Technical Report TR-ARP-11-00

Automated Reasoning Group, Computer Sciences Laboratory
Research School of Information Sciences and Engineering
Australian National University

December 18, 2000

Progress in Automated Theorem Proving, 1997–1999

Geoff Sutcliffe
School of Information Technology
James Cook University
Townsville, Australia
geoff@cs.jcu.edu.au

Matthias Fuchs
Automated Reasoning Group
Australian National University
Canberra, Australia
fuchs@arp.anu.edu.au

Christian Suttner
Schweppermannstr. 2
München, Germany
csuttner@antfactory.com

Abstract Despite some impressive individual achievements, the extreme difficulty of Automated Theorem Proving (ATP) means that progress in ATP is slow relative to, e.g., some aspects of commercial information technology. The (relatively) slow progress has two distinct disadvantages. First, for the researchers, it is difficult to determine if a direction of investigation is making a meaningful contribution. Second, for unaware observers, a lack of progress leads to a loss of interest and confidence in the field. A serious outcome of this loss of interest and confidence has been the withdrawal of significant funding for ATP research. In this context of slow progress, it is important that progress in ATP be measured, monitored, and recognized. This paper presents quantitative measures that show progress in ATP, from mid-1997 to the end of 1999. The measures are based on collected performance data from ATP systems.

Keywords: automated reasoning, theorem proving

1 Introduction

Automated Theorem Proving (ATP) is concerned with the development and use of systems (computer programs) that automate sound reasoning: the derivation of conclusions that follow inevitably from facts. This capability lies at the heart of many important computational tasks, e.g., software verification [Reif, 1995; Kaufmann, 1998], development of mathematical theories [McCune and Padmanabhan, 1996; Kunen, 1996], and security protocol analysis [Weidenbach, 1999]. ATP systems are presented with problems written in some logic. Classical 1st order logic is widely used because of its semi-decidability, and all references to ATP systems and problems in this work are for classical 1st order logic. The ideas presented can, however, readily be transferred to other cases.

The development of useful ATP systems started in the mid-1960s, and has progressed to a point now where current ATP systems are capable of solving non-trivial problems, e.g., EQP [McCune, 2000] solved the Robbins problem [McCune, 1997]. This progress is impressive, given that ATP is “possibly the hardest subfield of Computer Science” [Slaney, 1994]. Noteworthy landmarks in this history include:

- The resolution inference rule [Robinson, 1965].
- The series of early ATP systems developed at the Argonne National Laboratories [Lusk, 1992], which, among other contributions, introduced the ”given clause” control loop.
- Paramodulation as an alternative to the explicit use of equality axioms [Robinson and Wos, 1969].
- Subsumption as an effective means for controlling redundant information [Wos et al., 1993].
- The tableau and model elimination strategies [Furbach et al., 1998; Loveland, 1969], which are effective ATP strategies and also the basis for Prolog [Bratko, 1990].
- The Knuth-Bendix completion procedure[Knuth and Bendix, 1970] and related methods for unit equality reasoning [Wos et al., 1967; Dershowitz and Vigneron, 2000].
- Indexing techniques for highly efficient storage and access to the data structures used by ATP systems [Stickel, 1989; Ramakrishnan et al., 1999].
- The superposition inference rule [Bachmair et al., 1992].

Despite these individual achievements, the extreme difficulty of ATP means that progress in ATP is slow relative to, e.g., some aspects of commercial information technology. The (relatively) slow progress has two distinct disadvantages. First, for the researchers, it is difficult to determine if a direction of investigation is making a meaningful contribution. This is troublesome both in terms of motivation (obvious progress is always encouraging) and in terms of focus (expend more energy in directions that are successful). Second, for unaware observers, a lack of progress leads to a loss of interest and confidence in the field. An extract from a recent review of an ATP paper submitted to a reputable journal illustrates this attitude ...

> "The reviewer, just like probably most readers of the ... journal, believes that the style of systems as compared here ... (search based fully automatic TP systems), whose intellectual roots are still in the seventies and early eighties, are misguided ("intellectually frozen in a time warp of the early days and only kept alive by such a small isolated but dedicated community" is a famous value judgement by a well known worker in the field). Hence these ... are not only a waste of time, but they are counterproductive as they lure the young researcher into a style of work that is essentially wasted, and the ... authors of this paper should be forced to drink the poisoned cup of Socrates for corrupting the youth."

(Ellipses have been used only to prevent identification of the journal and reviewer - no substance has been removed. The reviewer did however recommend that the paper be accepted!) A more serious outcome of this loss of interest and confidence has been the withdrawal of significant funding for ATP research, e.g., the need for revitalized funding in the USA was highlighted in [Loveland, 1999], and in Germany the DFG Schwerpunktprogramm Deduktion ended in 1998 and has not been replaced.

In this context of slow progress, it is important that progress in ATP be measured, monitored, and recognized. This paper presents quantitative measures that show progress in ATP, from mid-1997 to the end of 1999. The measures are based on collected performance data from ATP systems. Section 2 describes the source, organization, and features of the performance data, which is then analysed in Section 3. Section 4 concludes the paper.

2 Performance Data

In order to demonstrate progress in ATP, it is necessary to evaluate ATP over time. Evaluation of individual theoretical results, implementation techniques, etc, is possible, but from a user perspective these separate contributions are of little interest. Evaluation of the final product of ATP research, that is, the combination of theoretical results, implementation techniques, etc, into ATP systems, satisfies both user and developer perspectives of progress. This work thus demonstrates progress in ATP through evaluation of ATP systems over time. Analytic approaches to ATP system evaluation, such as presented in [Dunker, 1994; Letz, 1993; Plaisted, 1994], provide insights into theoretical system capabilities. However, complete analysis of the search space at the 1st order level is of course impossible (for otherwise 1st order logic would be decidable). It is therefore necessary to make empirical evaluations of the ATP systems.

