The IJCAR ATP System Competition

G. Sutcliffe
(geoff@cs.miami.edu)
Department of Computer Science, University of Miami, USA

C.B. Suttner
(email@csuttner.de)
Cirrus Management, München, Germany

F.J. Pelletier
(jeffp@cs.ualberta.ca)
Department of Computing Science, University of Alberta, Canada

Abstract. The results of the IJCAR ATP System Competition are presented.

Keywords: competition, automated theorem proving

1. Introduction

The CADE ATP System Competition (CASC) is an annual evaluation of fully automatic, first-order ATP systems. It has been run at each CADE since 1996. In addition to the primary aim of evaluating the relative capabilities of ATP systems, CASC aims to stimulate ATP research in general, to stimulate ATP research towards autonomous systems, to motivate implementation and fixing of systems, to provide an inspiring environment for personal interaction between ATP researchers, and to expose ATP systems to researchers both within and outside the ATP community. Fulfillment of these objectives is expected to provide stimulus and insight that can lay the basis for the development of more powerful ATP systems, leading to increased and more effective usage.

The IJCAR ATP System Competition (CASC-JC) was held on 21st June 2001, as part of the International Joint Conference on Automated Reasoning, in Siena, Italy.[1] CASC-JC was the sixth competition in the CASC series, following the successful competitions at CADEs-13 to -17 (Sutcliffe and Suttner, 1997; Suttner and Sutcliffe, 1998; Sutcliffe and Suttner, 1999; Sutcliffe, 2000; Sutcliffe, 2001b). Twenty three ATP systems, listed in Table I, competed in the various competition and demonstration divisions. The winners of the CASC-17 divisions were automatically entered into their divisions, to provide benchmarks

[1] In 2001 CADE was part of the International Joint Conference on Automated Reasoning, hence "JC" for "Joint Conference".

against which progress can be judged. System descriptions for the entered systems are in (Sutcliffe, 2001a) and on the WWW site given below. Short descriptions of the division winners are given in Section 5.

The design and procedures of CASC-JC evolved from those of CASCs-13 to -17. The major changes over the years have been:

- To expand the range of system and problem types considered, as soon as the necessary underlying infrastructure had been developed. CASC13 had only two divisions, and CASC-JC had five.

- To refine the mechanisms used to determine what problems are used. Aspects of this include selecting problems of appropriate difficulty, removing problems with aberrant encoding characteristics, and controlling the effects of artifacts of the source of problems. Improvements in these aspects have been motivated by observations and analysis of successive CASCs' result data.

- To adjust the focus of the evaluation criteria, so as to reflect the realities of ATP research and usage. For example, CASC-JC was the first CASC to have a ranking based on proof output - see Section 2.

- To adjust the organizational requirements in a manner that encourages and requires entrants to produce easily installed and robustly executing ATP systems.

Details and motivations for changes since CASC-17 are given in (Sutcliffe, 2001a)

CASC-JC was organized by Geoff Sutcliffe and Christian Suttner, and was overseen by a panel consisting of Maria Paola Bonacina, Claude Kirchner, Jeff Pelletier, and Toby Walsh. The competition machines were supplied by Technische Universität München. The CASC-JC WWW site provides access to information and data used before, during, and after the event: `http://www.cs.miami.edu/~tptp/CASC/JC/`

2. Divisions

CASC-JC was run in divisions according to system and problem characteristics. There were five competition divisions in which the systems were explicitly ranked, according to the numbers of problems solved, with ties decided by average CPU times over problems solved. There was also a demonstration division, in which systems could demonstrate their abilities without being formally ranked.

