# The Development of CASC

Francis Jeffry Pelletier
*Department of Computing Science*
*University of Alberta*
*Edmonton, Alberta, Canada T6G 2H1*
*E-mail: jeffp@cs.ualberta.ca*

Geoff Sutcliffe
*Department of Computer Science*
*University of Miami*
*Coral Gables, FL 33124, USA*
*E-mail: geoff@cs.miami.edu*

Christian Suttner
*Cirrus Management*
*Munich, Germany*
*E-mail: christian@suttner.info*

Researchers who make theoretical advances also need some way to demonstrate that an advance really does have general, overall positive consequences for system performance. For this it is necessary to evaluate the system on a set of problems that is sufficiently large and diverse to be somehow representative of the intended application area as a whole. It is only a small step from system evaluation to a communal system competition. The CADE ATP System Competition (CASC) has been run annually since 1996. Any competition is difficult to design and organize in the first instance, and to then run over the years. In order to obtain the full benefits of a competition, a thoroughly organized event, with an unambiguous and motivated design, is necessary. For some issues relevant to the CASC design, inevitable constraints have emerged. For other issues there have been several choices, and decisions have had to be made. This paper describes the evolution of CASC, paying particular attention to its design, design changes, and organization.

Keywords: competition, automated theorem proving, history

## 1. Introduction

An advance in the underlying theory of a subdiscipline of AI can result in an apparently impressive improvement in the performance of a system that incorporates the advance. This conclusion typically comes from observing improved performance of the system on some test problems. However, the improvement in performance may be for only the problems used in the testing, while performance on other problems may be degraded, possibly resulting in an overall degradation of the system's performance. This typically comes about when the incorporation of the advance increases the resources required overall, but has benefits on only some problems (e.g., those used in the testing) and possibly deteriorates performance on other problems. In general, a localized theoretical advance is rarely sufficient to increase the overall performance of any complex system. Therefore, researchers who make theoretical advances also need some way to demonstrate that an advance really does have general, overall positive consequences for system performance. For this it is necessary to evaluate the system on a set of problems that is sufficiently large and diverse to be somehow representative of the intended application area as a whole.

Establishing a representative set of test problems, for systems to attempt, is non-trivial. One suggestion, that a randomly chosen set of problems from the problem domain be used, seems to be inappropriate in most areas of AI. This is because there is generally no notion of what the entire problem space is, and therefore there is no well-defined notion of a randomly chosen problem set. An alternative possibility is to determine a set of benchmark problems. The notion of a benchmark set is also often problematic, since in general it is not known what is the full extent of the problems in the area. Nonetheless, the notion of a benchmark set is, in principle, something that can be agreed upon, at least to some extent, by researchers in the relevant area. It is likely to be easier to construct this sort of test apparatus than it is to get an approximation to a randomly chosen set from the problem domain. If there are enough benchmark problems, then it may be possible to randomly choose only some of these to test the system's ability to deal adequately with the entire set of benchmarks.

AI Communications 0 () 0
ISSN 0921-7126 / $8.00 © , IOS Press

## 2. Evaluating ATP

The foregoing picture gives a perspective from which to view the field of Automated Theorem Proving for classical first order logic (ATP), as it developed over the last four decades. The need for empirical evaluation is absolute in ATP: Analytic evaluation of ATP techniques, such as presented in [1, 3, 9], provides insights into theoretical capabilities. However, complete analysis of the search space at the 1st order level is impossible. It is therefore necessary to make empirical evaluations of theoretical advances, as implemented in ATP systems.

Prior to 1993, research in ATP was characterized by researchers who used their own problems to test their systems. There were a few attempts to construct lists of test problems for everyone to use, so that announced results could be more meaningful. However, these lists, e.g., [7], were themselves constructed by individual researchers, and did not have the general backing of the community as benchmark problems. Given all this, it was difficult to determine which theoretical research efforts in the field of ATP were really promising, and which merely looked good thanks to the specific testing performed. Progress in the development of more powerful ATP systems was correspondingly hampered.

This state of affairs changed in 1993 with the release of the TPTP (Thousands of Problems for Theorem Provers) problem library [11], in which published test problems are collected, and to which ATP researchers and users (both academic and "real world") are encouraged to contribute new problems as they are discovered or conceived. Many researchers have taken the opportunity to contribute problems to this common pool, and this has led to the situation where the TPTP contains pretty much all the problems that current researchers think of as benchmarks. The TPTP is continually growing – the first release of the TPTP in November 1993 contained 2295 problems, while the June 2001 release contained 5882 problems, representing an average annual growth of close to 500 problems.

Having a benchmark set, such as the TPTP, is only half the battle in evaluating and improving (and evaluating the improvement of) research in an area. Additionally it is necessary for the benchmark problems to be used appropriately. A simple report of which problems in the benchmark set can be solved, and the resources used, does not provide specific enough information to be useful. Key elements of a meaningful evaluation of ATP systems include [13]:

- Clearly identifying the type(s) and properties of the systems being evaluated, e.g., the level of automation.
- Establishing the criteria for evaluation, e.g., the number of problems solved.
- Removing the influence of problems with aberrant encoding characteristics, e.g., excluding problems that are designed to be specifically well-suited or ill-suited for a particular calculus or system.
- Controlling the effects of artifacts of the benchmark set, e.g., limiting the effect of very large numbers of very similar problems.
- Using problems that are appropriately difficult, e.g., excluding easy problems that all systems can solve, because they do not provide any differentiation between systems.
- Grouping the benchmark problems into classes that are reasonably homogeneous with respect to the ATP systems that (attempt to) solve the problems. The systems are then evaluated separately for each class of problems, thus providing a fair evaluation in the face of system specialization.

