (2011), FM (2012), and since 2015 annually at ETAPS.

3 On the Future of Competitions

In this paper, we have provided an overview of the wide spectrum of different competitions and challenges. Each competition can be distinguished by its specific problem profile, characterized by analysis goals, resource and infrastructural constraints, application areas, and dedicated methodologies. Despite their differences, these competitions and challenges also have many similar concerns, related to, e.g., (1) benchmark selection, maintenance, and archiving, (2) evaluation and rating strategies, (3) publication and replicability of results, as well as (4) licensing issues.

TOOLympics aimes at leveraging the potential synergy by supporting a dialogue between competition organizers about all relevant issues. Besides increasing the mutual awareness about shared concerns, this also comprises:

- the potential exchange of benchmarks (ideally supported by dedicated interchange formats), e.g., from high-level competitions like VerifyThis, SV-COMP, and RERS to more low-level competitions like SMT-COMP, CASC, or the SAT competition,
- the detection of new competition formats or the aggregation of existing competition formats to establish a better coverage of verification problem areas in a complementary fashion, and
- the exchange of ideas to motivate new participants, e.g., by lowering the entrance hurdle.

There have been a number of related initiatives with the goal of increasing awareness for the scientific method of evaluating tools in a *competition*-based fashion, like the COMPARE workshop on Comparative Empirical Evaluation of Reasoning Systems [61], the Dagstuhl seminar on Evaluating Software Verification Systems in 2014 [26], the FLoC Olympics Games 2014 [19] and 2018 [20], and the recent Lorentz Workshop on Advancing Verification Competitions as a Scientific Method [21]. TOOLympics aims at joining forces with all these initiatives in order to establish a comprehensive hub where tool developers, users, participants, and organizers may meet and discuss current issues, share experiences, compose benchmark libraries (ideally classified in a way that supports cross competition usage), and develop ideas for future directions of competitions.

Finally, it is important to note that competitions have resulted in significant progress in the research areas that they belong to, repectively. Typically, new techniques and theories have been developed, and tools have become much stronger and more mature. This sometimes means that a disruption in the way that the competitions are handled is needed, in order to adapt the competition to these evolutions. It is our hope that platforms such as TOOLympics facilitate and improve this process.

[19] https://vsl2014.at/olympics/
[20] https://www.floc2018.org/floc-olympic-games/
[21] https://www.lorentzcenter.nl/lc/web/2019/1091/info.php3?wsid=1091

