The CADE-18 ATP System Competition

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

C.B. Suttner
(christian@suttner.info)
Cirrus Management, Germany

Abstract. The CADE ATP System Competition (CASC) is an annual evaluation of fully automatic, first-order Automated Theorem Proving systems. CASC-18 was the seventh competition in the CASC series. Twenty-four ATP system variants competed in the various competition and demonstration divisions. An outline of the design, and a commentated summary of the results, are presented.

Keywords: competition, automated theorem proving

1. Introduction

The CADE ATP System Competition (CASC) is an annual evaluation of fully automatic, first-order Automated Theorem Proving (ATP) systems. 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 both within and outside the ATP community.

CASC-18 was the seventh competition in the CASC series - see (Sutcliffe et al., 2002) and citations therein. Twenty-four ATP system variants, listed in Table I, competed in the various competition and demonstration divisions of CASC-18. The division winners of CASC-JC (the previous CASC) were automatically entered to provide benchmarks against which progress can be judged (but due to special software requirements E-SETHEO csp01 had to be withdrawn). Details of the CASC-18 design are in (Sutcliffe, 2002) and on the CASC-18 WWW site.[1]

CASC-18 was overseen by a panel consisting of Alan Bundy, John Harrison, and Jeff Pelletier. The competition machines were supplied

[1] The CASC design has by now evolved to a sophisticated state. Many alternatives, which spring to mind, have been carefully considered, and adopted or rejected with good reason.

by the University of Manchester. The CASC-18 WWW site provides access to the systems and competition resources:

`http://www.cs.miami.edu/~tptp/CASC/18/`

Table I. The ATP systems and entrants

ATP System	Divisions	Entrants
Bliksem 1.12a	MIX* UEQ FOF	Hans de Nivelle
DCTP 1.2	MIX	Gernot Stenz
DCTP 1.2-SAT	SAT	*DCTP 1.2 variant*
DCTP 1.2-EPR	EPR	*DCTP 1.2 variant*
DCTP 10.1p	MIX FOF EPR	Gernot Stenz
DCTP 10.1p-SAT	SAT	*DCTP 10.1p variant*
E 0.7	MIX UEQ	Stephan Schulz
EP 0.7	MIX*	*E 0.7 extension*
E-SETHEO csp02	MIX FOF EPR	Gernot Stenz, Reinhold Letz, Stephan Schulz
E-SETHEO csp02-SAT	SAT	E-SETHEO csp02 variant
E-SETHEO csp01	Withdrawn	CASC-JC MIX FOF EPR winner
EXLOG 2	MIX UEQ SAT FOF EPR demo	Ivan Kossey
Gandalf c-2.5	MIX UEQ	Tanel Tammet
Gandalf c-2.5-PROOF	MIX*	*Gandalf c-2.5 variant*
Gandalf c-2.5-SAT	SAT	*Gandalf c-2.5 variant*
GandalfSat 1.0	SAT	CASC-JC SAT winner
GrAnDe 1.1	EPR demo	Geoff Sutcliffe, Stephan Schulz
ICGNS 2002b	SAT	William McCune, Olga Shumsky Matlin, Michael Rose
Otter 3.2	MIX* UEQ FOF	William McCune
SCOTT 6.1	MIX* UEQ SAT FOF EPR	John Slaney, Kal Hodgson
Vampire 5.0	MIX* UEQ FOF EPR	Alexandre Riazanov, Andrei Voronkov
Vampire 5.0-CASC	MIX*	*Vampire 5.0 variant*
Vampire 2.0-CASC	MIX*	CASC-JC MIX* winner
Waldmeister 702	UEQ	Thomas Hillenbrand, Bernd Loechner
Waldmeister 601	UEQ	CASC-JC UEQ winner

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

2. Divisions

In CASC-18 there were five *competition divisions*, in which the systems were ranked according to the numbers of problems solved, with ties decided by average CPU times over problems solved.