Table I. The ATP systems and entrants

ATP System Divisions entered	Entrants *Affiliation*
Bliksem 1.12 MIX* UEQ FOF	Hans de Nivelle *Max-Planck-Institut für Informatik, Germany*
DCTP 0.1 MIX SAT EPR	Gernot Stenz, Reinhold Letz *Technische Universität München, Germany*
E 0.62 MIX UEQ EPR	Stephan Schulz *Technische Universität München, Germany*
EP 0.62 MIX*	A composition of E 0.62 and a proof presentation tool, for the MIX division Proof class
E 0.6 MIX	Entered as winner of the CASC-17 MIX division
E-SETHEO csp01 MIX SAT FOF EPR	Gernot Stenz, Reinhold Letz, Stephan Schulz *Technische Universität München, Germany*
Gandalf c-2.3 MIX UEQ EPR	Tanel Tammet *Tallinn Tech. Uni., Estonia* and *Safelogic, Sweden*
GandalfFOF c-2.3 FOF	A variant of Gandalf c-2.3 using Otter for conversion to CNF, for the FOF division
GandalfSat 1.1 SAT	A variant of Gandalf, for the SAT division
GandalfSat 1.0 SAT	Entered as winner of the CASC-17 SAT division
MACE 2.0 SAT	William McCune, Larry Wos, Bob Veroff *Argonne National Laboratory, USA*
MUSCADET 2.3 FOF	Dominique Pastre *Université René Descartes, France*
Otter 3.2 MIX* UEQ FOF	William McCune, Larry Wos, Bob Veroff *Argonne National Laboratory, USA*
Otter-MACE 3.2-2.0 EPR	A composition of Otter 3.2 and MACE 2.0, for the EPR division
PizEAndSATO 0.2 EPR Demonstration	Geoff Sutcliffe, Stephan Schulz *University of Miami, USA* and *Technische Universität München, Germany*
SCOTT 6.0.0 MIX* UEQ SAT FOF EPR	John Slaney, Kal Hodgson *Australian National University, Australia*
Vampire 2.0 MIX* UEQ	Alexandre Riazanov, Andrei Voronkov *University of Manchester, England*
VampireEPR 2.0 EPR	A variant of Vampire 2.0, for the EPR division
VampireFOF 2.0 FOF	A variant of Vampire 2.0, for the FOF division
VampireJC 2.0 MIX*	A variant of Vampire 2.0, for the MIX division
VampireFOF 1.0 FOF	Entered as winner of the CASC-17 FOF division
Waldmeister 601 UEQ	Thomas Hillenbrand, B. Loechner, A. Jaeger, A. Buch *Max-Planck-Institut für Informatik* and *Universität Kaiserslautern, Germany*
Waldmeister 600 UEQ	Entered as winner of the CASC-17 UEQ division

MIX* indicates participation in the MIX division Proof class - see Section 2.

The **MIX** division used mixed CNF really-non-propositional theorems. *Mixed* means Horn and non-Horn problems, with or without equality, but not unit equality problems (see the UEQ division below). *Really-non-propositional* means with an infinite Herbrand universe (so that the problems cannot necessarily be solved by finite saturation methods). The MIX division had five problem categories: **HNE** - Horn with No Equality, **HEQ** - Horn with some (not pure) Equality, **NNE** - Non-Horn with No Equality, **NEQ** - Non-Horn with some (not pure) Equality, and **PEQ** - Pure Equality. The MIX division had two ranking classes: the **Assurance** class - ranked according to the number of problems solved (a "yes" output, giving an *assurance* of the existence of a proof), and the **Proof** class - ranked according to the number of problems solved with an *acceptable proof* output. The competition panel judged whether or not each system's proof format is acceptable. CASC-JC was the first CASC to have a ranking based on proof output, motivated by an observed need for such output in applications of ATP.

The **UEQ** division used unit equality CNF really-non-propositional theorems.

The **SAT** division used CNF really-non-propositional non-theorems. The SAT division had two problem categories: **SNE** - SAT with No Equality, and **SEQ** - SAT with Equality.

The **FOF** division used FOF non-propositional theorems. *FOF* means "natural" First Order Form, including quantifiers. The FOF division had two problem categories: **FNE** - FOF with No Equality, and **FEQ** - FOF with Equality.

The **EPR** division used CNF effectively propositional theorems and non-theorems. *Effectively propositional* means syntactically non-propositional but with a finite Herbrand universe. Prior to CASC-17 these problems were used in the MIX division, but they were excluded in CASC-17 because they have highly distinctive characteristics. However, "real world" applications often produce such problems, and therefore they are of interest. Therefore this new EPR (effectively propositional) division was added to CASC-JC. The EPR division had two problem categories: **EPT** - Effectively Propositional Theorems (unsatisfiable clause sets), and **EPS** - Effectively Propositional non-theorems (Satisfiable clause sets).

3. Organization

CASC-JC was run on 25 SUN UltraSparc IIi workstations, each having a 440 MHz UltraSparc II CPU, 256MB memory, and the SunOS 5.8 operating system.