References

1. Abate, A., Blom, H., Cauchi, N., Haesaert, S., Hartmanns, A., Lesser, K., Oishi, M., Sivaramakrishnan, V., Soudjani, S., Vasile, C.I., Vinod, A.P.: Arch-comp18 category report: Stochastic modelling. ARCH18. 5th International Workshop on Applied Verification of Continuous and Hybrid Systems **54**, 71 – 103 (2018), https://easychair.org/publications/open/DzD8
2. Aoto, T., Hamana, M., Hirokawa, N., Middeldorp, A., Nagele, J., Nishida, N., Shintani, K., Zankl, H.: Confluence Competition 2018. In: Proc. 3rd International Conference on Formal Structures for Computation and Deduction (FSCD 2018). Leibniz International Proceedings in Informatics (LIPIcs), vol. 108, pp. 32:1–32:5. Schloss Dagstuhl–Leibniz-Zentrum fuer Informatik (2018), https://doi.org/10.4230/LIPIcs.FSCD.2018.32
3. Aoto, T., Hirokawa, N., Nagele, J., Nishida, N., Zankl, H.: Confluence Competition 2015. In: Proc. 25th International Conference on Automated Deduction (CADE-25). Lecture Notes in Artificial Intelligence, vol. 9195, pp. 101–104. Springer (2015), https://doi.org/10.1007/978-3-319-21401-6_5
4. Balint, A., Belov, A., Järvisalo, M., Sinz, C.: Overview and analysis of the SAT Challenge 2012 solver competition. Artificial Intelligence **223**, 120–155 (2015), https://doi.org/10.1016/j.artint.2015.01.002
5. Balyo, T., Heule, M.J.H., Järvisalo, M.: SAT Competition 2016: Recent developments. In: Singh, S.P., Markovitch, S. (eds.) Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence, February 4-9, 2017, San Francisco, California, USA. pp. 5061–5063. AAAI Press (2017)
6. Barrett, C., Deters, M., de Moura, L., Oliveras, A., Stump, A.: 6 years of SMT-COMP. Journal of Automated Reasoning **50**(3), 243–277 (2013), https://doi.org/10.1007/s10817-012-9246-5
7. Barrett, C., Deters, M., Oliveras, A., Stump, A.: Design and results of the 3rd Annual Satisfiability Modulo Theories Competition (SMT-COMP 2007). International Journal on Artificial Intelligence Tools **17**(4), 569–606 (2008)
8. Barrett, C., Deters, M., Oliveras, A., Stump, A.: Design and results of the 4th Annual Satisfiability Modulo Theories Competition (SMT-COMP 2008). Tech. Rep. TR2010-931, New York University (2010)
9. Barrett, C., de Moura, L., Stump, A.: Design and results of the 1st Satisfiability Modulo Theories Competition (SMT-COMP 2005). Journal of Automated Reasoning **35**(4), 373–390 (2005)
10. Barrett, C., de Moura, L., Stump, A.: Design and results of the 2nd Annual Satisfiability Modulo Theories Competition (SMT-COMP 2006). Formal Methods in System Design (2007)
11. Bartocci, E., Bonakdarpour, B., Falcone, Y.: First international competition on software for runtime verification. In: Bonakdarpour, B., Smolka, S.A. (eds.) Proc. of RV 2014: the 5th International Conference on Runtime Verification. LNCS, vol. 8734, pp. 1–9. Springer (2014), https://doi.org/10.1007/978-3-319-11164-3_1
12. Bartocci, E., Falcone, Y., Bonakdarpour, B., Colombo, C., Decker, N., Havelund, K., Joshi, Y., Klaedtke, F., Milewicz, R., Reger, G., Rosu, G., Signoles, J., Thoma, D., Zalinescu, E., Zhang, Y.: First International Competition on Runtime Verification: rules, benchmarks, tools, and final results of CRV 2014. International Journal on Software Tools for Technology Transfer **21**, 31–70 (Apr 2019), https://doi.org/10.1007/s10009-017-0454-5