- 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. *Really-non-propositional* means with an infinite Herbrand universe. 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 **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. The EPR division had two problem categories: **EPT** - Effectively Propositional Theorems (unsatisfiable clause sets), and **EPS** - Effectively Propositional non-theorems (Satisfiable clause sets).

Additionally, CASC has a *demonstration division*, in which systems demonstrate their abilities without being formally ranked.

3. Organization

The CASC-18 competition divisions were run on 42 Dell Precision 330 workstations, each having an Intel P4 993MHz CPU, 512MB memory, and the Linux 2.4.9-34 operating system. In the demonstration division, GrAnDe 1.1 ran on the competition machines, while EXLOG 2 ran on a Dell OptiPlex GX100 workstation, having an Intel P3 826MHz CPU, 256MB memory, and the Windows 2000 Professional operating system. The problems were taken from the TPTP problem library (Sutcliffe and Suttner, 1998), v2.5.0. TPTP v2.5.0 was not released until after the competition, so that new problems had not previously been seen by the entrants. See (Sutcliffe, 2002) for details of the problem selection procedures. The ATP systems were required to be sound and fully automatic. A 600 second CPU time limit was imposed on each solution attempt in the MIX division, and a 300 second limit on each solution attempt in the UEQ, SAT, FOF, and EPR divisions. A wall clock time limit of double the CPU time limit was imposed in all divisions, to limit very high memory usage that causes swapping.

4. Results

This section summarizes the results, and provides some commentary. Detailed results, including the systems' output files, are available from the CASC-18 WWW site.

4.1. The MIX Division

Tables II and III summarize the results in the MIX division. The improved performance of Vampire 5.0 over Vampire 2.0-CASC (the CASC-JC winner) illustrates the positive effects of changes in that system, including: increased limits on the signature size and the number of literals in a clause, a much higher degree of specialisation when checking ordering constraints on substitutions, an efficient implementation of backward demodulation using an indexing technique that combines path-indexing with database joins and compiled queries, and improved strategy schedules.

The top eight systems are variants of four systems (see Table I). E-SETHEO is a compositional system (a system constructed from multiple distinct ATP systems) that has, among others, E and DCTP as components. The E component solved 97 of the 148 problems solved by E-SETHEO. Vampire, E, and Gandalf are monolithic. Vampire, E-SETHEO, and Gandalf use *strategy scheduling*: a schedule is formed by allocating some fraction of the CPU time limit to each of several selected strategies, which are then run in succession until one finds a solution (or they all fail). All the systems classify a given problem according to its characteristics, and select a schedule or strategy according to the problem's class. Vampire assigns a MIX problem to one of 5 classes, according to the problem's competition category; all 5 classes occurred in CASC-18. A schedule of between 4 and 8 strategies, chosen from 28 strategies installed for the MIX division, is used. E-SETHEO's outer control system assigns a MIX problem to one of 7 classes, according to the problem's competition category and size; all 7 classes occurred in CASC-18. A schedule of between 2 and 6 strategies, chosen from 15 strategies installed for the MIX division, is used.

Table II. MIX division results

ATP System	MIX /175	Average time	Proof output?
Vampire 5.0	158	25.3	yes
Vampire 5.0-CASC	157	27.3	yes
Vampire 2.0-CASC	152	57.3	yes
E-SETHEO csp02	145	57.6	no
E 0.7	131	21.4	no
EP 0.7	129	25.6	yes
Gandalf c-2.5	129	82.9	no
Gandalf c-2.5-PROOF	125	77.5	yes
Bliksem 1.12a	88	74.5	yes
DCTP 10.1p	76	41.2	no
SCOTT 6.1	63	77.7	yes
Otter 3.2	60	56.3	yes
DCTP 1.2	56	45.0	no
Demonstration division			
EXLOG 2	24	119.0	no