2.1 Specialist Problem Classes

An empirical evaluation of ATP systems requires a selection of ATP problems for the systems to attempt. ATP problems have easily identifiable logical, language, and syntactic characteristics. Various ATP systems and techniques have been observed to be particularly well suited to problems with certain characteristics. For example, everyone agrees that special techniques are deserved for problems with equality, and the CASC-15 results [Sutcliffe and Suttner, 1999] showed that problems with true functions, i.e., with an infinite Herbrand universe, should be treated differently from those with only constants, i.e., effectively propositional problems. Due to this specialization, empirical evaluation of ATP systems must be done in the context of problem sets that are reasonably homogeneous with respect to the systems. These problem sets are called *Specialist Problem Classes* (SPCs), and are based on problem characteristics. The choice of what problem characteristics are used to form the SPCs is based on community input and on analysis of system performance data [Fuchs and Sutcliffe, 2000]. The range of characteristics that have so far been identified as relevant are:

- Theoremhood: Theorems vs Non-theorems
- Order: Essentially propositional vs Real 1st order
- Equality: No equality vs Some equality vs Pure equality
- Form: CNF (Clause Normal Form) vs FOF (First Order Form)
- Horness: Horn vs Non-Horn
- Unit equality: Unit equality vs Non-unit pure equality

Based on these characteristics, 14 SPCs have been defined, as indicated by the leaves of the tree in Figure 1.

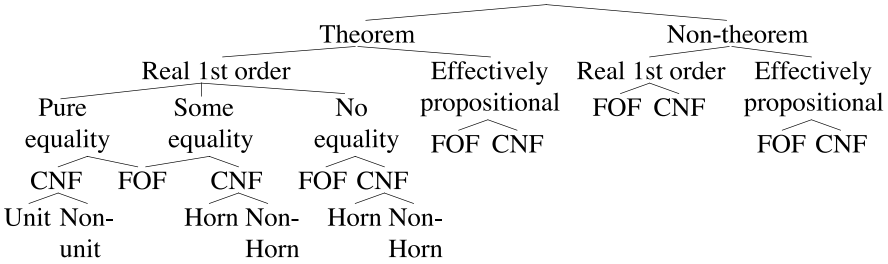


Figure 1: Specialist Problem Classes

For easy reference, the SPCs are referred to using mnemonic acronyms, abbreviating Theorem to `THM`, Non-theorem to `SAT`, Real 1st order to `RFO`, Effectively propositional to `EPR`, Pure equality to `PEQ`, Some equality to `SEQ`, No equality to `NEQ`, Unit (pure) equality to `UEQ`, Non-unit (pure) equality to `NUE`, Horn to `HRN`, and Non-Horn to `NHN`.

CNF and FOF are retained as is. For example, the SPC THM_RFO_SEQ_CNF_NHN contains theorems that are really 1st order, contain some equality, are presented in clause normal form, and are non-Horn.

2.2 The TPTP Problem Library

Currently there are not many "real" applications of ATP, and therefore there is no corpus of application problems that can be used for testing ATP systems. The TPTP (Thousands of Problems for Theorem Provers) problem library is a library of test problems for ATP systems [Sutcliffe and Suttner, 1998]. The TPTP is large enough to obtain statistical significance, and spans a diversity of subject matters. The TPTP is regularly updated with new problems, including problems from "real" applications of ATP. The TPTP is the best available collection of problems representing general purpose applications of ATP, and thus is the best source of problems for evaluating ATP systems. The TPTP also has an organizational structure designed for testing ATP systems. Since the first release of the TPTP in 1993, many researchers have used the TPTP as an appropriate and convenient basis for testing their ATP systems. Although other test problems do exist and are sometimes used, the TPTP is now the de facto standard for testing classical 1st order ATP systems.

Some researchers who have tested their ATP systems over the entire TPTP problem library have contributed their performance data to the TPTP results collection [Sutcliffe and Suttner, 2000a]. The results are for various ATP systems, various system versions, and various TPTP versions. The results collection thus provides snapshots of ATP systems' performances over time, and forms a basis for measuring progress in ATP.

2.3 System Performance Curves

The performance data in the TPTP results collection is provided by the individual system developers, which means that the systems have been tested using a range of CPU and memory resource limits. Analysis shows that the differences in resource limits do not significantly affect which problems are solved by each ATP system. Figure 2 plots the CPU times taken by several contemporary ATP systems to solve problems in the SPC THM_RFO_SEQ_CNF_NHN, for each solution found, in increasing order of time taken.[1] The relevant feature of these *performance curves* is that they are exponential in nature, as would be expected for a search problem in an exponentially growing search space (the performance curves in other SPCs have the same feature). Each system has a point at which the time taken to find solutions starts to increase dramatically. This point is called the system's *Peter Principle Point* (PPP), as it is the point at which the system has reached its level of incompetence.[2] A linear increase in the

[1] The numbers of solutions found are not comparable, as the systems attempted the SPC in different TPTP versions

[2] The Peter Principle is "The theory that employees within an organization will advance to their highest level of competence and then be promoted to and remain at a level at which they are incom-

CPU resources beyond the PPP would not lead to the solution of significantly more problems. The PPP thus defines a realistic CPU resource limit for the system. From an ATP perspective, after the PPP the search space has typically grown to a size where the system is unable to find a solution within the space. The PPP thus also defines a realistic memory resource limit for the system. Provided that enough CPU time and memory are allowed for the ATP system to pass its PPP, a usefully accurate measure of what problems it can solve within realistic resource limits is achieved. Performance curves provide a basis for evaluating the progress in ATP over time. This is described in Sections 3.1 and 3.2.