These, and more detailed issues, have been carefully investigated, resulting in two methodic schemes for the impartial empirical evaluation of ATP systems: system ranking by subsumption and state-of-the-art system rating [13]. The schemes are being used as the basis for a long term evaluation of ATP systems. This long term evaluation is providing meaningful information about the performance characteristics of the ATP systems, and, as a useful side effect, is providing evidence of progress in ATP [14]. The state-of-the-art scheme is the basis for the primary ranking scheme used in CASC. For ATP researchers, the existence of the TPTP and the evaluation schemes makes it possible to "demonstrate that the [theoretical] advance really does have general, overall positive consequences for system performance".

It is only a small step from system evaluation to a communal system competition. The CADE ATP System Competition (CASC) has been run annually since 1996.[1] In addition to the primary aim of evaluating the relative capabilities of the systems in the area, a competition, no matter what the field, has other effects. For the relevant community, a competition provides moti-

[1] The competitions are normally numbered the same as the corresponding CADE, i.e., CASC-13, CASC-14, etc. In 2001 CADE was part of the International Joint Conference on Automated Reasoning, and was named "CASC-JC" for "Joint Conference".

vation for implementing and fixing systems, and provides an inspiring environment for personal interaction among the researchers. For the wider community, a competition exposes the systems to researchers outside the narrow group, and introduces new or isolated researchers to the mainline of research. All these consequences have been evidenced in the ATP community, as a result of CASC.

The remainder of this paper describes the evolution of CASC, paying particular attention to its design, design changes, and organization. It is important, at all times, to keep in mind that CASC has the TPTP and the evaluation schemes as foundation stones. Without these, the main "raison d'être" for CASC – the evaluation of the ATP systems – would be on shaky ground, and it is doubtful that CASC would survive.

## 3. The Design of CASC

Any competition is difficult to design and organize in the first instance, and to then run over the years. In order to obtain the full benefits of a competition, a thoroughly organized event, with an unambiguous and motivated design, is necessary. Unlike, say, the experience of the chess community in initiating their computer chess competitions [4], the ATP community did not have a history of human tournaments to fall back on for a basic competition design. In 1987 Jeff Pelletier tried to interest people in an ATP system competition, but most developers thought either that it would be too difficult to prepare their systems for a competition, or that a competition was inappropriate for evaluating scientific progress. In 1993 Ross Overbeek ran a very specialized competition, using small sets of specifically selected problems, at CADE-11 [6]. This competition allowed a detailed analysis and comparison of the performances of the ATP systems on the selected problems, but did not aim to evaluate the general usefulness of the systems. Thus the design of CASC had to be developed from first principles.

In order for a comparison of different ATP systems to make sense, it is necessary that all the systems should be attempting to capture a common notion of truth, as is described in the Realist viewpoint in [8], whereby all the differing proof systems are viewed merely as different ways of demonstrating facts about the same abstract realm of logic. Given this commonality across all systems, it has been possible to design an ATP competition that determines winners, relative to some clearly specified constraints. For some issues relevant to the CASC design, inevitable constraints have emerged. For other issues there have been several choices, and decisions have had to be made.

As is the case in all competitions, and regardless of the care with which the competition has been designed, unforeseen circumstances arise. In order to provide for impartial resolution of such matters, CASC is overseen by a panel of knowledgeable researchers who are not participating in the event. Once the design details of each CASC have been finalized, only the panel has the right to make changes or exceptions.

The CASC design has several aspects: what ATP systems may compete, how the competition is split into divisions, what problems are eligible for use, how many and which problems are used, what resource limits are imposed, what systems properties are required, how the systems are ranked, and what organizational rules apply. For some aspects, the original decisions have not changed over the years, while for others there has been expansion and adaptation. Here the basic design of CASC is reviewed. Full motivations, details, and discussion are in [18].

**What ATP Systems:** CASC evaluates ATP systems that solve problems in classical first order logic. The systems have to run on a homogeneous suite of UNIX workstations, supplied by the competition organizers. The systems have to be fully automatic, i.e., any command line switches have to be the same for all problems. A system is considered to have solved a problem when it outputs an assurance ("yes") that a proof or disproof exists. However, systems that output solutions (e.g., proofs or models) are acknowledged in the presentation of the CASC results. In CASC-JC, proof output was also used for system ranking (see "Ranking" below). The systems have to be sound, and the organizers test for this by submitting non-theorems to the systems that search for proofs, and theorems to the systems that search for disproofs. Claiming to have found a proof of a non-theorem or a disproof of a theorem indicates unsoundness. This combats the use of the "winning strategy" by which a system merely says "yes" as soon as it is presented with a problem.[2] The systems do not have to be complete in any sense, including calculus, search control, implementation, or resource requirements.

**Divisions:** CASC is run in divisions, according to problem characteristics. Each division uses problems

[2] Since first order logic is semidecidable, there can be no absolute test of soundness. Empirical testing provides strong evidence of soundness.

that have certain logical, language, and syntactic characteristics, so that the ATP systems that compete in the division are, in principle, suited to attempting the problems. Since CASC-14 some divisions have been further divided into categories. The categories have no effect on the competition rankings, which are made at only the division level. The categories are defined in order to make it possible to analyze, at a more fine grained level, which systems work well for what types of problems.