The problems were taken from the TPTP Problem Library (Sutcliffe and Suttner, 1998), v2.4.0. There had been concern that in previous CASCs some systems had been overtuned to the TPTP problems. Such overtuning has the potential to improve an ATP system's ability to solve only TPTP problems, and not produce generally applicable advances. Therefore, TPTP v2.4.0 was not released until after CASC-JC, so that the systems could not be tuned for the new problems in TPTP v2.4.0. Overtuning 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.

Unbiased TPTP problems with a TPTP difficulty rating in the range 0.21 to 0.99 were eligible for use in all divisions. In addition, in order to make sufficient problems eligible, in the UEQ division problems with difficulty 1.00 (i.e., not yet solved by any system in normal testing) were also eligible, and in the EPR division problems with difficulty down to 0.16 were also eligible. The problems used were randomly selected from the eligible problems at the start of the competition, based on a seed supplied by the competition panel. A limiting procedure (Sutcliffe, 2000) was used to prevent the selection of an excessive number of very similar problems for any division or category. The selection mechanism was biased to select problems that were new in TPTP v2.4.0, until 50% of the problems in each category had been selected, after which random selection (from old and new problems) continued. The actual percentage of new problems used was dependent on how many new problems were eligible and the limitation on very similar problems. Table II gives the numbers of eligible problems, the maximal numbers that could be used after taking into account the limitation on very similar problems, and the numbers of problems used, in each division and category. Due to the small maximal number of usable problems in the EPS category, the limitation on the number of very similar problems could not be imposed. To ensure that no system received an advantage or disadvantage due to the specific presentation of the problems in the TPTP, the `tptp2X` utility was used to replace all predicate and function symbols with new symbols, randomly reorder the formulae and the clauses' literals, and randomly reverse the unit equalities in the UEQ problems.

The ATP systems were required to be sound and fully automatic. The organizers tested for soundness by submitting non-theorems to the systems participating in the MIX, UEQ, FOF, and EPR divisions, and theorems to the systems participating in the SAT and EPR divisions. Claiming to have found a proof of a non-theorem or a disproof of a theorem indicates unsoundness. One system failed this test and was repaired. Fully automatic operation meant that any command line

Table II. Numbers of eligible and used problems

Division			MIX			UEQ
Category	HNE	HEQ	NNE	NEQ	PEQ	
Eligible	95	72	47	565	64	114
New eligible	2	4	3	263	0	1
Max usable	36	69	27	565	64	114
Used	20	30	20	30	20	90
New used	2	4	3	10	0	1

Division	SAT		FOF		EPR	
Category	SNE	SEQ	FNE	FEQ	EPT	EPS
Eligible	80	145	151	463	25	78
New eligible	59	105	1	246	0	5
Max usable	55	97	151	463	25	8
Used	40	50	40	50	25	25
New used	25	31	1	16	0	5

switches had to be the same for all problems. A 300 second CPU time limit was imposed on each solution attempt.

4. Results

For each ATP system, for each problem, three items of data were recorded: whether or not a solution was found, the CPU time taken, and whether or not a solution (proof or model) was output. This section summarizes the results, and provides some analysis. The CPU times taken by each system, for each problem, are available from (Sutcliffe et al., 2001) or the CASC-JC WWW site.

4.1. The MIX Division

Table III summarizes the results in the MIX division. Due to the very close performances of E-SETHEO and VampireJC, the competition panel declared a tie in the Assurance class.[2] E-SETHEO does not output proofs, therefore VampireJC was the winner of the Proof class. The superior performances of the new systems, compared to that of the CASC-17 winner, E 0.6, indicates that progress has been made in the MIX division since CASC-17.

[2] Some systems, including VampireJC and Bliksem, may have solved some problems within the CPU time limit, but exceeded the CPU time limit while building a proof. Those solutions were hence not counted for the rankings. Variants of these systems that do not attempt to build a proof might have performed better in the Assurance class, as, e.g., E did, relative to EP.