All of E-SETHEO's schedules include use of E, which independently classifies the problem and selects one of its 60 strategies (see below). Thus E-SETHEO has effectively 420 schedules available, of which 40 were used in CASC-18. Gandalf assigns a MIX problem to one of 19 classes, according to the problem's competition category, size, and other characteristics; 12 of the 19 classes occurred in CASC-18. A schedule of between 7 and 14 strategies, chosen from 26 strategies installed for the MIX division, is used. E assigns a MIX problem to one of 474 classes, according to the problem's Hornness, use of equality, function symbols' arities, and groundness; 90 of the 474 classes occurred in CASC-18. A single strategy, chosen from the 60 strategies installed, is used; 36 of the 60 strategies were used in CASC-18.

Fifteen of the 22 new problems in the PEQ category are group theory problems generated by HR (Colton, 2002), a program for automatic concept and theorem formation. This was the first time that non-syntactic machine generated problems had been available for CASC, and provided an interesting "machine-vs-machine" challenge. None of the 15 were solved by all systems, and none were solved by no systems. Three systems, Vampire 5.0, Vampire 5.0-CASC, and DCTP 10.1p, solved all fifteen, and several systems solved none of them.

Table III. MIX category results

ATP System	HNE /35	HEQ /35	NNE /35	NEQ /35	PEQ /35	New /63
Vampire 5.0	33	31	33	32	29	59
Vampire 5.0-CASC	33	32	33	31	28	59
Vampire 2.0-CASC	34	31	31	30	26	53
E-SETHEO csp02	28	33	29	28	27	48
E 0.7	20	32	23	27	29	41
EP 0.7	20	31	23	26	29	40
Gandalf c-2.5	28	29	29	28	15	42
Gandalf c-2.5-PROOF	28	29	28	26	14	41
Bliksem 1.12a	17	23	22	9	17	30
DCTP 10.1p	16	20	22	15	3	28
SCOTT 6.1	13	24	15	6	5	18
Otter 3.2	16	21	10	9	4	16
DCTP 1.2	12	18	11	12	3	15
Demonstration division						
EXLOG 2	2	8	12	1	1	9

4.2. The UEQ Division

Table IV summarizes the results in the UEQ division. As was the case in CASCs-14 to -JC, Waldmeister is the winner. In (Sutcliffe et al., 2002) it was noted that there had apparently been very little improvement in the UEQ systems between CASC-17 and CASC-JC, and the same seems to be true since CASC-JC.

Table IV. UEQ division results

ATP System	UEQ /70	Average time	Proof output?
Waldmeister 702	70	3.2	yes
Waldmeister 601	70	4.1	yes
E-SETHEO csp02	40	23.9	no
E 0.7	36	15.6	no
Gandalf c-2.5	27	78.9	no
Vampire 5.0	25	76.5	yes
Otter 3.2	17	45.1	yes
SCOTT 6.1	17	130.7	yes
CiME 2	15	36.6	no
Bliksem 1.12a	11	104.2	yes
Demonstration division			
EXLOG 2	1	1.5	no

Table V. SAT division and category results

ATP System	SAT /70	Avg time	Model output?	SNE /35	SEQ /35	New /36
Gandalf c-2.5-SAT	61	11.5	no	28	33	32
ICGNS 2002b	50	0.7	yes	19	31	32
GandalfSat 1.0	44	5.7	no	23	21	26
SCOTT 6.1	37	0.0	yes	15	22	18
E-SETHEO csp02-SAT	34	36.6	no	19	15	20
DCTP 10.1p-SAT	31	2.8	no	17	14	19
DCTP 1.2-SAT	24	0.0	no	10	14	18
Demonstration division						
EXLOG 2	20	9.5	no	10	10	18

4.3. The SAT Division

Table V summarizes the results in the SAT division. The performances of Gandalf c-2.5-SAT and ICGNS 2002b are significantly better than that of the CASC-JC winner GandalfSat 1.0, indicating progress in the area. Gandalf c-2.5-SAT, a successor to GandalfSat 1.0, has benefited from the addition of two new strategies: finite model building by incremental search through function and predicate symbol interpretations, and finite model building using flattening plus non-ground splitting. Although they solved less problems, ICGNS, SCOTT, and DCTP had notably lower average solution times than the other systems.