The characteristics used to define the divisions and categories include:

- Whether or not the problem is a theorem. CASC has always had divisions in which the systems are required to prove theorems (by refutation for CNF problems). Since CASC-14 there has also been a division of CNF non-theorems, i.e., satisfiable clause sets (SAT problems), in which the systems are required to check that the problem clauses are satisfiable.
- Whether the problems are presented in first-order form (FOF problems) or in conjunctive normal form (CNF problems). Many ATP systems are designed to operate on only CNF problems, and solve FOF problems by converting them to a CNF form and checking for satisfiability. The first CASC used only CNF problems, but from CASC-14 on FOF problems have been used in a separate division.
- Whether or not equality is present in the problem. This has a major impact on both the logical features of problems and the algorithms employed to solve them. A particular type of CNF equality problem is that in which each clause is an identity claim or its negation. These problems are called UEQ (for "unit equality") problems in CASC.
- Whether or not the clauses of a CNF problem are all Horn. This distinction is relevant because there are many calculi that are complete for Horn problems, but incomplete for non-Horn problems. This distinction has been made since CASC-14.
- Whether a problem is "really first-order" or "effectively propositional". This difference can more precisely be expressed as whether the problem's Herbrand universe is infinite (really first-order) or finite (effectively propositional). In CASC-JC this characteristic was used to define a separate division.

Since there are important differences in the types of problems, and practical differences in the techniques required to solve such problems (e.g., a system that is able to solve SAT problems is typically not intended to be able to solve UEQ problems, and so on), CASC is run in divisions based on these characteristics.

Ever since the first CASC there has been a MIX division, which consists of a mixture of various types of CNF theorems other than unit equality problems (see the UEQ division below). The MIX division groups together problems that are reasonably similar from a user perspective, and historically have been solved using the same sorts of techniques. Some entrants view the MIX division as the central part of CASC. Other divisions that have appeared in CASC are UEQ (unit equality CNF problems), SAT (CNF non-theorems), FOF (FOF problems), SEM (semantically selected problems – see Section 8), and EPR (effectively propositional CNF problems). Table 1 in Section 10 provides a summary of which divisions were run in which CASCs. The divisions are discussed further in Sections 4 to 9.

Systems that cannot run on the standard UNIX hardware (such as those that use special hardware, e.g., Lisp machines or Macintoshes), or cannot be entered into the competition divisions for any other reason (e.g., the entrant is a competition organizer or panel member), can be entered into a demonstration division.

**Eligible Problems:** The problems for CASC are taken from the TPTP. Problems that are known to have been designed to be specifically well-suited or ill-suited for a particular calculus or system (documented as "biased" problems in the TPTP) are excluded. The problems have to be "difficult" according to the TPTP difficulty rating [13], so that they are expected to be solved by some but not all of the systems, so as to provide differentiation between the systems.

**Problem Selection:** The number of problems used in each division is chosen by the organizers, between a minimal value that is determined from confidence requirements (that the fraction of problems solved projects to the set of all eligible problems), and a maximal value constrained by the number of machines, the time available for the competition, the number of systems entered, and the CPU time limit imposed on each solution attempt (see "Resource Limits" below). The problems used are randomly selected from the eligible problems at the start of the competition, based on a seed supplied by the competition panel. To ensure that no system receives an advantage or disadvantage due to the specific presentation of the problems in the TPTP, the symbols in the problems are renamed and the formulae are randomly reordered.

**Resource Limits:** A CPU time limit is imposed on each solution attempt. The CPU time limit is chosen by the organizers, between a minimal value of 180 seconds, and a maximal value constrained by the number of machines, the time available for the competition, the number of systems entered, and the minimal numbers of problems that need to be used in each division. Additionally, wall clock, memory, and disk limits have been imposed in some CASCs.

**System Properties:** The precomputation and storage of any information about individual TPTP problems is not allowed. For every problem solved, the system's solution process has to be reproducible by running the system again.

**Ranking:** In each division, the systems are ranked according to the number of problems solved (a "yes" output, giving an assurance of the existence of a solution). If several systems solve the same number of problems, then those systems are ranked according to their average CPU times over problems solved. In CASC-JC a second ranking was done in the MIX division, ranking the systems according to the number of problems solved with a proof output. The addition of this ranking was motivated by the observed need for an explicit proof output in some applications of ATP.

**Organization:** Systems can be entered at only the division level, and can be entered into more than one division. A system that is not entered into a division is assumed to perform worse than the entered systems, for that type of problem. A "catch-all" rule is used to deal with any unforeseen circumstances: *No cheating is allowed.* The panel is allowed to disqualify entrants due to unfairness, and to adjust the competition rules in case of misuse.

The following sections track the design changes, outcomes, and observations from CASC-13 (1996) through to CASC-JC (2001). Full details of each competition appear in the paper cited in the section heading.

## 4. CASC-13 (Rutgers University, USA, 1996) [10]

The CASC design, described in Section 3, was developed for and used at CASC-13. In this first CASC, only two divisions were run: the MIX division and the UEQ division. Two ranking schemes were used in each division. The first scheme focused on the ability to find as many solutions as possible – essentially ranking the systems according to the numbers of problems solved, while the second scheme measured solutions-per-unit-time. Additional to the distinction between MIX and UEQ problems, in CASC-13 a distinction was made between "monolithic" and "compositional" systems. The idea was that in monolithic systems no special subprogram would be chosen just because the problem manifested a certain style or characteristic, whereas compositional systems could be made up from several distinct monolithic subsystems, and a subsystem chosen based on the given problem's characteristics. It was planned that if a compositional system solved the most problems in a division, then two winners would be declared - the compositional system and also the best performing monolithic system. As it turned out, monolithic systems solved the most problems in both divisions.