Table III. MIX division results

ATP System	MIX /120	Average time	Proof output?	HNE /20	HEQ /30	NNE /20	NEQ /30	PEQ /20
E-SETHEO	93	38.7	no	18	19	15	22	19
VampireJC	93	43.4	yes	17	19	13	25	19
E 0.62	84	36.6	no	16	20	11	19	18
E 0.6	81	34.7	no	15	20	10	18	18
Vampire	76	40.5	yes	15	13	10	21	17
EP	73	36.2	yes	15	18	8	17	15
Gandalf	61	56.2	yes	15	9	10	22	5
Otter	31	34.2	yes	2	7	6	8	8
SCOTT	30	77.7	yes	2	8	7	6	7
Bliksem	29	66.9	yes	5	1	4	6	13
DCTP	14	18.8	no	2	0	5	7	0

The rankings in the HNE, NNE, and PEQ categories align quite closely with the division ranking. The NEQ category ranking aligns least with the division ranking. The NEQ category is the only category in which VampireJC outperforms E-SETHEO. Vampire's stronger performance in the NEQ category apparently stems from its very efficient data structures for handling multi-literal clauses. These data structures, which are optimized for dealing with the symmetry of equality, allow simplifying inferences, e.g., subsumption resolution, to be performed simultaneously across all stored clauses. These simplifications result in changing the clause selection mechanism, as simplified clauses have better chances of being selected and are likely to contribute to the proof. The HEQ category is the only category in which E-SETHEO and VampireJC are not the two top systems. E's stronger performance in the HEQ category comes from the combination of its strength in handling unit-equational theories and good literal selection heuristics. This synergetic effect is boosted by E's use of negative literal selection for Horn problems, so that an important part of the proof search is conducted in the unit-equational fragment.

The individual problem results show that no problems were solved by all the systems. Eleven problems, spread across the HEQ, NNE, and NEQ categories, were unsolved. Of those, the four problems in the HEQ category have been solved by E in normal testing. E's failure to solve them in CASC-JC confirms a previous observation (Sutcliffe, 2001b) that E is sensitive to the reordering done to the problems for CASC. Two of the three unsolved problems in the NNE category have been solved by Vampire in normal testing, and again the reordering appears to be the cause of failure in the competition. The remaining five problems have been solved by systems that were not entered into CASC-JC,

namely S-SETHEO (now replaced by E-SETHEO), an earlier version of Gandalf, FDP (Baumgartner, 2000), and SPASS (Weidenbach et al., 1996).

All the systems except Otter and DCTP solved problems in times close to the CPU time limit. This is in contrast to previous CASCs, where almost all systems appeared to have reached their performance limits within the CPU time allowed. A higher time limit may be appropriate for the MIX division in future competitions.

Although the overall performances of E-SETHEO and VampireJC were very close, they did not solve very similar sets of problems. On the other hand, as might be expected, E 0.62 and E 0.6 solved almost the same problems, with E 0.62 subsuming (solving a superset of the problems solved by) E 0.6 and EP. E-SETHEO subsumed EP and DCTP, and VampireJC subsumed DCTP. No other system subsumed another, indicating that most of the systems have some unique abilities. It is pleasant to note that DCTP, which is a new system based on the disconnection calculus (Letz and Stenz, 2001), solved some problems much faster than the more established resolution and superposition based systems, e.g., DCTP solved the NEQ problem `SET019-4` in 0.4 seconds, while VampireJC, the top NEQ system, took 60.6 seconds.

4.2. The UEQ Division

Table IV summarizes the results in the UEQ division. As was the case in CASCs-14 to -17, Waldmeister is the winner. Waldmeister 601 solved the same problems as the CASC-17 winner, Waldmeister 600, but with a slightly lower average CPU time. It seems that very little progress has been made in the UEQ division since CASC-17.

Table IV. UEQ division results

ATP System	UEQ /90	Average time	Proof output?
Waldmeister 601	69	9.6	yes
Waldmeister 600	69	12.0	yes
E	43	34.3	no
SCOTT	23	93.5	yes
Otter	22	22.2	no
Bliksem	13	40.8	yes
Vampire	8	66.8	yes
Gandalf	7	108.1	yes

The individual problem results show that no problems were solved by all the systems. None of the twenty problems of rating 1.00 were

solved, but all of the other problems were solved by at least one system. Nineteen problems were solved by only the Waldmeister systems. The dominance of the Waldmeister systems is clear both in terms of problems solved and times taken, and only for some harder problems does Waldmeister 601 outperform the older version. It is interesting to note that the one new problem, `LAT038-1`, was solved by four systems, but not by either of the Waldmeister systems. Waldmeister failed to solve this problem because two search strategies were selected for use, and although the first strategy would have produced a solution within the CPU time limit, Waldmeister swapped to the second strategy before the solution was found. The Waldmeisters subsume Gandalf and Vampire, and E also subsumes Vampire.