13. Bartocci, E., Falcone, Y., Reger, G.: International competition on runtime verification (CRV). In: Proc. TACAS, part 3. LNCS this volume, Springer (2019)
14. Berre, D.L., Simon, L.: The essentials of the SAT 2003 Competition. In: Giunchiglia, E., Tacchella, A. (eds.) Theory and Applications of Satisfiability Testing, 6th International Conference, SAT 2003. Santa Margherita Ligure, Italy, May 5-8, 2003 Selected Revised Papers. Lecture Notes in Computer Science, vol. 2919, pp. 452–467. Springer (2004)
15. Berre, D.L., Simon, L.: Fifty-five solvers in Vancouver: The SAT 2004 Competition. In: Hoos, H.H., Mitchell, D.G. (eds.) Theory and Applications of Satisfiability Testing, 7th International Conference, SAT 2004, Vancouver, BC, Canada, May 10-13, 2004, Revised Selected Papers. Lecture Notes in Computer Science, vol. 3542, pp. 321–344. Springer (2005)
16. Beyer, D.: Competition on software verification (SV-COMP). In: Proc. TACAS. pp. 504–524. LNCS 7214, Springer (2012), https://doi.org/10.1007/978-3-642-28756-5_38
17. Beyer, D.: Second competition on software verification (Summary of SV-COMP 2013). In: Proc. TACAS. pp. 594–609. LNCS 7795, Springer (2013), https://doi.org/10.1007/978-3-642-36742-7_43
18. Beyer, D.: Status report on software verification (Competition summary SV-COMP 2014). In: Proc. TACAS. pp. 373–388. LNCS 8413, Springer (2014), https://doi.org/10.1007/978-3-642-54862-8_25
19. Beyer, D.: Software verification and verifiable witnesses (Report on SV-COMP 2015). In: Proc. TACAS. pp. 401–416. LNCS 9035, Springer (2015), https://doi.org/10.1007/978-3-662-46681-0_31
20. Beyer, D.: Reliable and reproducible competition results with BenchExec and witnesses (Report on SV-COMP 2016). In: Proc. TACAS. pp. 887–904. LNCS 9636, Springer (2016), https://doi.org/10.1007/978-3-662-49674-9_55
21. Beyer, D.: Software verification with validation of results (Report on SV-COMP 2017). In: Proc. TACAS. pp. 331–349. LNCS 10206, Springer (2017), https://doi.org/10.1007/978-3-662-54580-5_20
22. Beyer, D.: Automatic verification of C and Java programs: SV-COMP 2019. In: Proc. TACAS, part 3. LNCS this volume, Springer (2019)
23. Beyer, D.: Competition on software testing (Test-Comp). In: Proc. TACAS, part 3. LNCS this volume, Springer (2019)
24. Beyer, D., Dangl, M., Dietsch, D., Heizmann, M.: Correctness witnesses: Exchanging verification results between verifiers. In: Proc. FSE. pp. 326–337. ACM (2016), https://doi.org/10.1145/2950290.2950351
25. Beyer, D., Dangl, M., Dietsch, D., Heizmann, M., Stahlbauer, A.: Witness validation and stepwise testification across software verifiers. In: Proc. FSE. pp. 721–733. ACM (2015), https://doi.org/10.1145/2786805.2786867
26. Beyer, D., Huisman, M., Klebanov, V., Monahan, R.: Evaluating software verification systems: Benchmarks and competitions (Dagstuhl reports 14171). Dagstuhl Reports **4**(4), 1–19 (2014), https://doi.org/10.4230/DagRep.4.4.1
27. Beyer, D., Lemberger, T.: Software verification: Testing vs. model checking. In: Proc. HVC. pp. 99–114. LNCS 10629, Springer (2017), https://doi.org/10.1007/978-3-319-70389-3_7
28. Beyer, D., Löwe, S., Wendler, P.: Reliable benchmarking: Requirements and solutions. Int. J. Softw. Tools Technol. Transfer (2017), https://doi.org/10.1007/s10009-017-0469-y

