Um Provador de Teoremas Multi-Estratégia
A Multi-Strategy Theorem Prover

Adolfo Gustavo Serra Seca Neto
DAINF - UTFPR
`http://www.dainf.ct.utfpr.edu.br/~adolfo`

Data desta versão: 14 de novembro de 2008
Date of this version: November 14th, 2008

Sobre

Este é um capítulo da minha tese de Doutorado intitulada "Um Provador de Teoremas Multi-Estratégia". Esta tese, na área de Ciência da Computação, foi defendida em 30 de janeiro de 2007 no Instituto de Matemática e Estatística (IME) da Universidade de São Paulo (USP). Meu orientador foi o Prof. Dr. Marcelo Finger. O texto completo desta tese está disponível em
`http://www.teses.usp.br/teses/disponiveis/45/45134/tde-04052007-175943/`

About

This is a chapter of my Ph.D. thesis entitled "A Multi-Strategy Theorem Prover". This Computer Science thesis was defended on January 30th, 2007 at the Institute of Mathematics and Statistics (IME) of the University of São Paulo (USP). My advisor was Prof. Dr. Marcelo Finger. Thesis full text is available at
`http://www.teses.usp.br/teses/disponiveis/45/45134/tde-04052007-175943/`. Only the first chapter was written in Portuguese. All the following appendices were written in English.

Apêndice E

Conclusion

E.1 Test Conclusions

Let us discuss the results presented in Section D.2. First let us state that no **KEMS** prover configuration obtained incorrect results with the evaluation problem instances. Some of these instances, such as the ones from PHP and ST families, were very difficult to prove. That is, the proof search procedure for some very small instances did not finish for any strategy-sorter pair. Some other families were difficult only for some strategy-sorter pairs.

For **CPL** tests, a pair with Memory Saver Strategy (MSS) as the strategy achieved the best time results for all problem families except for Γ and B_PHP families (see Table E.1). The best pair for Γ had Learning Strategy (LS) as the strategy, but several other pairs had equally good results. And the best pair for B_PHP used, as expected, Backjumping Simple Strategy (BSS). Therefore, in general our first option when choosing a strategy should be Memory Saver Strategy.

The results for sorters were not as conclusive. We can see in Table E.1 that almost all sorters were present in a best pair for some family. Thus, the choice of the best sorter to run a problem with will depend on the problem being tackled. If we already know the the problem belongs to a given family, we can choose a sorter that worked well for that family. Otherwise, we cannot suggest any sorter.

Now let us analyse **mbC** and **mCi** tests. Both **mbC** strategies, **mbC** Simple Strategy

(MBCSS) and **mbC** Extended Strategy (MBCES), achieved equivalent results with valid problems (see Table E.2)[1]. MBCSS was slightly better with non-valid problem families. And for the other implemented **LFI**, **mCi** Simple Strategy (MCISS) and **mCi** Extended Strategy (MCIES) also achieved comparable results (see Table E.3). Therefore, if we are going to run a problem that does not belong to one of the families we have used for evaluating **KEMS**, we cannot indicate a specific strategy to be tried first. Nor any sorter.

In all **LFI** tests the sorter in the best pairs varied according to the problem family. And it was interesting to notice that for some families the sorter choice was almost as important as the strategy choice. For instance, with second family instances only one sorter was able to prove the biggest problem solved, independently of the strategy.

These results we obtained by **KEMS** with the **LFI** families are the first benchmark results for these families. All results obtained with the three logical systems can be compared with other provers for these logics and are available at [92].

Bigger instance solved	Problem size	Best time pair	Best size pair
Γ_{360}	3606	`<LS,inc>`	`<LS,inc>`
H_6	959	`<MSS,F>`	`<MSS,F>`
Statman_{29}	3338	`<MSS,F>`	`<MSS,F>`
PHP_7	692	`<MSS,or>`	`<MSS,or>`
U_{13}	51	`<MSS,bi-impli>`	`<MSS,bi-impli>`
ST_4	80	`<MSS,rev>`	`<CS,dec>`
B_PHP_6^3	465	`<BSS,or>`	`<BSS,or>`
cnf_2	526	`<MSS,nfo>`	`<MSS,nfo>`
Φ_{130}^1	1173	`<MSS,F>`	`<MSS,rfo>`
Φ_{20}^2	623	`<MSS,T>`	`<CS,rfo>`
Φ_{20}^3	672	`<MSS,F>`	`<CS,or>`
Φ_{100}^4	1000	`<MSS,F>`	`<MSS,F>`
Φ_{20}^7	296	`<MSS,xor>`	`<MSS,xor>`
Φ_{10}^8	166	`<MSS,and>`	`<MSS,and>`
Φ_{25}^9	272	`<MSS,T>`	`<MSS,T>`

Table E.1: Best **CPL** strategy-sorter pairs.