4.4. The FOF Division

Table VI summarizes the results in the FOF division. All the systems work by converting to CNF and producing a refutation.

Table VI. FOF division and category results

ATP System	FOF /70	Avg time	Proof output?	FNE /10	FEQ /60
Vampire 5.0	55	15.9	yes	9	46
E-SETHEO csp02	46	15.3	no	9	37
DCTP 10.1p	25	39.1	no	3	22
Bliksem 1.12a	15	25.4	yes	0	15
SCOTT 6.1	15	30.7	yes	1	14
Otter 3.2	10	120.1	yes	1	9
Demonstration division					
EXLOG 2	9	9.2	no	0	9

Table VII. EPR division and category results

ATP System	EPR /70	Avg time	Proof output?	Model output?	EPT /35	EPS /35	New /24
E-SETHEO csp02	60	22.6	no	no	33	27	15
Gandalf c-2.5-SAT	58	86.1	no	no	34	24	12
DCTP 10.1p	49	39.0	no	no	25	24	16
DCTP 1.2-EPR	44	36.1	no	no	17	27	18
Vampire 5.0	32	14.6	yes	no	26	6	10
SCOTT 6.1	9	11.1	yes	yes	7	2	5
Demonstration division							
GrAnDe 1.1	38	0.3	no	no	21	17	1
EXLOG 2	25	43.5	no	no	10	15	9

4.5. The EPR Division

Table VII summarizes the results in the EPR division. E-SETHEO csp02 and Gandalf c-2.5-SAT have close overall performance. An examination of Gandalf's output files shows that Gandalf was stopped for four problems because it reached the wall clock time limit, indicating very high memory usage that caused swapping. The very low average time taken by GrAnDe (which ran on the same hardware as the competition division systems) may be useful for some applications.

Twenty three of the 24 new problems are translations of problems in modal logic (Hustadt and Schmidt, 2002). These problems are large, typically with thousands of clauses, predicate symbols, functors, and variables. The use of these problems in motivated some entrants to improve the front-ends of their systems to cope with larger problems.

5. Winning System Descriptions

Vampire 5.0 (Riazanov and Voronkov, 2002), the MIX and FOF divisions winner, is an automatic theorem prover for first-order classical logic. Its kernel implements the calculi of ordered binary resolution, with superposition for handling equality. Splitting is simulated by introducing new predicate symbols. Standard redundancy criteria and simplification techniques are used: subsumption, tautology deletion, subsumption resolution, and rewriting by ordered unit equalities. The term ordering used for the MIX and FOF divisions is a non-recursive version of the Knuth-Bendix ordering that allows efficient approximation algorithms for solving ordering constraints. A number of efficient indexing techniques are used to implement all major operations on sets of terms and clauses. The most important options that determine the kernel's search strategy are the choice of the main saturation procedure, optional simplifications, the simplification ordering, and the literal selection function. Vampire is implemented in C++, and is available at: `http://www.cs.man.ac.uk/~riazanoa/Vampire`.

Waldmeister 702, the UEQ division winner, is an implementation of unfailing Knuth-Bendix completion with extensions towards ordered completion and basicness. The system saturates the input axiomatization, distinguishing active facts, which induce a rewrite relation, and passive facts, which are the one-step conclusions of the active ones up to redundancy. The saturation process is parameterized by a reduction ordering and a heuristic assessment of passive facts. The central data structures are perfect discrimination trees for the active facts, element-wise compressions for the passive ones, and sets of rewrite successors for the conjectures. The proof search is controlled by choosing search parameters according to the algebraic structure given in the problem specification (Hillenbrand et al., 1999). Waldmeister is implemented in ANSI-C and runs under Solaris and Linux. It is available at: `http://www-avenhaus.informatik.uni-kl.de/waldmeister`.