Winners:

1. MIX: E-SETHEO, entered by R. Letz, of Technische Universität München.
2. UEQ: Otter 3.0.4z, entered by W. McCune, of Argonne National Laboratories.

The major observations from CASC-13 were:

- This was the first time that the relative capabilities of ATP systems had been seriously compared.
- The competition stimulated ATP research – most entrants made special efforts to improve the autonomous performance of their systems, and all the entrants had to complete the implementation and debugging of their systems.
- It is hard to clearly distinguish between monolithic and compositional systems. Some intuitively "monolithic" systems have some autonomous adaptation to the given problem, and it is not clear when such adaptation makes the system "compositional".
- The competition provided an inspiring environment for personal interaction between ATP researchers – there was more excitement and activity than the organizers expected!
- Many of the CADE conference delegates came to see the competition – the competition thus exposed the ATP systems to researchers both within and outside the ATP community.

## 5. CASC-14 (Townsville, Australia, 1997) [19]

The success of CASC-13 motivated expansion in CASC-14. A new competition division was added: the SAT division containing satisfiable CNF problems.

A FOF demonstration division was also added, to give natural deduction systems, which typically operate with the full range of connectives of first order logic (the "natural form"), a chance to demonstrate their abilities. It was a demonstration division rather than a competition division because the infrastructure for a FOF competition division was not yet in place. Systems that operated on CNF could also enter the FOF division, by prepending a FOF-to-CNF converter. This required them to spend some CPU time doing the conversion to CNF, before starting deduction. Some contestants (and other theorists) thought that a FOF division should be the "central part" of CASC.

The distinction drawn between compositional and monolithic systems at CASC-13 was not very successful or meaningful. It was difficult to distinguish between the two types of systems, and the results of CASC-13 showed no salient evidence that the compositional systems had an advantage over the monolithic systems. At the same time, the CASC-13 results suggested that it would be interesting to separate the systems' performances within the MIX division according to finer grained problem types. It was then realized that with an appropriately fine grained categorization, compositional systems would invoke the same monolithic component for every problem in such a category. This would then enable the individual subsystems of a compositional system to be evaluated fairly against monolithic systems. Therefore, in CASC-14, the monolithic-compositional distinction was abandoned, and the MIX division was divided into four categories: HNE (Horn problems with No Equality), HEQ (Horn problems with Equality), NNE (Non-Horn problems with No Equality), and NEQ (Non-Horn problems with Equality).

The two ranking schemes of CASC-13 identically ranked all the systems, for both divisions. As it happens, systems that solve many problems also solve them quickly. Therefore the solutions-per-unit-time ranking scheme was abandoned. Ranking according to the number of problems solved, with ties decided by the average CPU times taken over problems solved, was established as the primary CASC ranking scheme.

Winners:

1. MIX: Gandalf, entered by T. Tammet, of Göteborg University.
2. UEQ: Waldmeister, entered by A. Buch and T. Hillenbrand, of Universität Kaiserslautern.
3. SAT: SPASS 0.77, entered by C. Weidenbach, of Max-Planck-Institut für Informatik.

The major observations from CASC-14 were:

- There were many new entrants – people came out of the woodwork. There were both long-standing theorists in the field who decided to enter the competition, and new researchers who were attracted to the competition. Table 1 in Section 10 shows that 10 of the 14 systems had not been entered in CASC-13.
- The introduction of the SAT and FOF divisions made it possible for those types of systems to enter. Two systems employing natural deduction were entered into the FOF demonstration division. They were, however, outperformed by the systems that converted to CNF. The best performing system in the FOF demonstration division was SPASS 0.77, entered by C. Weidenbach, of Max-Planck-Institut für Informatik.
- Many of the systems were more refined at the control level. Several entrants produced "automatic" modes, which autonomously adapt the system to the given problem, according to its characteristics.
- Gandalf was the first system in CASC to use a "time-slicing" control strategy, where one deduction strategy is attempted for a short while, and if it doesn't produce a solution then it is stopped, and the proof attempt is begun anew with a different strategy. Variants of this approach have since been used in other ATP systems, e.g., E-SETHEO, Vampire, and SSCPA.
- Waldmeister began its stranglehold on the UEQ division.

## 6. CASC-15 (Lindau, Germany, 1998) [12]

In CASC-15 the PEQ (Pure Equality) category was added to the MIX division, containing problems with only the `equal`/2 predicate, which previously were in the HEQ and NEQ categories. The problems were removed from the HEQ and NEQ categories, so that they were limited to problems with some, but not pure, equality. This finer grained categorization further allowed specialized systems to show their particular abilities. The CASC-14 FOF demonstration division was promoted to be a competition division in CASC-15, with two categories: FEQ (FOF with equality) and FNE (FOF without equality).

In CASC-13 and CASC-14, the minimal numbers of problems to be used in each division and category were based on simple statistical confidence measures

[10]. The problem of confidence in the CASC results aroused the interest of some statisticians at the Technische Universität München, who developed a more elegant model for determining the minimal number of problems to be used [2]. Given the number of eligible problems and a required average degree of confidence that the fraction of problems solved projects to the entire population, the tables in [2] specify how many of the eligible problems have to be used. This new scheme has been used since CASC-15.

In CASC-14 the organizers took on the task of ensuring that the submitted systems ran correctly in the competition environment. It was decided that for CASC-15 some of the competition control scripts would be made available to the entrants, who would then have to ensure that their systems behaved as required. This allowed more automation during the event itself. In an attempt to impose more standardization on the competition, all output was required to be to `stdout`, and no other output was deemed relevant.