---

[1] Actually this is not clear in the table, where we show only one of the best when more than one pair tied, but can be seen when we examine the evaluation files produced by **KEMS**.

Bigger instance solved	Problem size	Best time pair	Best size pair
Φ^1_{90}	993	`<MBCES,dec>`	`<MBCSS,inc>`
Φ^2_{14}	472	`<MBCSS,rev>`	`<MBCSS,rev>`
Φ^3_{14}	509	`<MBCSS,rev>`	`<MBCSS,rev>`
Φ^4_{90}	1079	`<MBCES,rfo>`	`<MBCSS,rfo>`
Φ^7_{20}	336	`<MBCES,F>`	`<MBCSS,nfo>`

Table E.2: Best **mbC** strategy-sorter pairs.

Bigger instance solved	Problem size	Best time pair	Best size pair
Φ^1_{90}	993	`<MCIES,dec>`	`<MCISS,inc>`
Φ^2_{17}	649	`<MCISS,rev>`	`<MCISS,rev>`
Φ^3_{12}	402	`<MCISS,rev>`	`<MCISS,rev>`
Φ^4_{90}	1079	`<MCISS,rfo>`	`<MCISS,rfo>`
Φ^7_{20}	336	`<MCIES,and>`	`<MCISS,nfo>`
Φ^8_{50}	946	`<MCISS,or>`	`<MCISS,or>`
Φ^9_{75}	1197	`<MCIES,inc>`	`<MCIES,inc>`

Table E.3: Best **mCi** strategy-sorter pairs.

E.2 Thesis Conclusions and Contributions

We succeeded in developing a multi-strategy theorem prover where we can vary the strategy without modifying the core of the implementation. **KEMS** allows us to describe several proof strategies for the same logical system, and to implement different logical systems.

We list below some of the contributions of this work:

- an analytic, correct and complete **KE** system for **mbC**;
- a correct and complete **KE** system for **mCi**;
- a multi-strategy prover with the following characteristics:
  - accepts problems in three logical systems: **CPL**, **mbC** and **mCi**[2];
  - has 6 implemented strategies for **CPL**, 2 for **mbC** and 2 for **mCi**;
  - has 13 sorters to be used alongside with the strategies;
  - implements simplification rules of **CPL**;

---

[2] We know of no other prover for **mbC** and **mCi**.

  - provides a proof viewer with a graphical user interface;
  - it is open source and available at [92].

- benchmark results obtained by **KEMS** comparing its **CPL** strategies with several problem families;
- seven problem families designed to evaluate **LFI** provers;
- the first benchmark results for **LFI** families obtained with several **KEMS mbC**, **mCi** and **CPL** strategies.

E.3 Future Works

On the logical side, it would be useful to have a general procedure for automatically generating correct and complete **KE** systems for **LFI**s and other logical systems, similar to the procedure for generating tableau systems presented in [11]. This could help us to extend **KEMS** to other logical systems. Marcelo Coniglio (one of the authors of [11]) informed us that they have already thought of adapting their method that obtains what we called here $\mathbf{C^3M}$ systems (which for **LFI**s are a kind of mixture of **AT** and **KE**) to be able to produce **KE** systems.

On the implementation side, from the results we have obtained[3], we can see that it would be very useful to have an adaptive strategy that changes its behavior according to features of the problem presented to it. This strategy can behave as other implemented strategies and it will be able to vary its actions not only for different problems but also for different subproblems of the same problem given as input.

Next, we plan to implement new strategies making heavy use of Aspect Orientation. The objective is to easily mix strategy features to produce new strategies. For instance, we have a Backjumping Simple Strategy for **CPL** but not for **LFI**s. Without using aspect orientation, in **KEMS** we would have to create a new class for implementing a Backjumping Simple Strategy for **LFI**s. If we use aspects we could have strategies for

---

[3] That is, from the conclusion that there is no strategy-sorter pair which is the best for every problem family.

CPL, **mbC**, and **mCi** and we would be able to obtain Backjumping versions of them by implementing a Backjumping Aspect that changes the behavior of these strategies.