Gandalf c-2.5-SAT, the SAT division winner, is a member of the Gandalf family of provers (Tammet, 1998), which includes systems for classical logic, type theory, intuitionistic logic, and linear logic. One of the basic ideas used in Gandalf is strategy scheduling (see Section 4.1). Additionally, selected clauses from unsuccessful runs are sometimes used in later runs. The following strategies are used for satisfiability checking: finite model building by incremental search through function and predicate symbol interpretations, ordered binary resolution (ordered by term depth) for problems not containing equality, and finite model building using MACE-style flattening plus non-ground splitting of clauses. Gandalf is implemented in Scheme and compiled to C using the Hobbit Scheme-to-C compiler. The finite model building uses the Zchaff propositional logic solver (Moskewicz et al., 2001) as an external program for one of the strategies. Gandalf is available at: `http://www.ttu.ee/it/gandalf`.

E-SETHEO csp02, the EPR division winner, is a strategy scheduling (see Section 4.1) theorem prover, combining the systems E, DCTP, and SETHEO, along with specialized procedures for propositional and near-propositional formulae. The various component systems implement different calculi and proof procedures, such as superposition, model elimination, and semantic trees (the DPLL procedure). Transformation techniques may split the formula into independent subparts or may perform ground instantiation. During strategy scheduling, one component may be used several times with different control parameters. Schedules are computed from experimental data using machine learning techniques (Stenz and Wolf, 1999). The different components are written in a variety of different programming languages including C, Prolog, and Scheme. The control component is written in Perl.

6. Conclusion

The CADE-18 ATP System Competition was the seventh large scale competition for first-order ATP systems. The improved performance of new systems over the previous years' winners in the MIX and SAT divisions, showed that there have been advances in those ATP systems. This year's winners had both improved strategies and improved implementations. Some of the non-winning systems found solutions much more quickly than the winning systems, which may make them more useful in real-time situations. The use of the very large problems in the EPR division also focused attention on the need for efficient input parsing and large capacity data structures. The need to be able to cope with such large problems in some industrial applications of ATP is noted in (Schumann, 2002).

References

Colton, S.: 2002, *Automated Theory Formation in Pure Mathematics*. 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.

Hustadt, U. and R. Schmidt: 2002, 'Using Resolution for Testing Modal Satisfiability and Building Models'. *Journal of Automated Reasoning* **28**(2), 205–232.

Moskewicz, M., C. Madigan, Y. Zhao, L. Zhang, and S. Malik: 2001, 'Chaff: Engineering an Efficient SAT Solver'. In: B. D. and L. Lavagno (eds.): *Proceedings of the 39th Design Automation Conference*. pp. 530–535.

Riazanov, A. and A. Voronkov: 2002, 'The Design and Implementation of Vampire'. *AI Communications* p. To appear.

Schumann, J.: 2002, *Automated Theorem Proving in Software Engineering*. Springer-Verlag.

Stenz, G. and A. Wolf: 1999, 'Strategy Selection by Genetic Programming'. In: A. Kumar and I. Russell (eds.): *Proceedings of the 12th Florida Artificial Intelligence Research Symposium*. pp. 346–350, AAAI Press.

Sutcliffe, G.: 2002, *Proceedings of the CADE-18 ATP System Competition*. Copenhagen, Denmark.

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., C. Suttner, and F. Pelletier: 2002, 'The IJCAR ATP System Competition'. *Journal of Automated Reasoning* **28**(3), 307–320.

Tammet, T.: 1998, 'Towards Efficient ATP Progress'. In: C. Kirchner and H. Kirchner (eds.): *Proceedings of the 15th International Conference on Automated Deduction*. pp. 427–440, Springer-Verlag.