4.3. The SAT Division

Table V summarizes the results in the SAT division. For the first time ever, in any division, the previous CASC's division winner, here GandalfSat 1.0, outperformed the new systems, including the new version of the same system.[3] Therefore no winner was announced. In contrast to the other divisions, where the winners all solved more than 75% of the problems, GandalfSat 1.0 solved only 53% of the problems. This difference in success rate may be partially attributable to "harder" problems in the SAT division - they had an average difficulty rating of 0.57, as opposed to 0.49, 0.54, 0.47, and 0.31 in the MIX, UEQ, FOF, and EPR divisions, respectively. An interesting feature is the extremely low average CPU time of SCOTT. The FINDER (Slaney, 1994) component within SCOTT is a fast finite domain constraint solver, used for finding models to guide SCOTT's theorem proving component (Otter) when SCOTT is searching for proofs. For many problems in the SAT division, FINDER quickly found a model by itself, and for others the deduction steps performed by Otter quickly guided FINDER to a model.

The individual problem results show that no problems were solved by all the systems, and twelve problems were unsolved. Of those, nine have been solved by SPASS (Weidenbach et al., 1996), which was not entered into CASC-JC, two have been solved by an older version of MACE, and one has been solved by SCOTT (the clause reordering is assumed to be the cause of SCOTT's failure in CASC-JC). The low CPU times of SCOTT are evident in the individual problem times: over problems solved by both SCOTT and GandalfSat 1.0, SCOTT is an order of magnitude faster. No system subsumes another.

[3] Subsequent email from the system developer suggests that he had neglected to enable certain features in the new version, so the new system did not perform as well as it was intended to.

Table V. SAT division and category results

ATP System	SAT /90	Avg time	SNE /40	Avg time	SEQ /50	Avg time	Model output?
GandalfSat 1.0	48	15.9	27	10.8	21	22.5	no
GandalfSat 1.1	46	26.7	22	14.0	24	38.3	no
E-SETHEO	44	35.5	21	54.7	23	18.0	no
SCOTT	41	1.5	17	2.1	24	1.2	yes
MACE	25	20.4	8	47.3	17	7.8	yes
DCTP	20	10.1	19	10.7	1	0.0	no

4.4. The FOF Division

Table VI summarizes the results in the FOF division. All the systems except MUSCADET work by converting to CNF and producing a CNF refutation. The second place system, VampireFOF 1.0, is the winner of the FOF division from CASC-17, suggesting that only modest progress has been made in the FOF division since CASC-17. VampireFOF 1.0 outperformed VampireFOF 2.0, apparently because the new version uses a new experimental clausifier, while the old version uses FLOTTER (Weidenbach et al., 1996). The new clausifier has some advanced optimizations on the formula level that FLOTTER does not have, but has a very primitive clausification algorithm compared to FLOTTER's.

Table VI. FOF division and category results

ATP System	FOF /90	Avg time	FNE /40	Avg time	FEQ /50	Avg time	Proof output?
E-SETHEO	75	17.6	40	2.7	35	34.7	no
VampireFOF 1.0	72	8.6	39	4.0	33	14.0	yes
VampireFOF 2.0	71	27.3	39	12.7	32	45.5	yes
GandalfFOF	68	32.5	30	7.2	38	52.4	yes
Otter	43	21.6	27	0.1	16	58.0	no
SCOTT	39	17.6	28	13.0	11	29.2	yes
Bliksem	34	9.4	26	0.7	8	37.7	yes
MUSCADET	18	0.9	2	0.3	16	1.0	no

The individual problem results show that three problems were solved by all the systems, and one problem was unsolved. In the FNE category, all except one of the problems (`SWV014+1`) had a finite Herbrand universe. Most of the FNE problems were solved quickly by all systems except MUSCADET, indicating that conversion to CNF is effective for such problems. MUSCADET's specialization to the FEQ category is highlighted by its solution of five FEQ set theory problems that were

not solved by any other system. Another interesting aspect of MUSCADET's performance is its consistently very low average CPU times, on the problems that it can solve. E-SETHEO subsumes SCOTT, GandalfFOF subsumes Otter, and VampireFOF subsumes Bliksem.