In CASC-14 it was noticed that excessive memory usage could cause a system to run for a very long time, due to swapping. To counter this, for CASC-15 it was decided that a wall clock time limit should be imposed. However the organizers only got around to implementing it in the control scripts for CASC-JC – see Section 9.

Winners:

1. MIX: Gandalf c-1.1, entered by T. Tammet, of Göteborg University.
2. UEQ: Waldmeister 798, entered by T. Hillenbrand, et al., of Universität Kaiserslautern.
3. SAT: SPASS 1.0.0a, entered by C. Weidenbach, of Max-Planck-Institut für Informatik.
4. FOF: SPASS 1.0.0a, entered by C. Weidenbach, of Max-Planck-Institut für Informatik.

The major observations from CASC-15 were:

- No natural deduction systems were entered in the FOF division. There was speculation whether the developers of the natural deduction systems entered in CASC-14 had realized that their natural deduction systems were not competitive with the CNF conversion systems. Natural deduction made a return in CASC-16, with the entry of the MUSCADET system.
- In general, there was a realization for some entrants that their systems had fallen behind the rapidly improving state-of-the-art.
- Rather than entering systems that were to solve "all problems", or at least were designed to be general purpose, some entrants spent considerable time tuning their systems for only the eligible problems in CASC-15. This was achieved by optimizing the system control mechanisms for the eligible problems, without concern for performance on non-eligible problems. This "overtuning" was considered by some to be unproductive in the broader context of ATP development, and it continued to be a contentious issue until CASC-JC. At CASC-JC the issue was resolved through the use of a large number of problems from an unreleased version of the TPTP, and it was observed that tuning is apparently effective in general – see Section 9.
- A particularly effective form of adaptation to the given problem was (and still is) employed by Waldmeister. Waldmeister examines the problem to determine the underlying "theory" of a problem (e.g., rings, groups, condensed detachment, ...) and chooses a selection strategy and reduction ordering based on this information. As is described in Section 7, this later led to complaints from other entrants.
- The winners were new versions of the CASC-14 division winners.
- Some skewing was evident in the results of the FNE category, caused by large numbers of very similar problems, in particular the 'ALC' problems within the TPTP SYN domain.[3]
- The leading systems in CASC were too good for the problems with low TPTP difficulty ratings, often solving them in less-than-measurable time.
- The influence of CASC was being acknowledged. Many contestants claimed that the particular research they carried out over the year was due to a desire to be competitive in future CASCs. Good performance in CASC was also affecting publications and grant funding. For the first time CASC attracted newspaper publicity, with an article appearing in the local press.

## 7. CASC-16 (Trento, Italy, 1999) [16]

The competition division structure stayed the same for CASC-16. The demonstration division, which was

[3] These problems are first order logic encodings of problems from multi-modal K-logics [5].

initially conceived as a place for systems requiring special hardware, was expanded to allow panel members and organizers to enter the competition.[4]

An important novelty, adopted from CASC-16 onwards, was to enter the winning systems from the previous competition in their respective divisions (the competition archive provides access to those systems' executables and source code). This provides benchmarks against which the performance of the new systems can be judged, making it possible to make definitive statements about the progress of ATP. Table 2 in Section 10 shows evidence from CASC of progress in ATP.

In a first attempt to limit tuning for the eligible problems, in CASC-16 the lists of eligible problems were not published until after the systems had been installed on the competition machines. Further, the most up-to-date TPTP problem difficulty ratings, which have a role in determining which problems are eligible, were not released before the competition. As a result, entrants could only inaccurately determine exactly which problems would be eligible, thus reducing the extent to which tuning was possible.

For CASC-16, lists of very similar problems in the TPTP were identified, and a limit was imposed on the number of very similar problems in any division or category. Between CASC-15 and CASC-16 the TPTP problem difficulty rating scheme was improved, and in CASC-16 the minimal difficulty rating for eligible problems was increased to 0.21. These two changes provided a more appropriate selection of problems in terms of both breadth and difficulty.

Winners:

1. MIX: Vampire 0.0, entered by A. Riazanov and A. Voronkov, of University of Manchester.
2. UEQ: Waldmeister 799, entered by T. Hillenbrand, et al., of Universität Kaiserslautern.
3. SAT: OtterMACE 437, entered by W. McCune, of Argonne National Laboratories.
4. FOF: SPASS 1.0.0T, entered by C. Weidenbach, of Max-Planck-Institut für Informatik.

The major observations from CASC-16 were:

- Before CASC-16, there was, for the first time, some acrimonious debate regarding the design and implementation of the competition. The debate started with a complaint concerning the way Waldmeister adapts to the given problem, but expanded to a range of issues.
- It was evident that not releasing the lists of eligible problems was insufficient to limit tuning for the eligible problems – an estimate of which problems would be eligible was still possible. Some entrants lobbied for the introduction of rules in the CASC design to limit tuning. The organizers refrained from introducing formal rules to limit tuning, because no sharp border between what is acceptable and what is not could be identified, and it would anyway be extremely diifficult to check conformance to such a rule. The CASC-JC design introduced changes that make excessive tuning ineffective - see Section 9.
- There was a significant increase of interest from within the ATP community. As a result, a session at CADE was dedicated to discussing the CASC results. There was an interesting debate regarding the desirability of a focus on implementation versus attention to theory development. It seemed clear that much effort was being spent on carefully constructing and tuning systems, and this was felt by some to be at the expense of basic research that had not yet been well implemented.