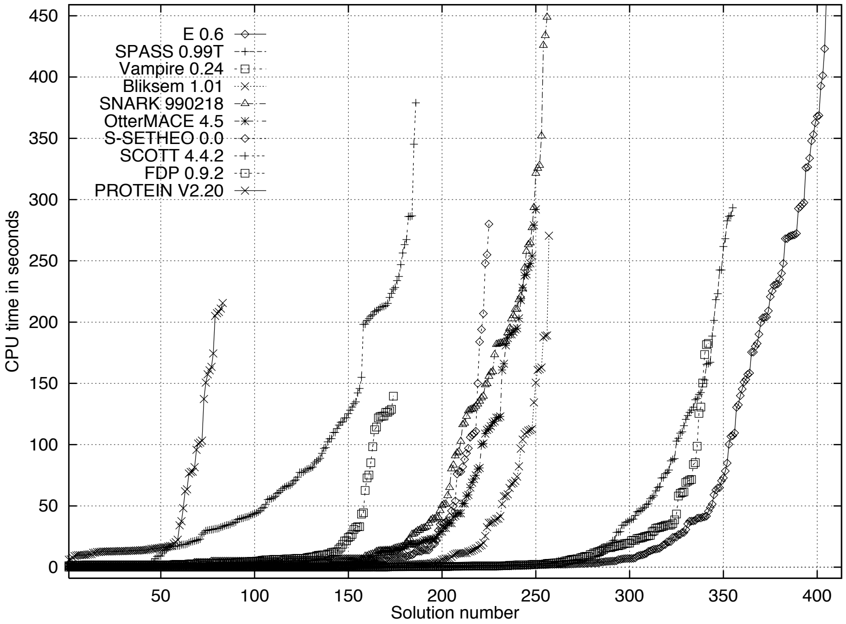


Figure 2: Proof number vs CPU time in the SPC THM_RFO_SEQ_CNF_NHN

2.4 ATP System and Problem Evaluation

[Sutcliffe and Suttner, 2000b] presents methodologies for the empirical evaluation of ATP systems and problems, within individual SPCs. The methodologies may be summarized as follows. Initially a partial ordering of the systems is determined by *subsumption*: a system that solves (within realistic resource limits) a strict superset of the problems solved by another system subsumes, and is better than, the other system. The non-subsumed systems are designated *rating contributors*. If the number of rating contributors is less than a threshold (currently three) then other high performing but subsumed systems are also made rating contributors (this use of subsumed rating contributors improves the ratings produced, as is explained in [Sutcliffe and Suttner, 2000b]). A problem is then rated according to the fraction of rating contributors that

petent." [Peter and Hull, 1969]

fail to solve the problem. Problems with a rating of zero are *easy*, with a rating between zero and one are *difficult*, and with a rating of one are *unsolved*. Finally, the ATP systems are rated according to the fraction of difficult problems they can solve. The TPTP results collection is used to rate the systems and the problems in the TPTP. The change of a problem rating from unsolved to difficult captures the point at which a problem is solved for the first time by an ATP system (according to the collected data), which is an indication of progress in ATP. This is described in Section 3.3. Overall reductions in problem ratings over time are also a measure of progress in ATP. This is described in Section 3.4.

Each year since 1996, an empirical evaluation of 1st order ATP systems has been performed at CADE [Sutcliffe and Suttner, 1997; Suttner and Sutcliffe, 1998; Sutcliffe and Suttner, 1999; Sutcliffe, 2000; Sutcliffe, 2001].[3] The CADE ATP System Competition (CASC) evaluates the performance of fully automatic ATP systems for classical 1st order logic. The evaluation is in terms of the number of problems solved and the average runtime for successful solutions, in the context of a bounded number of eligible problems chosen from the TPTP, and a specified CPU time limit for each solution attempt. The CPU time limit, and the memory in the computers used, are adequate for the ATP systems to reach their PPPs. The CASC results can be influential with regard to funding and other recognition for the ATP system developers. As a result, most of the decent contemporary ATP systems are entered, and the CASC results provide a way to show relative progress of ATP systems over time. This is described in Section 3.5.

3 Progress in ATP

To measure the progress in ATP, the performance of ATP systems has been analysed in two ways. First, the performance data in the TPTP results collection, over a two and a half year period, has been analysed. The results analysed are for problems in the TPTP versions v2.0.0, released on 5th June 1997, v2.1.0, released on 17th December 1997, v2.2.0, released on 11th February 1999, and v2.3.0, released on 16th November 1999. Second, the performance of ATP systems in CASC over a two year period has been analysed. In all cases, as is explained above, the analysis is done in the context of individual SPCs.

3.1 SOTA System Performance

In order to evaluate overall quality and progress in ATP, the individual ATP systems tested on an SPC in a TPTP version are combined to form a *state-of-the-art* (SOTA) system. For any problem, a SOTA system has the performance of the best available individual system for the problem, i.e., the time taken by the SOTA system to solve a problem is simply computed as the minimum of the times taken by the available

[3] CADE, the Conference on Automated Deduction, is the major forum for the presentation of new research in automated deduction.

individual systems. A SOTA system can really be built, by running the individual systems in competition parallel, as done in the SSCPA system [Sutcliffe and Seyfang, 1999]. A SOTA system's performance is thus a realistic measure of the combined quality of the ATP systems of the time. A comparison of the SOTA systems' performances for an SPC in two TPTP versions provides evidence of progress in ATP for that SPC. Note that the contributions of the individual systems to a SOTA system are dependent on the problems being attempted, but like the individual system performance curves, the performance curve of a SOTA system has an exponential shape.

An initial comparison of the SOTA systems for two TPTP versions can be made by comparing their raw performance on problems that were in both TPTP versions. For example, for the SPC `THM_RFO_SEQ_CNF_NHN`, there were 314 problems that were in both TPTP versions v2.0.0 and v2.3.0. The SOTA system for TPTP v2.0.0 solved 181 of the problems, with a maximal CPU time of 831 seconds, while the SOTA system for v2.3.0 solved 219 of the problems (39 additional problems that are not solved by the v2.0.0 system, less 1 unsolved problem that is solved by the v2.0.0 system[4]), with a maximal CPU time of 508 seconds. Such a raw increase in the number of problems solved between TPTP versions v2.0.0 and v2.3.0 occurs in many of the SPCs, as is shown in the first two columns of Table 1 (the other columns in Table 1 give the results of the analysis described in Section 3.2). Section 2.3 shows that these numbers of problems solved are almost independent of the different resources available to the systems, so that the extra problems solved by the v2.3.0 systems is evidence of progress in ATP. Section 3.2 refines this analysis to compensate for the small dependence on resources available.