4.5. The EPR Division

Table VII summarizes the results in the EPR division. For this type of problem E-SETHEO relies largely on grounding and invoking a propositional decision procedure, using first-order techniques only when the grounding program fails because there are too many ground instances. PizEAndSATO, which also ran on the general hardware, uses only the grounding approach. Evidently the grounding approach is effective for this type of problem.

Table VII. EPR division and category results

ATP System	EPR /50	Avg time	EPT /25	Avg time	EPS /25	Avg time	Proof output?	Model output?
E-SETHEO	49	20.5	24	32.4	25	9.1	no	no
Otter-MACE	28	25.9	11	57.1	17	5.6	no	yes
VampireEPR	27	35.9	19	26.3	8	58.7	yes	no
DCTP	20	14.4	4	28.6	16	10.9	no	no
E	17	24.7	8	38.5	9	12.5	no	no
SCOTT	15	10.7	8	19.4	7	0.8	yes	yes
Gandalf	14	67.9	14	67.9	0	-	no	-
Demonstration division								
PizEAndSATO	41	4.8	19	1.8	22	7.4	no	no

The individual problem results show that no problems were solved by all the systems (including PizEAndSATO), and no problems were unsolved. E-SETHEO subsumes DCTP, E, Gandalf, and Otter-MACE.

5. Winning System Descriptions

VampireJC 2.0, the MIX division Proof class winner and a MIX division Assurance class co-winner, is a system for first-order classical logic. It implements the calculi of ordered binary resolution, hyperresolution, and superposition for handling equality. The splitting rule is simulated by introducing new predicate symbols. A number of standard redundancy criteria and simplification techniques are used for pruning the search space: subsumption, tautology deletion, subsumption resolution, and rewriting by ordered unit equalities. A number

of efficient indexing techniques are used to implement all major operations on sets of terms and clauses, such as an improved version of code trees (Riazanov and Voronkov, 2000) for forward subsumption, and a combination of path indexing (Stickel, 1989) and database joins for backward subsumption. In a preprocessing stage, VampireJC exploits a number of elementary techniques, such as elimination of simple predicate and function definitions. VampireJC adjusts its search strategy based on some syntactic properties of the problem, such as presence of multiliteral, non-Horn and ground clauses, equations and non-equational literals. Vampire is implemented in C++, and is available at `http://www.cs.man.ac.uk/~riazanoa/Vampire`

E-SETHEO csp01, a MIX division Assurance class co-winner, the FOF division winner, and the EPR division winner, is a compositional theorem prover for first-order classical logic. Its principal components are the superposition prover E (Schulz, 2001), the model-elimination prover SETHEO (Moser et al., 1997), and the disconnection prover DCTP (Letz and Stenz, 2001). It also includes a grounding procedure and a propositional prover for near-propositional and propositional proof tasks, and uses FLOTTER (Weidenbach et al., 1996) to transform FOF problems into CNF. E-SETHEO first classifies the given problem into one of a set of predetermined categories, and selects a corresponding *schedule* that assigns resources to the different component systems. The components are then invoked sequentially with the predetermined resource limits, and try to solve the proof tasks individually. Schedules are computed automatically (using a combination of genetic algorithms and hill climbing) from results of the different strategies on a test set. E-SETHEO runs under Solaris 2.6, and sources are available by emailing `stenzg@informatik.tu-muenchen.de`.

Waldmeister 601, the UEQ division winner, is a system for unit equational deduction. Its theoretical basis is unfailing completion, in the sense of (Bachmair et al., 1989), with refinements towards ordered completion. The prover saturates the input axiomatization in a repeated cycle that works on a set of active and passive facts. The selection of the reduction ordering and the heuristic guidance of the proof search are described in (Hillenbrand et al., 1999). Recently, stronger redundancy criteria have been integrated, including ground joinability tests with ordering constraints on variables (Avenhaus et al., 2000). In several problem domains this technique is helpful especially for harder proof tasks, as can be seen in the competition results when comparing the system with last year's version. The Waldmeister WWW page is located at `http://www-avenhaus.informatik.uni-kl.de/waldmeister`

6. Conclusion

The IJCAR ATP System Competition was the sixth large scale competition for first-order ATP systems. Two significant changes to the competition design produced positive outcomes. First, the use of unseen problems provided disincentive for excessive tuning to old TPTP problems. Second, the ranking of systems in the MIX division, according to the number of proofs output, stimulated interest and research into proof production.