In August 1999, i.e., about a month after the competition, E 0.5 and E-SETHEO 99csp were found to be unsound in certain rare circumstances. The unsoundness was due to a bug in E, which was also used as a component of E-SETHEO. Unsoundness is unacceptable, and the competition panel retrospectively disqualified the two systems from being ranked in the competition. (It must be noted that the unsoundness was entirely accidental, and that there was no attempt to deceive. Further testing indicated that the unsoundness had not affected the systems' performances in the competition, thus although the systems were unranked, their performance data was still valid.) Given the empirical nature of the soundness testing performed before the competition (see Section 3) it is not surprising that unsoundness might go unnoticed. The highlighting of the soundness issue after CASC-16 led to a revival of interest in proof output and verification. Although proof output and verification at CASC would not ensure the soundness of systems (because, as was the case with E and E-SETHEO, any unsoundness may not arise in the solutions of the selected competition

[4] Christian Suttner, one of the organizers of CASC-13, had entered his system, SPTHEO, in that competition. Some contestants had questioned the wisdom of allowing this (but there were no concerns about improprieties at that competition). Furthermore, Geoff Sutcliffe wanted some way to enter his SSCPA system in the CASC-16 competition.

problems), in general, the combination of an ATP system and a sound verification system does constitute a sound ATP system.

## 8. CASC-17 (CMU, USA, 2000) [17]

At CASC-16 there was debate about the relevance of CASC, with claims that the "problems did not reflect real usage". Although this debate is complex, it was decided to react to this apparent interest in applications by adding a SEM division ("semantically selected") in CASC-17. The idea was to use problems from a chosen application domain, so that systems could be explicitly tuned for that type of problem. The problems used were set theoretical theorems formulated within Gödel-von Neuman-Bernays set theory.

Discussions at CASC-16 regarding the 'ALC' problems highlighted the issue of "effectively propositional" problems. These problems can be translated to propositional form and solved using specialized propositional techniques. It was considered that these problems are particularly well suited to specialized provers, and not suitable for the evaluation of general purpose first-order systems. Therefore effectively propositional problems were not eligible in CASC-17 (but were reintroduced in a separate division in CASC-JC – see Section 9).

System installation for CASC-17 required that the entrants supply the organizers with "installation packages". The motivation was to encourage developers to make installation and practical usage easier for potential users. However, more than in previous competitions, the systems required modification after installation, before they could execute in the production environment of the competition.

Winners:

1. MIX: E 0.6, entered by S. Schultz, of Technische Universität München.
2. UEQ: Waldmeister 600, entered by T. Hillenbrand, et al., of Universität Kaiserslautern.
3. SAT: GandalfSat 1.0, entered by T. Tammet, of Tallin Technical University.
4. FOF: VampireFOF 1.0, entered by A. Riazanov and A. Voronkov, of University of Manchester.
5. SEM: E-SETHEO 2000csp, entered by S. Schultz, et al., of Technische Universität München.

The major observations from CASC-17 were:

- Again, many researchers invested in significant, year long, development in preparation for CASC.
- At CASC-17 there were fewer systems, but all the systems were reasonably strong.
- Entrants were still making efforts to make their systems effective only for the eligible problems. There was not only tuning of the systems . . . entrants were beginning to understand the process whereby a problem becomes eligible for CASC, and some were submitting TPTP performance data that was affecting problem eligibility in their favor (at that stage, performance data supplied by developers was being used to compute the TPTP problem ratings, which in turn are important for determining eligibility for CASC).

## 9. CASC-JC (Siena, Italy, 2001) [15]

A lack of interest in the SEM division, and its overlap with the existing syntactically defined divisions (the set theory SEM problems could also be used in the FOF division), led to the demise of the SEM division in CASC-JC. Analysis of system performance data on problems in the SAT division showed that there is specialization between SAT problems with equality and without equality. Therefore the SAT division was divided into two categories for CASC-JC: SEQ (SAT with equality) and SNE (SAT without equality). At CASC-17 some entrants expressed a continued interest in effectively propositional problems, claiming that first order techniques could be more effective than translation to propositional form and the use of a specialized propositional system. This prompted the introduction of the EPR (Effectively Propositional) division in CASC-JC, containing CNF problems with a finite Herbrand universe. Such problems were not used in any of the other divisions.

A TPTP problem is labelled as non-standard if the formulae are based on a known theory, e.g., set theory, but axioms not required to solve the problem have been removed (hence the axioms do not completely capture the theory), or lemmas have been supplied, to make the problem easier for an ATP system. Up to CASC-17, non-standard problems were excluded from CASC, as there was a perceived danger that the problems might be biased towards a particular ATP system. Between CASC-17 and CASC-JC it was concluded that such modifications are generally effective for all ATP systems. Therefore non-standard problems were eligible in CASC-JC.

In order to make tuning for all the eligible problems impossible, the CASC-JC problems were taken from an unreleased version of the TPTP. The systems could thus not be tuned for the new problems in that TPTP version. Over-tuning for the old problems in the TPTP was potentially disadvantageous, because it could degrade performance on the new problems, with a consequent degradation in overall performance.

A survey of ATP users after CASC-17 indicated that for some applications of ATP there is a need for the production of proofs and models, and that the output must be produced as part of the system run, i.e., it cannot be deferred to a later run of the system. To encourage research into proof and model presentation, and the implementation of proof generation and verification as part of an integrated reasoning process, the CASC-JC MIX division was ranked in two classes, the Assurance class and the Proof class. The Assurance class was ranked as before, according to the number of problems solved, while the Proof class was ranked according to the number of problems solved with an acceptable proof output. Systems that did not, for whatever reason, generate proofs, were able to compete in only the Assurance class.

The wall clock limit, designed for CASC-15, was implemented, and a new set of "clean execution" requirements were established to help ensure that the systems would run correctly in the competition environment.