3.2 Exponential Curve Fitting

The increase in the number of problems solved by SOTA systems, from one TPTP version to another, may be extrapolated to resource limits beyond those found in the performance data. This is achieved by fitting *exponential curves* of the form $f(x) = ae^{bx}$ to the SOTA systems' performance curves. An approximation method has been used for this. The method lets b range from a start value α to an end value β, in steps of ε. For each b, a is computed to minimize the error of fit $E(a, b)$.

$$E(a,b) = \sum_{i=1}^{n} \left(y_i - ae^{bx_i}\right)^2$$

This is achieved by letting

$$a = \frac{\sum_{i=1}^{n} y_i e^{bx_i}}{\sum_{i=1}^{n} e^{2bx_i}}$$

Among all pairs (a, b), the one with the smallest error $E(a, b)$ is selected. In this study, b ranging from $\alpha = 0$ to $\beta = 2$ in steps of $\varepsilon = 10^{-3}$ provided adequate accuracy. For

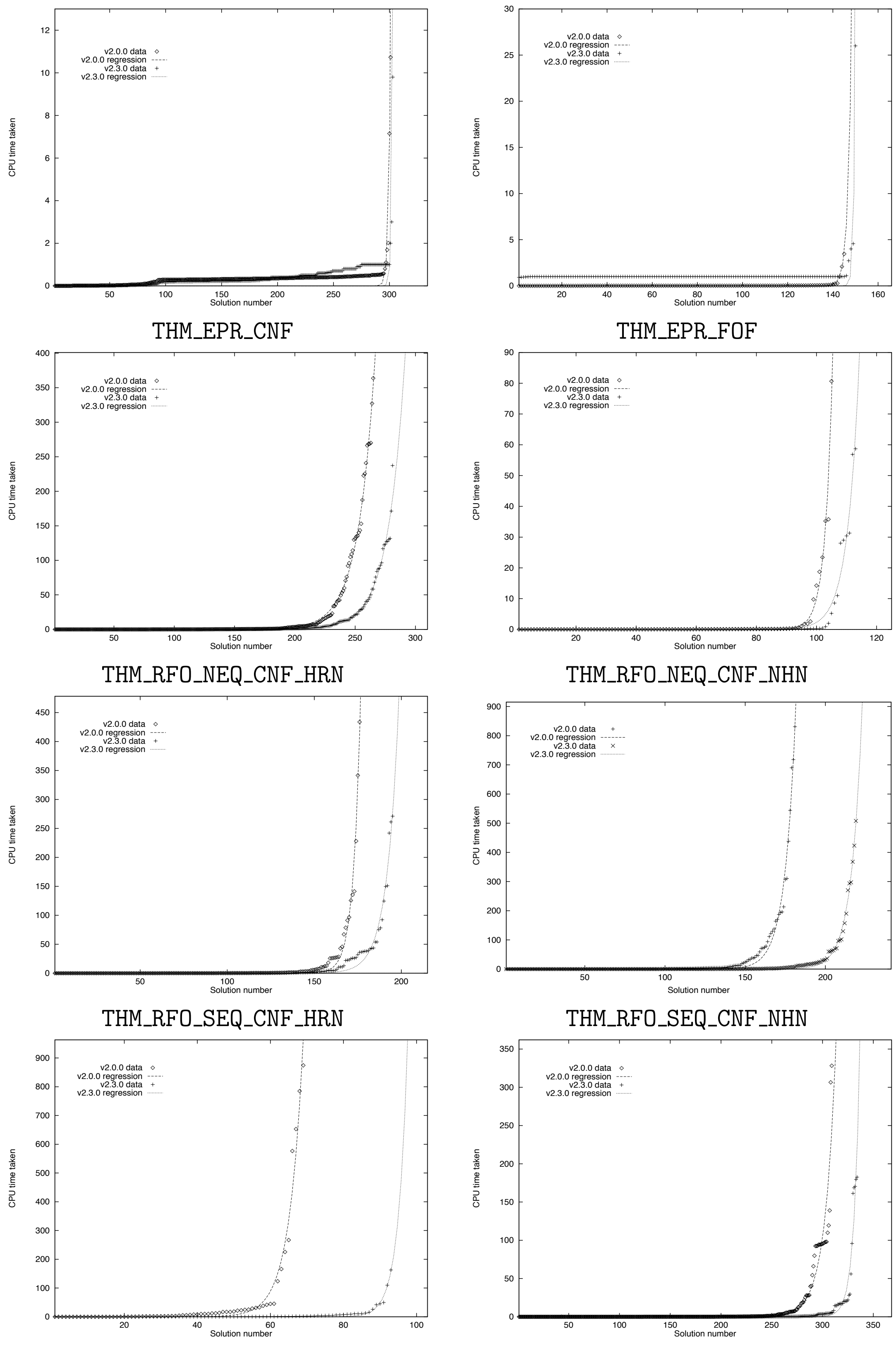
[4] Problems may change from solved to unsolved when an ATP system is improved so that it can solve more problems overall, but as a result cannot solve those particular problems, i.e., progress in a particular ATP system is not necessarily monotonic with respect to the problems it can solve.

the purpose of demonstrating progress in ATP, it is important that the exponential curves fit the performance curves most accurately at the higher solution numbers, as these data points correspond to harder ATP problems.

Exponential curves have been fitted to the performance curves of the SOTA systems for each of the SPCs in TPTP versions v2.0.0 and v2.3.0. For the SPCs involving SAT and most of the SPCs involving FOF, there was insufficient data for meaningful curve fitting and analysis. The performance and exponential curves for the other eight SPCs are shown in Figure 3, and the curve parameters are given in the first two columns of Table 1. The gaps between the steep parts of the exponential curves in the SPCs THM_RFO_*_CNF_* are visual evidence of the progress in ATP in that period, for those SPCs.