29. Beyer, D., Wendler, P.: Reuse of verification results: Conditional model checking, precision reuse, and verification witnesses. In: Proc. SPIN. pp. 1–17. LNCS 7976, Springer (2013), https://doi.org/10.1007/978-3-642-39176-7_1
30. Beyer, D., Stahlbauer, A.: BDD-based software verification. International Journal on Software Tools for Technology Transfer **16**(5), 507–518 (Oct 2014)
31. Bormer, T., Brockschmidt, M., Distefano, D., Ernst, G., Filliâtre, J.C., Grigore, R., Huisman, M., Klebanov, V., Marché, C., Monahan, R., Mostowski, W., Polikarpova, N., Scheben, C., Schellhorn, G., Tofan, B., Tschannen, J., Ulbrich, M.: The COST IC0701 verification competition 2011. In: Beckert, B., Damiani, F., Gurov, D. (eds.) International Conference on Formal Verification of Object-Oriented Systems (FoVeOOS 2011). LNCS, vol. 7421, pp. 3–21. Springer (2011)
32. Cok, D.R., Déharbe, D., Weber, T.: The 2014 SMT competition. Journal on Satisfiability, Boolean Modeling and Computation **9**, 207–242 (2014), https://satassociation.org/jsat/index.php/jsat/article/view/122
33. Cok, D.R., Griggio, A., Bruttomesso, R., Deters, M.: The 2012 SMT competition (2012), available online at http://smtcomp.sourceforge.net/2012/reports/SMTCOMP2012.pdf
34. Cok, D.R., Stump, A., Weber, T.: The 2013 evaluation of SMT-COMP and SMT-LIB. Journal of Automated Reasoning **55**(1), 61–90 (2015), https://doi.org/10.1007/s10817-015-9328-2
35. Denker, G., Talcott, C.L., Rosu, G., van den Brand, M., Eker, S., Serbanuta, T.F.: Rewriting logic systems. Electronic Notes in Theoretical Computer Science **176**(4), 233–247 (2007), https://doi.org/10.1016/j.entcs.2007.06.018
36. Durán, F., Garavel, H.: The Rewrite Engines Competitions: A RECtrospective. In: Proc. TACAS, part 3. LNCS this volume, Springer (2019)
37. Durán, F., Roldán, M., Bach, J.C., Balland, E., van den Brand, M., Cordy, J.R., Eker, S., Engelen, L., de Jonge, M., Kalleberg, K.T., Kats, L.C.L., Moreau, P.E., Visser, E.: The third Rewrite Engines Competition. In: Ölveczky, P.C. (ed.) Proceedings of the 8th International Workshop on Rewriting Logic and Its Applications (WRLA'10), Paphos, Cyprus. LNCS, vol. 6381, pp. 243–261. Springer (2010), https://doi.org/10.1007/978-3-642-16310-4_16
38. Durán, F., Roldán, M., Balland, E., van den Brand, M., Eker, S., Kalleberg, K.T., Kats, L.C.L., Moreau, P.E., Schevchenko, R., Visser, E.: The second Rewrite Engines Competition. Electronic Notes in Theoretical Computer Science **238**(3), 281–291 (2009), https://doi.org/10.1016/j.entcs.2009.05.025
39. Ernst, G., Huisman, M., Mostowski, W., Ulbrich, M.: Verifythis — verification competition with a human factor. In: Proc. TACAS, part 3. LNCS this volume, Springer (2019)
40. Falcone, Y., Nickovic, D., Reger, G., Thoma, D.: Second international competition on runtime verification CRV 2015. In: Proc. of RV 2015: the 6th International Conference on Runtime Verification. LNCS, vol. 9333, pp. 405–422. Springer (2015), https://doi.org/10.1007/978-3-319-23820-3
41. Garavel, H., Tabikh, M.A., Arrada, I.S.: Benchmarking implementations of term rewriting and pattern matching in algebraic, functional, and object-oriented languages – The 4th Rewrite Engines Competition. In: Rusu, V. (ed.) Proceedings of the 12th International Workshop on Rewriting Logic and its Applications (WRLA'18), Thessaloniki, Greece. LNCS, vol. 11152, pp. 1–25. Springer (Apr 2018), https://doi.org/10.1007/978-3-319-99840-4_1
42. Geske, M., Isberner, M., Steffen, B.: Rigorous examination of reactive systems:. In: Bartocci, E., Majumdar, R. (eds.) Runtime Verification (2015)