Winners:

1. MIX Proof class: VampireJC 2.0, entered by A. Voronkov and A. Riazanov, of University of Manchester.
2. MIX Assurance class: A tie was declared between VampireJC 2.0, entered by A. Voronkov and A. Riazanov, of University of Manchester, and E-SETHEO csp01, entered by G. Stenz, et al., of Technische Universität München
3. UEQ: Waldmeister 601, entered by T. Hillenbrand, et al., of Universität Kaiserslautern and Max-Planck-Institut.
4. SAT: GandalfSat 1.0, entered by T. Tammet, of Tallin Technical University.
5. FOF: E-SETHEO csp01, entered by G. Stenz, et al., Technische Universität München.
6. EPR: E-SETHEO csp01, entered by G. Stenz, et al., Technische Universität München.

The major observations from CASC-JC were:

- There was a high level of enthusiasm and interest from both entrants and observers. The entrants made significant efforts to meet the requirements of the competition design, and as a result the systems were more robust and usable than before.
- There was generally strong performance on the new problems in the TPTP release used. This countered the concern that systems had been overtuned for the competition: if the systems were tuned using TPTP problems, then that tuning also worked for the new problems, and therefore seems to be effective in general. The results on the unseen problems, especially in comparison with the results for the old problems, provided interesting insights into the systems.
- Instituting the Proof class in the MIX division further stimulated interest and research into proof production. The effort required to produce proofs was evident. In particular, some systems solved some problems within the time limit but ran overtime while building the proofs. Due to the lack of differentiation between the time when a solution was found and the time when the system terminated, those solutions were not counted for the rankings. Future CASCs will differentiate between the two cases.
- In the environment of the combined IJCAR conference, observers with a broad range of perspectives were evidently interested in the competition and its outcomes. In particular, it was pleasing to see some commercial interest in the best performing systems.

## 10. Conclusion

This paper has described how an underlying need for empirical evaluation of ATP systems has been translated into an annual ATP system competition. The underlying need motivated the development of a communally accepted benchmark set – the TPTP, and the specification of formal schemes for evaluating ATP systems. These in turn have provided the practical foundations for the design and development of CASC. CASC is now an established and influential event in the ATP calendar. Table 1 provides an overview of the expansion and stabilization of CASC over the years.[5]

CASC has fulfilled its aims: evaluation of the relative capabilities of ATP systems, stimulation of ATP research, providing motivation for improving implementations, and having an exciting event that exposes ATP systems to researchers both within and outside

[5] The numbers of systems exclude close variants of the same systems.

| | CASC-13 | CASC-14 | CASC-15 | CASC-16 | CASC-17 | CASC-JC |
|---|---|---|---|---|---|---|
| Divisions | MIX<br>UEQ | MIX<br>UEQ<br>SAT | MIX<br>UEQ<br>SAT<br>FOF | MIX<br>UEQ<br>SAT<br>FOF | MIX<br>UEQ<br>SAT<br>FOF<br>SEM | MIX<br>UEQ<br>SAT<br>FOF<br>EPR |
| Problems | 100 | 152 | 180 | 165 | 215 | 440 |
| Systems | 14 | 16 | 14 | 13 | 12 | 14 |
| New systems | 14 | 10 | 4 | 5 | 2 | 1 |

Table 1
CASC overview data

the ATP community. For the entrants, their research groups, and their systems, there has been substantial publicity, both within and outside the ATP community. The significant efforts that have gone into developing the ATP systems have received public recognition. The competition has provided an overview of which researchers and research groups have decent, running, fully automatic ATP systems.

For the entrants, there have been some useful side-effects:

- The systems have necessarily had to be debugged and engineered to run and stop correctly without user intervention. An important facet of this is improved autonomous adaptation to the given problem, according to its characteristics. As a result of these developments it has become easier to try out and use the ATP systems.
- As well as being useful in its own right, the improved stability and autonomy of ATP systems has made it possible to perform extensive automatic testing of ATP systems, leading to further insights and improvements.
- The decision by some developers to tune for CASC has led to the production of automatic tuning techniques and tools. This is a useful development, as it allows the system to be tuned for particular applications by submitting sample problems.
- By having CASC as a live event at CADE, interested researchers have been brought together in an inspiring environment, and there have been fruitful exchanges of idea. As one entrant has said “Digging for and reading papers is a lot more time-consuming (and has a higher entry barrier) than sitting round the desk at the CASC dinner and swapping war stories ;-)”.

For the ATP community, CASC shows how theoretical advances are embodied in real implementations, and provides evidence of the corresponding progress in ATP. In each CASC since CASC-16, the new systems have outperformed the previous year’s division winners (which are automatically entered - see Section 7), as shown in Table 2. Further evidence of progress in ATP, e.g., declining TPTP problem ratings and the solution of previously unsolved problems, is given in [14]. CASC is a contributing cause of this improvement. The online CASC archives, including the competition problems and the systems’ source and binary codes, allows developers and users to experiment with the systems, leading to further insights and improvements in ATP.