In the period between TPTP versions there may be hardware improvements that would make even an unchanged system perform better in a resource limited situation. Such hardware improvements may undermine confidence in the conclusion drawn in Section 3.1. However, the hardware improvements can be taken into account, as follows. In order to estimate the *hardware improvement factor* between two TPTP versions, the times taken to solve those problems solved by both versions' SOTA systems are extracted. The geometric average of the ratio of the times is computed, and used as an upper bound on the hardware improvement factor.[5] The computed hardware improvement factor is an upper bound because it assumes that all changes in the times taken to solve the problems are caused by hardware improvements, while in reality some portion of the changes is caused by improvements in the systems. The hardware improvement factor is used to scale the SOTA systems' exponential curves, and a comparison of the results then provides a lower bound on the progress in ATP. First, an estimate of the number of problems the old system would solve, if run on the new hardware with a time limit of the maximal time taken by the new system, is computed. To do this, the maximal time taken by the new system is scaled up by the hardware improvement factor, and the inverse of the old system's exponential curve is applied. This estimated number of problems solved can be compared to the number solved by the new system. For example, for the THM_RFO_SEQ_CNF_NHN example above, the scaled time limit is 2.23 * 508 seconds = 1132 seconds, which leads to an estimated 184 problems being solved by the v2.0.0 system, 35 less than the 219 problems solved by the v2.3.0 system. Second, an estimate of the number of problems the new system would solve, if run on the old hardware with a time limit of the maximal time taken by the old system, is computed. To do this, the maximal time taken by the old system is scaled down by the hardware improvement factor, and the inverse of the new system's exponential curve is applied. This estimated number of problems solved can be compared to the number solved by the old system. For the THM_RFO_SEQ_CNF_NHN example, the scaled time limit is 831 seconds / 2.23 = 373 seconds, which leads to an estimated 218 problems being solved by the v2.3.0 system, 37 more than the 181 problems solved by the v2.0.0 system. Third, the old system's exponential curve is used to estimate the CPU time that would be required by the old system on the old hardware to solve the number

[5] A geometric average is used rather than an arithmetic average, so that extreme ratios caused by the occasionally unstable performances of ATP systems do not have an excessive effect.



THM_EPR_CNF

THM_EPR_FOF

THM_RFO_NEQ_CNF_HRN

THM_RFO_NEQ_CNF_NHN

THM_RFO_SEQ_CNF_HRN

THM_RFO_SEQ_CNF_NHN

THM_RFO_PEQ_CNF_NUE

THM_RFO_PEQ_CNF_UEQ

Figure 3: Performance and exponential curves, for TPTP versions v2.0.0 and v2.3.0

of problems solved by the new system. The ratio of this time and the maximal time taken by the new system is the hardware improvement factor that would be required for the old system to solve the same number of problems as the new system, within the same maximal time. If this ratio is much greater than the computed hardware improvement factor, or simply unrealistic, then such a claim cannot be made. For the THM_RFO_SEQ_CNF_NHN example, 228101 seconds would have been required by the v2.0.0 system to solve 219 problems. The ratio of times is 228101/508 = 449.02, i.e., a 449 fold hardware improvement would have been required in that two and a half year period, which is not realistic.

This hardware sensitive analysis has been applied to the SOTA systems' performance curves shown in Figure 3, and the results are given in Table 1. The first two columns of Table 1 give the raw performance data and exponential curve parameters, as discussed above. The third column shows the hardware improvement factor, the maximal time taken by the v2.3.0 system scaled up by the hardware improvement factor, and the number of problems the v2.0.0 system is estimated to solve in that time. This estimated number of problems solved can be compared to the number solved by the v2.3.0 system. The fourth column shows the hardware improvement factor, the maximal time taken by the v2.0.0 system scaled down by the hardware improvement factor, and the number of problems the v2.3.0 system is estimated to solve in that time. This estimated number of problems solved can be compared to the number solved by the v2.0.0 system. The final column of the table shows the time that the v2.0.0 system is estimated to require in order to solve the number of problems solved by the v2.3.0 system. This time required can be compared to the maximal time taken by the v2.3.0 system, and the ratio is the required hardware improvement factor that is shown. For the SPCs THM_EPR_CNF and THM_EPR_CNF there is little or no evidence of progress in ATP. For the SPCs THM_RFO_*_CNF_* there is clear evidence of progress in the period between these TPTP versions - June 1997 to November 1999.

3.3 Problems Solved for the First Time

The first time solution of a problem that ATP systems had previously failed to solve is an indication of progress at the leading edge of ATP, and indicates that the solving system defines that part of the edge. This is particularly noticeable when the problem is one that humans have had an interest in, but have failed to solve. A high profile example of this was EQP's solution of the Robbins problem [McCune, 1997]. Another noteworthy example was the characterization of quasigroups using the MGTP system [Fujita et al., 1993].

The first time solution of a problem is easily detected from the TPTP problem ratings, which have been included in the TPTP since version v2.0.0. When a problem rating changes from 1.00 (unsolved) to less than 1.00 (difficult), the problem has been solved for the first time. If the problem was added to the TPTP in a version that precedes the version in which the rating dropped, then ATP systems had failed to solve the problem in the intervening period. The second to fifth columns of Table 2 show the numbers of problems solved for the first time in the periods between TPTP versions v2.0.0 and

v2.0.0 system		v2.3.0 system		v2.0.0 scaled ↑		v2.3.0 scaled ↓		v2.0.0 needs	
Time	Solved	Time	Solved	Time	Solved	Time	Solved	Time	Solved
e curve		e curve		H.I. factor		H.I. factor		H.I. factor	
THM_EPR_CNF - 304 problems									
11s	301	10s	303	10s	301	11s	304	36s	303
1.82E-77$e^{0.595x}$		4.5E-116$e^{0.884x}$		1.02		1.02		3.60	
THM_EPR_FOF - 157 problems									
4s	145	27s	150	27s	149	4s	149	83s	150
5.61E-40$e^{0.632x}$		2.30E-91$e^{1.41x}$		0.99		0.99		3.07	
THM_RFO_NEQ_CNF_HRN - 327 problems									
364s	265	238s	281	390s	267	222s	284	1098s	281
2.38E-6$e^{0.071x}$		4.09E-7$e^{0.071x}$		1.64		1.64		4.61	
THM_RFO_NEQ_CNF_NHN - 131 problems									
81s	105	59s	113	63s	105	76s	114	1848s	113
3.46E-17$e^{0.402x}$		1.74E-11$e^{0.256x}$		1.07		1.07		31.32	
THM_RFO_SEQ_CNF_HRN - 213 problems									
434s	176	272s	195	326s	176	363s	197	40801s	195
1.31E-16$e^{0.242x}$		6.69E-11$e^{0.149x}$		1.20		1.20		150.00	
THM_RFO_SEQ_CNF_NHN - 314 problems									
831s	181	508s	219	1132s	184	373s	218	228101s	219
1.91E-9$e^{0.148x}$		3.34E-12$e^{0.149x}$		2.23		2.23		449.02	
THM_RFO_PEQ_CNF_NUE - 112 problems									
875s	69	163s	93	621s	68	230s	94	558090s	93
1.11E-5$e^{0.265x}$		1.76E-14$e^{0.395x}$		3.81		3.81		3423.87	
THM_RFO_PEQ_CNF_UEQ - 358 problems									
329s	309	183s	334	291s	312	207s	334	2463s	334
7.97E-11$e^{0.093x}$		8.92E-24$e^{0.175x}$		1.59		1.59		13.46	