A positive aspect of CASC-JC was the level of enthusiasm and interest from both entrants and observers. The entrants made significant efforts to meet the requirements imposed by the competition design, and as a result the systems were more robust and usable than in the past. In the environment of the combined IJCAR conference, observers with a broad range of perspectives showed interest in the competition and its outcomes. In particular, it was pleasing to see some commercial interest in the best performing systems.

The organizers believe that the competition fulfilled its main motivations. It stimulated the development of improved ATP systems, and provided an environment where the developers could exchange ideas for further improvements. For the entrants, their research groups, and their systems, there has been substantial publicity both within and outside the ATP community. The competition has provided an overview of the abilities of running, fully automatic, ATP systems.

6.1. Future ATP System Competitions

For CASC-18 it is planned to rank more divisions by number of problems solved with a solution output (proof or model). An increased CPU time limit will be used in the MIX division, and it will be noted when a system has solved a problem but runs past the CPU time limit while building the solution. In the long term, some form of automated proof and model checking is envisioned.

As is always the case, it is hoped that the TPTP will continue to grow, so that many new problems will be available for use.

References

Avenhaus, J., T. Hillenbrand, and B. Löchner: 2000, 'On Using Ground Joinable Equations in Equational Theorem Proving'. In: P. Baumgartner and H. Zhang (eds.): *Proceedings of the 3rd International Workshop on First Order Theorem Proving*. pp. 33–43.

Bachmair, L., N. Dershowitz, and D. Plaisted: 1989, 'Completion Without Failure'. In: H. Ait-Kaci and M. Nivat (eds.): *Resolution of Equations in Algebraic Structures*. Academic Press, pp. 1–30.

Baumgartner, P.: 2000, 'FDPLL - A First Order Davis-Putnam-Logeman-Loveland Procedure'. In: D. McAllester (ed.): *Proceedings of the 17th International Conference on Automated Deduction*. pp. 200–219, Springer-Verlag.

Hillenbrand, T., A. Jaeger, and B. Löchner: 1999, 'Waldmeister - Improvements in Performance and Ease of Use'. In: H. Ganzinger (ed.): *Proceedings of the 16th International Conference on Automated Deduction*. pp. 232–236, Springer-Verlag.

Letz, R. and G. Stenz: 2001, 'System Description: DCTP - A Disconnection Calculus Theorem Prover'. In: R. Gore, A. Leitsch, and T. Nipkow (eds.): *Proceedings of the International Joint Conference on Automated Reasoning*. pp. 381–385, Springer-Verlag.

Moser, M., O. Ibens, R. Letz, J. Steinbach, C. Goller, J. Schumann, and K. Mayr: 1997, 'SETHEO and E-SETHEO: The CADE-13 Systems'. *Journal of Automated Reasoning* **18**(2), 237–246.

Riazanov, A. and A. Voronkov: 2000, 'Partiallly Adaptive Code Trees'. In: M. Ojeda-Aciego, I. de Guzman, G. Brewka, and L. Pereira (eds.): *Proceedings of the 7th European Workshop on Logics in Artificial Inteligence*. pp. 209–223, Springer-Verlag.

Schulz, S.: 2001, 'System Abstract: E 0.61'. In: R. Gore, A. Leitsch, and T. Nipkow (eds.): *Proceedings of the International Joint Conference on Automated Reasoning*. pp. 370–375, Springer-Verlag.

Slaney, J.: 1994, 'FINDER: Finite Domain Enumerator, System Description'. In: A. Bundy (ed.): *Proceedings of the 12th International Conference on Automated Deduction*. pp. 798–801, Springer-Verlag.

Stickel, M.: 1989, 'A Prolog Technology Theorem Prover: A New Exposition and Implementation in Prolog'. Technical Report Technical Note 464, SRI International, Menlo Park, USA.

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

Sutcliffe, G.: 2001a, *Proceedings of the IJCAR ATP System Competition*. Siena, Italy.

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

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

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

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

Sutcliffe, G., C. Suttner, and J. Pelletier: 2001, 'The IJCAR ATP System Competition: Allthe Details'. Technical Report UM-CSC-2001-001, Department of Computer Science, University of Miami, Miami, USA.

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

Weidenbach, C., B. Gaede, and G. Rock: 1996, 'SPASS and FLOTTER'. In: M. McRobbie and J. Slaney (eds.): *Proceedings of the 13th International Conference on Automated Deduction*. pp. 141–145, Springer-Verlag.