An obvious **KEMS** extension is to develop strategies for $\mathbf{C}_1$, the simplest in da Costa's C_n hierarchy of paraconsistent logics [27]. To achieve this, we first have to provide a **KE** system for $\mathbf{C}_1$. We have already developed the rules of this system and soon we will publish them and prove that this $\mathbf{C}_1$ **KE** system is correct and complete. Another possible future work is to develop strategies for the other **LFI**s presented in [18]. Several **LFI**s are built there by adding/removing axioms to other **LFI**s. It does not seem to be difficult to implement a module in **KEMS** where we could build a particular **LFI** by choosing some features and then adapt some general predefined **LFI** strategies for this specific system. To evaluate strategies for these **LFI**s we would have to design new problem families which are valid in these logics.

Another extension would be to implement some restricted simplification rules for **LFI**s to obtain more efficient strategies. **KEMS** could also be improved by developing strategies for other propositional logics that have **KE** systems. It would be interesting to extend **KEMS** to first-order logics and then use some of the ideas described in [72] to vary strategies.

Logical systems for approximate reasoning are presented in [48, 49, 50], and in [47] the relationship between some of these systems and paraconsistency is discussed. It would be interesting to try to implement these logical systems in **KEMS** to evaluate how easy is to adapt **KEMS** to other logics. We know that it is not so easy to implement different logical systems in **KEMS**, therefore in the future we plan to document how this can be done.

Finally, we plan to implement in **KEMS** a probabilistic strategy such as GSAT [105]. This strategy, given a valid problem, would not always find a **KE** proof. However, it would be able in most cases to find fast valuations for satisfiable problems. Yet another idea is to allow the dynamic building of strategies either from some predefined options or from some strategy-definition language, in a way which is similar to what is done in [37].

Referências Bibliográficas

[1] Michael Alekhnovich, Jan Johannsen, Toniann Pitassi, and Alasdair Urquhart. An exponential separation between regular and general resolution. In *STOC '02: Proceedings of the thirty-fourth annual ACM symposium on Theory of computing*, pages 448–456, New York, NY, USA, 2002. ACM Press.

[2] Noriko Arai, Toniann Pitassi, and Alasdair Urquhart. The complexity of analytic tableaux. In *STOC '01: Proceedings of the thirty-third annual ACM symposium on Theory of computing*, pages 356–363. ACM Press, 2001.

[3] Bernhard Beckert, Richard Bubel, Elmar Habermalz, and Andreas Roth. jTAP - a Tableau Prover in Java, February 1999. Universitat Karlsruhe. Available at `http://i12www.ira.uka.de/~aroth/jTAP/`. Last accessed, November 2006.

[4] Bernhard Beckert and Joachim Posegga. leanTAP: Lean tableau-based deduction. *Journal of Automated Reasoning*, 15(3):339–358, 1995.

[5] Evert W. Beth. *The Foundations of Mathematics.* North-Holland Publishing Company, Amsterdam, 1959.

[6] Maria Paola Bonacina and Thierry Boy de la Tour. Fifth Workshop on Strategies in Automated Deduction - Workshop Programme, 2004. `http://tinyurl.com/y8dkbj`. Last accessed, November 2006.

[7] Maria Luisa Bonet and Nicola Galesi. A study of proof search algorithms for resolution and polynomial calculus. In *FOCS '99: Proceedings of the 40th Annual*

Symposium on Foundations of Computer Science, page 422, Washington, DC, USA, 1999. IEEE Computer Society.

[8] Krysia Broda, Marcello D'Agostino, and Marco Mondadori. A Solution to a Problem of Popper. In *The Epistemology of Karl Popper*. Kluwer, 1995. `http://citeseer.nj.nec.com/broda95solution.html`. Last accessed, November 2006.

[9] S. R. Buss. Polynomial size proofs of the propositional pigeonhole principle. *Journal of Symbolic Logic*, 52:916–927, 1987.

[10] Liming Cai. *Nondeterminism and optimization.* PhD thesis, Texas A&M University, USA, 1994.

[11] Carlos Caleiro, Walter Carnielli, Marcelo E. Coniglio, and Joao Marcos. Two's company: "The humbug of many logical values". In *Logica Universalis*, pages 169–189. Birkhäuser Verlag, Basel, Switzerland, 2005. Pre-print available at `http://tinyurl.com/yb5qbz`. Last accessed, November 2006.