Table 1: Curve analysis, for TPTP versions v2.0.0 and v2.3.0

v2.1.0 (a 6 month period), v2.1.0 and v2.2.0 (a 14 month period), and v2.2.0 and v2.3.0 (a 9 month period). The numbers have been split up according to the TPTP versions in which those problems were added to the TPTP. Blank entries mean that no problems were solved for the first time, and if no new problems were solved in a period, that row of the table has been omitted. A ”-” entry means that the TPTP version in which problems were added does not precede the version in which solutions could be found. The sixth column of Table 2 gives the total for each of the periods. These numbers of problems solved for the first time include those solved due to hardware improvements. The number of problems that were in v2.0.0, and that could be expected to be solved for the first time by version v2.3.0 due to hardware improvements, can be estimated using the ideas of Section 3.2. It is either the estimated number of problems more that would be solved by the v2.0.0 SOTA system when its maximal time taken is scaled up by the hardware improvement factor, or the estimated number of problems less that would be solved by the v2.3.0 SOTA system when its maximal time taken is scaled down by the hardware improvement factor. The larger of these two numbers is used, in order to be conservative. This number can be subtracted from the total number of problems that were in v2.0.0, and that were solved for the first time. These figures are shown in the final column of Table 2.

The numbers in Table 2 show regular first time solution of problems, indicating regular progress at the leading edge of ATP. The progress is particularly evident in the `THM_RFO_*_CNF_*` SPCs. Hardware improvements account for only some fraction of the first solutions.

3.4 Decreasing Problem Ratings

While the change of a problem rating from 1.00 to less than 1.00 shows progress at the leading edge of ATP, general reductions in problem ratings show general improvement of ATP systems. A problem's rating decreases when a higher fraction of the rating contributors, as described in Section 2.4, are able to solve the problem. This may come about either by the number of rating contributors staying the same but with a new system that can solve the problem replacing a previous rating contributor, or by the number of rating contributors increasing with a new system that can solve the problem. In both cases the new system has improved the overall quality of the available ATP systems, which is progress in ATP. Note that a change in rating cannot be caused by a subsumed system.

The average problem ratings have been computed for those problems that were in all the TPTP versions being considered. These averages are shown in Table 3. For some SPCs the problems that were in v2.0.0, and hence all subsequent versions, are all easy. Thus there is no change in the average rating. For the SPCs that have some meaningful change in average rating, the rate of change as a function of time is shown in Figure 4. There is a clear overall downward trend in the problem ratings, which means that the systems are getting better and better, i.e., there is progress in ATP. The analysis of Section 3.2 shows that only a small part of the progress can be attributed to hardware improvements. In some SPCs there is an increased average rating between TPTP versions v2.0.0 and v2.1.0. This came about due to new rating contributors that could not solve problems that were solved by the existing rating contributors. This inappropriate variation in ratings has not occurred since more decent ATP systems' performance data has been collected.

3.5 CASC Fixed Points

CASC is organized into divisions, which correspond closely to the SPCs. The divisions are `MIX` - mixed CNF really 1st order theorems ("mixed" means Horn and non-Horn problems, with or without equality, but not unit equality problems), `UEQ` - unit equality CNF really 1st order theorems, `SAT` - mixed CNF really 1st order non-theorems, and `FOF` - mixed FOF really 1st order theorems. A winner is announced in each division of each CASC. For the last two CASCs (CASC-16 and CASC-17), the CASC organizers

First solved	Added to the TPTP in version						v2.0.0
in the period	v1.0.0	v1.1.0	v1.2.0	v2.0.0	v2.1.0	Totals	Total - H.I.#
`SAT_EPR_CNF`							
v2.0.0 - v2.1.0			2	-	-	2	52 - ?
v2.1.0 - v2.2.0			40	10	-	50	
v2.2.0 - v2.3.0					4	4	
`SAT_EPR_FOF`							
v2.2.0 - v2.3.0					6	6	0 - ?
`SAT_RFO_CNF`							
v2.0.0 - v2.1.0	14		2	-	-	16	23 - ?
v2.1.0 - v2.2.0	1		1	2	-	4	
v2.2.0 - v2.3.0	1		1	1	1	4	
`THM_EPR_CNF`							
v2.2.0 - v2.3.0	1		2	1	6	10	4 - 1
`THM_EPR_FOF`							
v2.2.0 - v2.3.0					4	4	0 - 1
`THM_RFO_EQU_FOF`							
v2.2.0 - v2.3.0				1		1	1 - ?
`THM_RFO_NEQ_CNF_HRN`							
v2.0.0 - v2.1.0	6	1		-	-	7	31 - 8
v2.1.0 - v2.2.0	7		2		-	9	
v2.2.0 - v2.3.0	10	5				15	
`THM_RFO_NEQ_CNF_NHN`							
v2.0.0 - v2.1.0	1	1		-	-	2	6 - 1
v2.2.0 - v2.3.0	1	2		1	21	25	
`THM_RFO_SEQ_CNF_HRN`							
v2.0.0 - v2.1.0	8	2	4	-	-	14	24 - 1
v2.1.0 - v2.2.0	1		2		-	3	
v2.2.0 - v2.3.0	4	3				7	
`THM_RFO_SEQ_CNF_NHN`							
v2.0.0 - v2.1.0	8	3	7	-	-	18	38 - 6
v2.1.0 - v2.2.0	5				-	5	
v2.2.0 - v2.3.0	13			2	30	45	
`THM_RFO_PEQ_CNF_NUE`							
v2.0.0 - v2.1.0	9	1	7	-	-	17	27 - 5
v2.1.0 - v2.2.0	1			3	-	4	
v2.2.0 - v2.3.0	4		1	1		6	
`THM_RFO_PEQ_CNF_UEQ`							
v2.0.0 - v2.1.0	12		4	-	-	16	35 - 9
v2.1.0 - v2.2.0	1				-	1	
v2.2.0 - v2.3.0	14		1	3		18	