| | Division winner<br>Problems/Solved by winner/Solved by previous winner (Place)<br>CASC-15 | <br><br>CASC-16 | <br><br>CASC-17 | <br><br>CASC-JC |
|---|---|---|---|---|
| `MIX` | Gandalf c-1.1<br>80/61/– | Vampire 0.0<br>75/51/39 (4th) | E 0.6<br>75/57/37 (5th) | VampireJC 2.0/<br>E-SETHEO csp01<br>120/93/81 (4th) |
| `UEQ` | Waldmeister 798<br>30/30/– | Waldmeister 799<br>30/30/19 (2nd) | Waldmeister 600<br>30/30/29 (2nd) | Waldmeister 601<br>90/69/69 (2nd) |
| `SAT` | SPASS 0.95T<br>30/22/– | OtterMACE 437<br>30/16/9 (3rd) | GandalfSat 1.0<br>30/25/21 (4th) | GandalfSat 1.0<br>90/48/48 (1st) |
| `FOF` | SPASS 0.95T<br>40/39/– | SPASS 1.00T<br>30/22/19 (3rd) | VampireFOF 1.0<br>60/53/51 (2nd) | E-SETHEO csp01<br>90/75/72 (2nd) |

Table 2
Performance of previous CASC division winners

For the CASC organizers, each year reveals further issues that need careful consideration and response. Entrants in CASC have been forthcoming with ideas, criticisms, and suggestions. The changing demands of CASC have led to improvements in the ways that ATP systems are evaluated. A particularly important instance was the introduction of unseen problems into CASC (see Section 9). The success of the systems on the new problems has provided evidence that using TPTP for testing newly implemented ideas, and gauging the quality of the ideas based on the results, does not just lead to systems that can solve only TPTP problems. Rather, performance on TPTP problems apparently generalizes well to new problems and applications. A key to sustaining the value of CASC in the future is continued growth of the TPTP. Developers and users are strongly encouraged to contribute to the TPTP, particularly problems from emerging commercial applications of ATP. An important issue to be addressed in future CASCs is automated verification of the solutions (proof and models) that the systems output.

There were 35 years of theoretical research and individually-evaluated systems in automated theorem proving. In that time, many techniques and ideas were generated that needed to be evaluated in order to determine which were viable, and to integrate them into systems that were more flexible and powerful than before. For all these goals, experimental system evaluation is a crucial research tool, and competitions provide stimulus and insight that can lay the basis for the development of future ATP systems.

## References

[1] U. Dunker. Search Space and Proof Complexity of Theorem Proving Strategies. In G. Sutcliffe and C.B. Suttner, editors, *Proceedings of the CADE-12 Workshop 2C - Evaluation of Automated Theorem Proving Systems*, 1994.

[2] M. Greiner and M. Schramm. A Probablistic Stopping Criterion for the Evaluation of Benchmarks. Technical Report I9638, Institut für Informatik, Technische Universität München, München, Germany, 1996.

[3] R. Letz. On the Polynomial Transparency of Resolution. In R. Bajcsy, editor, *Proceedings of the 13th International Joint Conference on Artificial Intelligence*, pages 123–129. Morgan Kaufmann, 1993.

[4] M. Newborn. *Kasparov versus Deep Blue: Computer Chess Comes of Age*. Springer, 1997.

[5] H.J. Ohlbach and R.A. Schmidt. Functional Translation and Second-Order Frame Properties of Modal Logics. Technical Report MPI-I-95-2-002, Max-Planck-Institut für Informatik, Saarbrücken, Germany, 1995.

[6] R. Overbeek. The CADE-11 Competitions: A Personal View. *Journal of Automated Reasoning*, 11(3):315–316, 1993.

[7] F.J. Pelletier. Seventy-five Problems for Testing Automatic Theorem Provers. *Journal of Automated Reasoning*, 2(2):191–216, 1986.

[8] F.J. Pelletier. The Philosophy of Automated Theorem Proving. In Mylopolous J. and R. Reiter, editors, *Proceedings of the 12th International Joint Conference on Artificial Intelligence* , pages 1039–1045. Morgan-Kaufmann, 1991.

[9] D.A. Plaisted. The Search Efficiency of Theorem Proving Strategies. In A. Bundy, editor, *Proceedings of the 12th International Conference on Automated Deduction*, number 814 in Lecture Notes in Artificial Intelligence, pages 57–71. Springer-Verlag, 1994.

[10] G. Sutcliffe and C.B. Suttner. Special Issue: The CADE-13 ATP System Competition. *Journal of Automated Reasoning*, 18(2), 1997.

[11] G. Sutcliffe and C.B. Suttner. The TPTP Problem Library: CNF Release v1.2.1. *Journal of Automated Reasoning*, 21(2):177–203, 1998.

[12] G. Sutcliffe and C.B. Suttner. The CADE-15 ATP System Competition. *Journal of Automated Reasoning*, 23(1):1–23, 1999.

[13] G. Sutcliffe and C.B. Suttner. Evaluating General Purpose Automated Theorem Proving Systems. *Artificial Intelligence*, 131(1-2):39–54, 2001.

[14] G. Sutcliffe, M. Fuchs, and C. Suttner. Progress in Automated Theorem Proving, 1997-1999. In H. Hoos and T. Stützle, editors, *Proceedings of the IJCAI'01 Workshop on Empirical Methods in Artificial Intelligence*, pages 53–60, 2001.

[15] G. Sutcliffe, C. Suttner, and F.J. Pelletier. The IJCAR ATP System Competition. *Journal of Automated Reasoning*, To appear, 2002.

[16] G. Sutcliffe. The CADE-16 ATP System Competition. *Journal of Automated Reasoning*, 24(3):371–396, 2000.

[17] G. Sutcliffe. The CADE-17 ATP System Competition. *Journal of Automated Reasoning*, 27(3):227–250, 2001.

[18] C.B. Suttner and G. Sutcliffe. The Design of the CADE-13 ATP System Competition. *Journal of Automated Reasoning*, 18(2):139–162, 1997.

[19] C.B. Suttner and G. Sutcliffe. The CADE-14 ATP System Competition. *Journal of Automated Reasoning*, 21(1):99–134, 1998.