[12] Alessandra Carbone and Stephen Semmes. *Graphic Apology for Symmetry and Implicitness.* Oxford University Press, 2000.

[13] W. A. Carnielli. Systematization of the finite many-valued logics through the method of tableaux. *The Journal of Symbolic Logic*, 52:473–493, 1987.

[14] W.A. Carnielli and J. Marcos. Ex Contradictione Non Sequitur Quodlibet. *Bulletin of Advanced Reasoning and Knowledge*, 1:89–109, 2001.

[15] W.A. Carnielli and J. Marcos. Tableau systems for logics of formal inconsistency. *Proceedings of the International Conference on Artificial Intelligence (IC-AI'2001)*, pages 848–852, 2001.

[16] Walter Carnielli. How to build your own paraconsistent logic: an introduction to the Logics of Formal (In)Consistency. *Proceedings of the Workshop on Paraconsistent Logic (WoPaLo)*, 2002.

[17] Walter Carnielli, Marcelo Coniglio, and Ricardo Bianconi. *Logic and Applications: Mathematics, Computer Science and Philosophy (in Portuguese).* Unpublished, 2005. Preliminary version. Chapters 1 to 5.

[18] Walter Carnielli, Marcelo E. Coniglio, and Joao Marcos. Logics of Formal Inconsistency. In *Handbook of Philosophical Logic*, volume 12. Kluwer Academic Publishers, 2005. To appear. Pre-print available at `http://tinyurl.com/ybn4yw`. Last accessed, November 2006.

[19] Walter A. Carnielli and Richard L. Epstein. *Computabilidade – Funções Computáveis, Lógica e os Fundamentos da Matemática.* Editora da Unesp, 2006.

[20] Stephen Cook. The P versus NP problem, 2000. `http://tinyurl.com/n5thm`. Last accessed, May 2005.

[21] Stephen Cook. The importance of the P versus NP question. *J. ACM*, 50(1):27–29, 2003.

[22] Stephen Cook and Robert Reckhow. On the lengths of proofs in the propositional calculus (preliminary version). In *STOC '74: Proceedings of the sixth annual ACM symposium on Theory of computing*, pages 135–148, New York, NY, USA, 1974. ACM Press.

[23] Stephen A. Cook. The complexity of theorem-proving procedures. In *STOC '71: Proceedings of the third annual ACM symposium on Theory of computing*, pages 151–158, New York, NY, USA, 1971. ACM Press.

[24] Stephen A. Cook. A short proof of the pigeon hole principle using extended resolution. *SIGACT News*, 8(4):28–32, 1976.

[25] W. Cook, C. R. Coullard, and G. Turán. On the complexity of cutting-plane proofs. *Discrete Appl. Math.*, 18(1):25–38, 1987.

[26] Thomas H. Cormen, Charles E. Leiserson, Ronald L. Rivest, and Clifford Stein. *Introduction to Algorithms - Second Edition.* MIT Press, 2001.

[27] Newton C. A. da Costa, Décio Krause, and Otávio Bueno. Paraconsistent logics and paraconsistency: Technical and philosophical developments. *CLE e-prints (Section Logic)*, 4(3), 2004. Pre-print available at `http://tinyurl.com/yxhon7`. Last accessed, November 2006.

[28] Marcello D'Agostino. Are Tableaux an Improvement on Truth-Tables? Cut-Free proofs and Bivalence, 1992. Available at `http://citeseer.nj.nec.com/140346.html`. Last accessed, May 2005.

[29] Marcello D'Agostino. Tableau methods for classical propositional logic. In Marcello D'Agostino et al., editor, *Handbook of Tableau Methods*, chapter 1, pages 45–123. Kluwer Academic Press, 1999.

[30] Marcello D'Agostino, Dov Gabbay, and Krysia Broda. Tableau methods for substructural logics. In Marcello D'Agostino et al., editor, *Handbook of Tableau Methods*, chapter 6, pages 397–467. Kluwer Academic Press, 1999.

[31] Marcello D'Agostino and Marco Mondadori. The taming of the cut: Classical refutations with analytic cut. *Journal of Logic and Computation*, pages 285–319, 1994.

[32] Martin Davis, George Logemann, and Donald Loveland. A machine program for theorem-proving. *Commun. ACM*, 5(7):394–397, 1962.

[33] Martin Davis and Hilary Putnam. A computing procedure for quantification theory. *J. ACM*, 7(3):201–215, 1960.