Table 2: Numbers of problems solved for the first time

TPTP version Months since v2.0.0	v2.0.0 0	v2.1.0 6	v2.2.0 20	v2.3.0 29
SPC	Average problem rating			
SAT_EPR_CNF	0.40	0.53	0.27	0.13
SAT_EPR_FOF	0.00	0.00	0.00	0.00
SAT_RFO_CNF	0.54	0.61	0.47	0.39
SAT_RFO_FOF	0.00	0.00	0.00	0.00
THM_EPR_CNF	0.03	0.23	0.02	0.03
THM_EPR_FOF	0.00	0.00	0.00	0.00
THM_RFO_NEQ_FOF	0.00	0.00	0.00	0.00
THM_RFO_EQU_FOF	0.00	0.00	0.00	0.00
THM_RFO_NEQ_CNF_HRN	0.57	0.52	0.48	0.19
THM_RFO_NEQ_CNF_NHN	0.30	0.33	0.33	0.12
THM_RFO_SEQ_CNF_HRN	0.54	0.46	0.45	0.21
THM_RFO_SEQ_CNF_NHN	0.59	0.53	0.44	0.38
THM_RFO_PEQ_CNF_NUE	0.71	0.47	0.43	0.42
THM_RFO_PEQ_CNF_UEQ	0.46	0.35	0.32	0.07
Average of non-0 averages	0.46	0.45	0.36	0.22

Table 3: Arithmetic average of problem ratings

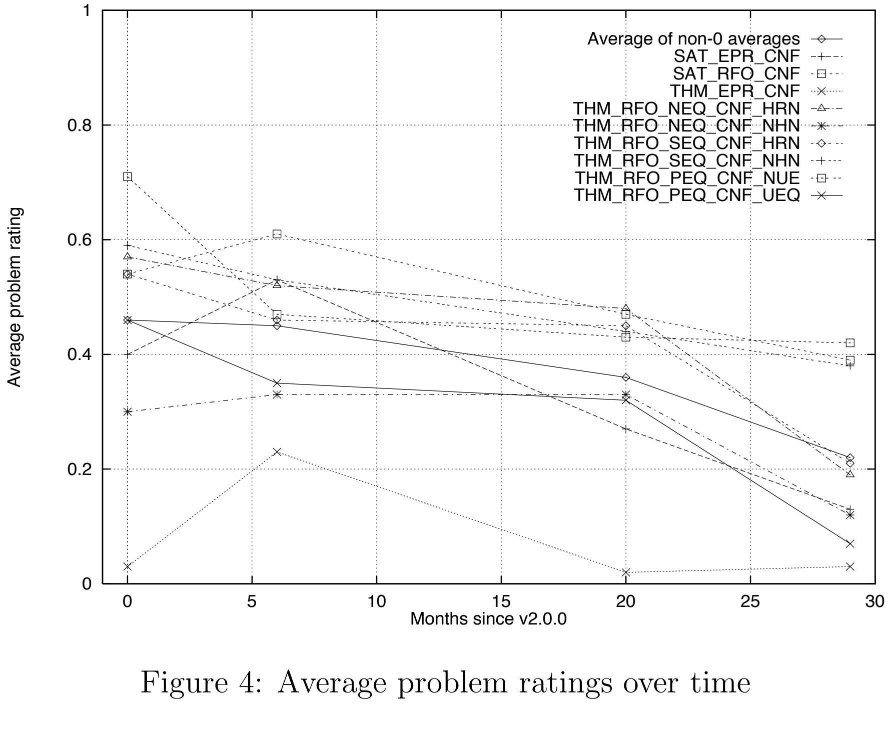


Figure 4: Average problem ratings over time

have entered the previous CASC's division winners into their divisions. The previous winners provide fixed points against which the new systems can be judged, using the same resource limits.

Table 4 shows the performance of the previous CASC's winners in CASC-16 and CASC-17. In both CASCs, in all divisions, the previous winner is beaten by one or more of the new systems. These results indicate that there is progress in ATP between each CASC. It may be claimed that the different eligible and randomly selected problems of each CASC is the cause of a new system beating the previous winner, but the consistency with which previous winners are outperformed makes this claim incredibly unlikely. Conversely, the addition of new problems to the TPTP each year, and the different selection of problems used in the competition, means that the improvements in the new systems cannot be attributed simply to tuning for the previous CASC's problems.

Division	Division winner		
	Problems/Solved by winner/By previous winner (Position)		
	CASC-15	CASC-16	CASC-17
MIX	Gandalf c-1.1 80/61/-	Vampire 0.0 75/51/39 (4th)	E 0.6 75/57/37 (5th)
UEQ	Waldmeister 798 30/30/-	Waldmeister 799 30/30/19 (2nd)	Waldmeister 600 30/30/29 (2nd)
SAT	SPASS 0.95T 30/22/-	OtterMACE 437 30/16/9 (3rd)	GandalfSat 1.0 30/25/21 (4th)
FOF	SPASS 0.95T 40/39/-	SPASS 1.00T 30/22/19 (3rd)	VampireFOF 1.0 60/53/51 (2nd)

Table 4: Performance of previous CASC division winners

4 Conclusion

This paper presents quantitative measures that show progress in ATP, from mid-1997 to the end of 1999. The measures are based on collected performance data from ATP systems, and from the results of the CADE ATP System Competitions. The performance data comes from testing ATP systems on TPTP problems, which are divided into 14 Specialist Problem Classes. The performance data has been analysed in four different ways, and for six of the SPCs the analyses consistently indicate serious progress in ATP. The six SPCs are the THM_RFO_*_CNF_* SPCs that represent the "mainstream" of ATP and ATP applications. There is some evidence, but a lesser amount, of progress in the remaining eight SPCs. The comparisons of ATP systems based on the CASC results similarly provide convincing evidence of progress in ATP, especially in the "mainstream" divisions.