43. Geske, M., Jasper, M., Steffen, B., Howar, F., Schordan, M., van de Pol, J.: RERS 2016: Parallel and sequential benchmarks with focus on LTL verification. In: ISoLA. LNCS, vol 9953. pp. 787–803. Springer (2016)
44. Giesl, J., Mesnard, F., Rubio, A., Thiemann, R., Waldmann, J.: Termination competition (termCOMP 2015). In: Felty, A., Middeldorp, A. (eds.) CADE-25. LNCS 9195, Springer (2015), https://doi.org/10.1007/978-3-319-21401-6_6
45. Giesl, J., Rubio, A., Sternagel, C., Waldmann, J., Yamada, A.: The termination and complexity competition. In: Proc. TACAS, part 3. LNCS this volume, Springer (2019)
46. Hahn, E.M., Hartmanns, A., Hensel, C., Klauck, M., Klein, J., Křetínský, J., Parker, D., Quatmann, T., Ruijters, E., Steinmetz, M.: The 2019 comparison of tools for the analysis of quantitative formal models. In: Proc. TACAS, part 3. LNCS this volume, Springer (2019)
47. Howar, F., Isberner, M., Merten, M., Steffen, B., Beyer, D.: The RERS grey-box challenge 2012: Analysis of event-condition-action systems. In: Proc. ISoLA. pp. 608–614. LNCS 7609, Springer (2012)
48. Howar, F., Isberner, M., Merten, M., Steffen, B., Beyer, D., Păsăreanu, C.: Rigorous examination of reactive systems. The RERS challenges 2012 and 2013. STTT **16**(5), 457–464 (2014)
49. Howar, F., Steffen, B., Merten, M.: From ZULU to RERS. In: Margaria, T., Steffen, B. (eds.) Leveraging Applications of Formal Methods, Verification, and Validation. pp. 687–704. Springer Berlin Heidelberg, Berlin, Heidelberg (2010)
50. Huisman, M., Klebanov, V., Monahan, R.: VerifyThis verification competition 2012 – organizer's report. Tech. Rep. 2013-01, Department of Informatics, Karlsruhe Institute of Technology (2013), available at http://digbib.ubka.uni-karlsruhe.de/volltexte/1000034373
51. Huisman, M., Monahan, R., Mostowski, W., Müller, P., Ulbrich, M.: VerifyThis 2017: A program verification competition. Tech. rep., Karlsruhe Reports in Informatics (2017)
52. Huisman, M., Monahan, R., Müller, P., Paskevich, A., Ernst, G.: VerifyThis 2018: A program verification competition. Tech. rep., Inria (2019)
53. Huisman, M., Monahan, R., Müller, P., Poll, E.: VerifyThis 2016: A program verification competition. Tech. Rep. TR-CTIT-16-07, Centre for Telematics and Information Technology, University of Twente, Enschede (2016)
54. Huisman, M., Klebanov, V., Monahan, R.: VerifyThis 2012. Int. J. Softw. Tools Technol. Transf. **17**(6), 647–657 (Nov 2015)
55. Huisman, M., Klebanov, V., Monahan, R., Tautschnig, M.: VerifyThis 2015. A program verification competition. Int. J. Softw. Tools Technol. Transf. **19**(6), 763–771 (2017)
56. Jacobs, S., Bloem, R., Brenguier, R., Ehlers, R., Hell, T., Könighofer, R., Pérez, G.A., Raskin, J., Ryzhyk, L., Sankur, O., Seidl, M., Tentrup, L., Walker, A.: The first reactive synthesis competition (SYNTCOMP 2014). STTT **19**(3), 367–390 (2017). https://doi.org/10.1007/s10009-016-0416-3, https://doi.org/10.1007/s10009-016-0416-3
57. Jasper, M., Fecke, M., Steffen, B., Schordan, M., Meijer, J., Pol, J.v.d., Howar, F., Siegel, S.F.: The RERS 2017 Challenge and Workshop (invited paper). In: Proceedings of the 24th ACM SIGSOFT International SPIN Symposium on Model Checking of Software. pp. 11–20. SPIN 2017, ACM (2017)
58. Jasper, M., Mues, M., Murtovi, A., Schlüter, M., Howar, F., Bernhard Steffen, M.S., Hendriks, D., Schiffelers, R., Kuppens, H., Vaandrager, F.: RERS 2019:

Combining synthesis with real-world models. In: Proc. TACAS, part 3. LNCS this volume, Springer (2019)
59. Jasper, M., Mues, M., Schlüter, M., Steffen, B., Howar, F.: RERS 2018: CTL, LTL, and reachability. In: ISoLA'18. LNCS, vol 11245. pp. 433–447. Springer International Publishing (2018)
60. Kant, G., Laarman, A., Meijer, J., van de Pol, J., Blom, S., van Dijk, T.: LTSmin: High-performance language-independent model checking. In: Baier, C., Tinelli, C. (eds.) Tools and Algorithms for the Construction and Analysis of Systems (2015)
61. Klebanov, V., Beckert, B., Biere, A., Sutcliffe, G. (eds.): Proceedings of the 1st International Workshop on Comparative Empirical Evaluation of Reasoning Systems, Manchester, United Kingdom, June 30, 2012, CEUR Workshop Proceedings, vol. 873. CEUR-WS.org (2012), http://ceur-ws.org/Vol-873
62. Kordon, F., Garavel, H., Hillah, L.M., Hulin-Hubard, F., Amparore, E., Beccuti, M., Berthomieu, B., Ciardo, G., Dal Zilio, S., Liebke, T., Linard, A., Meijer, J., Miner, A., Srba, J., Thierry-Mieg, J., van de Pol, J., Wolf, K.: Complete Results for the 2018 Edition of the Model Checking Contest. http://mcc.lip6.fr/2018/results.php (June 2018)
63. Kordon, F., Garavel, H., Hillah, L.M., Hulin-Hubard, F., Berthomieu, B., Ciardo, G., Colange, M., Dal Zilio, S., Amparore, E., Beccuti, M., Liebke, T., Meijer, J., Miner, A., Rohr, C., Srba, J., Thierry-Mieg, Y., van de Pol, J., Wolf, K.: Complete Results for the 2017 Edition of the Model Checking Contest. http://mcc.lip6.fr/2017/results.php (June 2017)
64. Kordon, F., Garavel, H., Hillah, L.M., Hulin-Hubard, F., Chiardo, G., Hamez, A., Jezequel, L., Miner, A., Meijer, J., Paviot-Adet, E., Racordon, D., Rodriguez, C., Rohr, C., Srba, J., Thierry-Mieg, Y., Trịnh, G., Wolf, K.: Complete Results for the 2016 Edition of the Model Checking Contest. http://mcc.lip6.fr/2016/results.php (June 2016)
65. Kordon, F., Garavel, H., Hillah, L.M., Hulin-Hubard, F., Linard, A., Beccuti, M., Evangelista, S., Hamez, A., Lohmann, N., Lopez, E., Paviot-Adet, E., Rodriguez, C., Rohr, C., Srba, J.: HTML results from the Model Checking Contest @ Petri Net (2014 edition). http://mcc.lip6.fr/2014 (2014)
66. Kordon, F., Garavel, H., Hillah, L.M., Hulin-Hubard, F., Linard, A., Beccuti, M., Hamez, A., Lopez-Bobeda, E., Jezequel, L., Meijer, J., Paviot-Adet, E., Rodriguez, C., Rohr, C., Srba, J., Thierry-Mieg, Y., Wolf, K.: Complete Results for the 2015 Edition of the Model Checking Contest. http://mcc.lip6.fr/2015/results.php (2015)
67. Kordon, F., Hulin-Hubard, F.: BenchKit, a tool for massive concurrent benchmarking. In: Proc. ACSD. pp. 159–165. IEEE (2014), https://doi.org/10.1109/ACSD.2014.12
68. Kordon, F., Linard, A., Buchs, D., Colange, M., Evangelista, S., Lampka, K., Lohmann, N., Paviot-Adet, E., Thierry-Mieg, Y., Wimmel, H.: Report on the Model Checking Contest at Petri Nets 2011. Transactions on Petri Nets and Other Models of Concurrency (ToPNoC) **VI**, 169–196 (march 2012)
69. Kordon, F., Garavel, H., Hillah, L.M., Hulin-Hubard, F., Jezequel, L., Paviot-Adet, E.: The model checking contest (2019). In: Proc. TACAS, part 3. LNCS this volume, Springer (2019)
70. Kordon, F., Linard, A., Beccuti, M., Buchs, D., Fronc, L., Hillah, L., Hulin-Hubard, F., Legond-Aubry, F., Lohmann, N., Marechal, A., Paviot-Adet, E., Pommereau, F., Rodríguez, C., Rohr, C., Thierry-Mieg, Wimmel, H., Wolf, K.: Model checking contest @ Petri Nets, report on the 2013 edition. CoRR **abs/1309.2485** (2013), http://arxiv.org/abs/1309.2485