The conclusion that ATP is making progress sends out messages to users, researchers, and observers. Take heed ...

- To users: ATP research is steadily producing more powerful systems that can solve your problems.
- To researchers: The long hard effort is paying off.
- To funding bodies: Your money is being well spent, as support for ATP research is producing real results.

References

Bachmair, L., Ganzinger, H., Lynch, C., and Snyder, W. (1992). Basic Paramodulation and Superposition. In D., Kapur, editor, *Proceedings of the 11th International Conference on Automated Deduction*, number 607 in Lecture Notes in Artificial Intelligence, pages 462–476. Springer-Verlag.

Bratko, I. (1990). *Prolog Programming for Artificial Intelligence, 2nd Edition.* Addison-Wesley.

Dershowitz, N. and Vigneron, L. (2000). Rewriting Home Page. http://rewriting.loria.fr/.

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

Fuchs, M. and Sutcliffe, G. (2000). Homogeneous Sets of ATP Problems. Technical Report TR-ARP-09-00, Automated Reasoning Project, Australian National University, Canberra, Australia.

Fujita, M., Slaney, J.K., and Bennett, F. (1993). Automatic Generation of Some Results in Finite Algebra. In Bajcsy, R., editor, *Proceedings of the 13th International Joint Conference on Artificial Intelligence*, pages 52–57. Morgan Kaufmann.

Furbach, U., Beckert, B., Hähnle, R., Letz, R., Baumgartner, P., Egly, U., Bible, W., Brüning, S., Otten, J., Rath, T., and Schaub, T. (1998). Tableau and Connection Calculi. In Bibel, W. and Schmitt, P.H., editors, *Automated Deduction - A Basis for Applications, Volume 1: Foundations - Calculi and Methods*, pages 3–179. Kluwer.

Kaufmann, M. (1998). ACL2 Support for Verification Projects. In Kirchner, C. and Kirchner, H., editors, *Proceedings of the 15th International Conference on Automated Deduction*, number 1421 in Lecture Notes in Artificial Intelligence, pages 220–238. Springer-Verlag.

Knuth, D.E. and Bendix, P.B. (1970). Simple Word Problems in Universal Algebras. In J., Leech, editor, *Computational Problems in Abstract Algebras*, pages 263–297. Pergamon Press.

Kunen, K. (1996). Quasigroups, Loops, and Associative Laws. *Journal of Algebra*, 185:194–204.

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

Loveland, D.W. (1969). A Simplified Format for the Model Elimination Theorem-Proving Procedure. *Journal of the ACM*, 16(3):349–363.

Loveland, D.W. (1999). Automated Deduction - Looking Ahead. *AI Magazine*, 20(1):77–98.

Lusk, E.L. (1992). Controlling Redundancy in Large Search Spaces: Argonne-Style Theorem Proving Through the Years. In A., Voronkov, editor, *Proceedings of the 3rd International Conference on Logic Programming and Automated Reasoning* , number 624 in Lecture Notes in Artificial Intelligence. Springer-Verlag.

McCune, W.W. (1997). Solution of the Robbins Problem. *Journal of Automated Reasoning*, 19(3):263–276.

McCune, W.W. (2000). EQP: Equational Prover. http://www-unix.mcs.anl.gov/AR/eqp/.

McCune, W.W. and Padmanabhan, R. (1996). *Automated Deduction in Equational Logic and Cubic Curves*, volume 1095 of *Lecture Notes in Artificial Intelligence.* Springer-Verlag.

Peter, L.J. and Hull, R. (1969). *The Peter Principle.* Souvenir Press.

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

Ramakrishnan, I.V., Sekar, R., and Voronkov, A. (1999). Term Indexing. In Robinson, A. and Voronkov, A., editors, *Handbook of Automated Reasoning.* Elsevier Science.

Reif, W. (1995). The KIV-approach to Software Verification. In Broy, M. and Jähnichen, S., editors, *KORSO: Methods, Languages, and Tools for the Construction of Correct Software - Final Report*, number 1009 in Lecture Notes in Computer Science.

Robinson, G.A. and Wos, L. (1969). Paramodulation and Theorem Proving in First-Order Theories with Equality. *Machine Intelligence*, 4:135–150.

Robinson, J.A. (1965). A Machine-Oriented Logic Based on the Resolution Principle. *Journal of the ACM*, 12(1):23–41.

Slaney, J.K. (1994). CADE-12 invited talk: The Crisis in Finite Mathematics: Automated Reasoning as Cause and Cure.

Stickel, M.E. (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. (To appear 2001). The CADE-17 ATP System Competition. *Journal of Automated Reasoning*.

Sutcliffe, G. and Seyfang, D. (1999). Smart Selective Competition Parallelism ATP. In A., Kumar and I., Russell, editors, *Proceedings of the 12th Florida Artificial Intelligence Research Symposium*, pages 341–345. AAAI Press.

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

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

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

Sutcliffe, G. and Suttner, C.B. (2000a). ATP System Results for the TPTP Problem Library. http://www.cs.jcu.edu.au/~tptp/TPTP/Results.html.

Sutcliffe, G. and Suttner, C.B. (2000b). Evaluating General Purpose Automated Theorem Proving Systems. Technical Report 2000/2, School of Information Technology, James Cook University, Townsville, Australia.

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

Weidenbach, C. (1999). Towards and Automatic Analysis of Security Protocols in First Order Logic. In Ganzinger, H., editor, *Proceedings of the 16th International Conference on Automated Deduction*, number 1632 in Lecture Notes in Artificial Intelligence, pages 314–328. Springer-Verlag.

Wos, L., Overbeek, R., and Lusk, E. (1993). Subsumption, a Sometimes Undervalued Procedure. Technical Report MCS-P93-0789, Mathematics and Computer Science Division, Argonne National Laboratory, Argonne, USA.

Wos, L., Robinson, G.A., Carson, D.F., and Shalla, L. (1967). The Concept of Demodulation in Theorem Proving. *Journal of the ACM*, 14(4):698–709.