71. Kordon, F., Linard, A., Buchs, D., Colange, M., Evangelista, Fronc, L., Hillah, L.M., Lohmann, N., Paviot-Adet, E., Pommereau, F., Rohr, C., Thierry-Mieg, Y., Wimmel, H., Wolf, K.: Raw report on the model checking contest at Petri Nets 2012. CoRR **abs/1209.2382** (2012), http://arxiv.org/abs/1209.2382
72. Lonsing, F., Seidl, M., Gelder, A.V.: The QBF gallery: Behind the scenes. Artif. Intell. **237**, 92–114 (2016). https://doi.org/10.1016/j.artint.2016.04.002, https://doi.org/10.1016/j.artint.2016.04.002
73. Marché, C., Zantema, H.: The termination competition. In: Baader, F. (ed.) Proc. RTA. LNCS 4533, Springer (2007), https://doi.org/10.1007/978-3-540-73449-9_23
74. Meijer, J., van de Pol, J.: Sound black-box checking in the LearnLib. In: Dutle, A., Muñoz, C., Narkawicz, A. (eds.) NASA Formal Methods. Springer International Publishing (2018)
75. Middeldorp, A., Nagele, J., Shintani, K.: Confluence competion 2019. In: Proc. TACAS, part 3. LNCS this volume, Springer (2019)
76. Morse, J., Cordeiro, L., Nicole, D., Fischer, B.: Applying symbolic bounded model checking to the 2012 rers greybox challenge. International Journal on Software Tools for Technology Transfer **16**(5), 519–529 (Oct 2014)
77. Nieuwenhuis, R.: The Impact of CASC in the Development of Automated Deduction Systems. AI Communications **15**(2-3), 77–78 (2002)
78. Pelletier, F., Sutcliffe, G., Suttner, C.: The Development of CASC. AI Communications **15**(2-3), 79–90 (2002)
79. van de Pol, J., Ruys, T.C., te Brinke, S.: Thoughtful brute-force attack of the RERS 2012 and 2013 challenges. International Journal on Software Tools for Technology Transfer **16**(5), 481–491 (Oct 2014)
80. Reger, G., Hallé, S., Falcone, Y.: Third international competition on runtime verification - CRV 2016. In: Proc. of RV 2016: the 16th International Conference on Runtime Verification. LNCS, vol. 10012, pp. 21–37. Springer (2016), https://doi.org/10.1007/978-3-319-46982-9
81. Reger, G., Havelund, K. (eds.): RV-CuBES 2017. An International Workshop on Competitions, Usability, Benchmarks, Evaluation, and Standardisation for Runtime Verification Tools, Kalpa Publications in Computing, vol. 3. EasyChair (2017)
82. Schordan, M., Prantl, A.: Combining static analysis and state transition graphs for verification of event-condition-action systems in the RERS 2012 and 2013 challenges. International Journal on Software Tools for Technology Transfer **16**(5), 493–505 (Oct 2014)
83. Sighireanu, M., Cok, D.: Report on SL-COMP'14. JSAT **9**, 173–186 (2014)
84. Sighireanu, M., Pérez, J.A.N., Rybalchenko, A., Gorogiannis, N., Iosif, R., Reynolds, A., Serban, C., Katelaan, J., Matheja, C., Noll, T., Zuleger, F., Chin, W.N., Le, Q.L., Ta, Q.T., Le, T.C., Nguyen, T.T., Khoo, S.C., Cyprian, M., Rogalewicz, A., Vojnar, T., Enea, C., Lengal, O., Gao, C., Wu, Z.: SL-COMP: Competition of solvers for separation logic. In: Proc. TACAS, part 3. LNCS this volume, Springer (2019)
85. Simon, L., Berre, D.L., Hirsch, E.A.: The SAT2002 competition. Annals of Mathematics and Artificial Intelligence **43**(1), 307–342 (2005), https://doi.org/10.1007/s10472-005-0424-6
86. Steffen, B., Jasper, M., Meijer, J., van de Pol, J.: Property-preserving generation of tailored benchmark Petri nets. In: 17th International Conference on Application of Concurrency to System Design (ACSD). pp. 1–8 (June 2017)

87. Steffen, B., Howar, F., Isberner, M., Naujokat, S., Margaria, T.: Tailored generation of concurrent benchmarks. STTT **16**(5), 543–558 (Oct 2014)
88. Steffen, B., Isberner, M., Naujokat, S., Margaria, T., Geske, M.: Property-driven benchmark generation. In: Model Checking Software - 20th International Symposium, SPIN 2013, Stony Brook, NY, USA, July 8-9, 2013. Proceedings. pp. 341–357 (2013)
89. Steffen, B., Isberner, M., Naujokat, S., Margaria, T., Geske, M.: Property-driven benchmark generation: synthesizing programs of realistic structure. International Journal on Software Tools for Technology Transfer **16**(5), 465–479 (Oct 2014)
90. Steffen, B., Jasper, M.: Property-preserving parallel decomposition. In: Models, Algorithms, Logics and Tools. LNCS, vol 10460, pp. 125–145. Springer (2017)
91. Stump, A., Sutcliffe, G., Tinelli, C.: StarExec: A cross-community infrastructure for logic solving. In: Proc. IJCAR, pp. 367–373. LNCS 8562, Springer (2014), https://doi.org/10.1007/978-3-319-08587-6_28
92. Sutcliffe, G.: The CADE-16 ATP System Competition. Journal of Automated Reasoning **24**(3), 371–396 (2000)
93. Sutcliffe, G.: The CADE-17 ATP System Competition. Journal of Automated Reasoning **27**(3), 227–250 (2001)
94. Sutcliffe, G.: The IJCAR-2004 Automated Theorem Proving Competition. AI Communications **18**(1), 33–40 (2005)
95. Sutcliffe, G.: The CADE-20 Automated Theorem Proving Competition. AI Communications **19**(2), 173–181 (2006)
96. Sutcliffe, G.: The 3rd IJCAR Automated Theorem Proving Competition. AI Communications **20**(2), 117–126 (2007)
97. Sutcliffe, G.: The CADE-21 Automated Theorem Proving System Competition. AI Communications **21**(1), 71–82 (2008)
98. Sutcliffe, G.: The 4th IJCAR Automated Theorem Proving Competition. AI Communications **22**(1), 59–72 (2009)
99. Sutcliffe, G.: The CADE-22 Automated Theorem Proving System Competition - CASC-22. AI Communications **23**(1), 47–60 (2010)
100. Sutcliffe, G.: The 5th IJCAR Automated Theorem Proving System Competition - CASC-J5. AI Communications **24**(1), 75–89 (2011)
101. Sutcliffe, G.: The CADE-23 Automated Theorem Proving System Competition - CASC-23. AI Communications **25**(1), 49–63 (2012)
102. Sutcliffe, G.: The 6th IJCAR Automated Theorem Proving System Competition - CASC-J6. AI Communications **26**(2), 211–223 (2013)
103. Sutcliffe, G.: The CADE-24 Automated Theorem Proving System Competition - CASC-24. AI Communications **27**(4), 405–416 (2014)
104. Sutcliffe, G.: The 7th IJCAR Automated Theorem Proving System Competition - CASC-J7. AI Communications **28**(4), 683–692 (2015)
105. Sutcliffe, G.: The 8th IJCAR Automated Theorem Proving System Competition - CASC-J8. AI Communications **29**(5), 607–619 (2016)
106. Sutcliffe, G.: The CADE ATP System Competition - CASC. AI Magazine **37**(2), 99–101 (2016)
107. Sutcliffe, G.: The CADE-26 Automated Theorem Proving System Competition - CASC-26. AI Communications **30**(6), 419–432 (2017)
108. Sutcliffe, G.: The 9th IJCAR Automated Theorem Proving System Competition - CASC-29. AI Communications **31**(6), 495–507 (2018)
109. Sutcliffe, G., Suttner, C.: The CADE-18 ATP System Competition. Journal of Automated Reasoning **31**(1), 23–32 (2003)

110. Sutcliffe, G., Suttner, C.: The CADE-19 ATP System Competition. AI Communications **17**(3), 103–182 (2004)
111. Sutcliffe, G., Suttner, C.: The State of CASC. AI Communications **19**(1), 35–48 (2006)
112. Sutcliffe, G., Suttner, C., Pelletier, F.: The IJCAR ATP System Competition. Journal of Automated Reasoning **28**(3), 307–320 (2002)
113. Sutcliffe, G., Suttner, C.: Special Issue: The CADE-13 ATP System Competition. Journal of Automated Reasoning **18**(2) (1997)
114. Sutcliffe, G., Suttner, C.: The CADE-15 ATP System Competition. Journal of Automated Reasoning **23**(1), 1–23 (1999)
115. Sutcliffe, G., Urban, J.: The CADE-25 Automated Theorem Proving System Competition - CASC-25. AI Communications **29**(3), 423–433 (2016)
116. Suttner, C., Sutcliffe, G.: The CADE-14 ATP System Competition. Journal of Automated Reasoning **21**(1), 99–134 (1998)
117. Waldmann, J.: Report on the termination competition 2008. In: Proc. of WST (2009)