[34] Sandra de Amo, Walter Carnielli, and João Marcos. A Logical Framework for Integrating Inconsistent Information in Multiple Databases. In Thomas Eiter and Klaus-Dieter Schewe, editors, *Lecture Notes in Computer Science*, volume 2284, pages 67–84. Springer-Verlag, Berlim., 2002.

[35] Eleonora de Moraes. *Lista de discussão gestar bem interior-sp*, 2004. `http://br.groups.yahoo.com/group/gestarbeminterior-sp`. Last accessed, December 2006.

[36] Rina Dechter. *Constraint Processing*. Morgan Kaufmann, 2003.

[37] Luis Fariñas del Cerro, David Fauthoux, Olivier Gasquet, Andreas Herzig, Dominique Longin, and Fabio Massacci. Lotrec: The generic tableau prover for modal and description logics. In *IJCAR '01: Proceedings of the First International Joint Conference on Automated Reasoning*, pages 453–458. Springer-Verlag, 2001.

[38] Wagner Dias. Tableaux implementation for approximate reasoning (in portuguese). Master's thesis, Computer Science Department, Institute of Mathematics and Statistics, University of São Paulo, 2002.

[39] Simone Grilo Diniz and Ana Cristina Duarte. *Parto normal ou cesárea? O que toda mulher deve saber (e todo homem também)*. Editora da Unesp, São Paulo, 2004.

[40] Heidi Dixon. *Automating Pseudo-Boolean Inference Within a DPLL Framework*. PhD thesis, University of Oregon, USA, Dec 2004. Available at `http://www.cirl.uoregon.edu/dixon/dixonDissertation.pdf`. Last accessed, August 2005.

[41] Amigas do Parto. *Lista de discussão Parto Nosso*, 2003. `br.groups.yahoo.com/group/partonosso`. Last accessed, December 2006.

[42] Itala M. Loffredo D'Ottaviano and Milton Augustinis de Castro. Analytical Tableaux for da Costa's Hierarchy of Paraconsistent Logics $C_n, 1 \leq n \leq \omega$. *Journal of Applied Non-Classical Logics*, 15(1):69–103, 2005.

[43] Tzilla Elrad, Robert E. Filman, and Atef Bader. Aspect-Oriented Programming. *Communications of the ACM*, 44, 2001.

[44] M. Enkin, M. Skiers, J. Nelson, C. Crowder, L. Duly, and E. Hodnett. *A guide to effective care during pregnancy and childbirth*. Oxford, UK: Oxford University Press, 2000.

[45] Fadynha. *Lista de discussão partonatural*, 1999. `br.groups.yahoo.com/group/partonatural`. Last accessed, December 2006.

[46] Luis Fariñas del Cerro, David Fauthoux, Olivier Gasquet, Andreas Herzig, Dominique Longin, and Fabio Massacci. Lotrec: the generic tableau prover for modal and description logics. In *International Joint Conference on Automated Reasoning*, LNCS, page 6. Springer Verlag, 18-23 juin 2001.

[47] M. Finger and R. Wassermann. Approximate Reasoning and Paraconsistency- Preliminary Report. *Proceedings of the Eigth Workshop on Logic, Language, Information and Comunication (WoLLIC), Braslia, Brazil*, 2001.

[48] M. Finger and R. Wassermann. The universe of approximations. *Electronic Notes in Theoretical Computer Science*, 84:1–14, 2003.

[49] M. Finger and R. Wassermann. Anytime Approximations of Classical Logic from Above. *Journal of Logic and Computation*, 2006.

[50] M. Finger and R. Wassermann. The universe of propositional approximations. *Theoretical Computer Science*, 355(2):153–166, 2006.

[51] Melvin Fitting. *First-order logic and automated theorem proving (2nd ed.).* Springer-Verlag New York, Inc., 1996.

[52] Melvin Fitting. Introduction. In Marcello D'Agostino et al., editor, *Handbook of Tableau Methods*, chapter 1, pages 1–43. Kluwer Academic Press, 1999.

[53] Zhaohui Fu, Yogesh Mahajan, and Sharad Malik. New Features of the SAT'04 versions of zChaff, 2004. `http://www.princeton.edu/~chaff/zchaff/sat04.pdf`. Last accessed, September 2005.

[54] Dov M. Gabbay. *Labelled Deductive Systems, Volume 1.* Oxford University Press, Oxford, 1996.

[55] Erich Gamma, Richard Helm, Ralph Johnson, and John Vlissides. *Design Patterns: Elements of Reusable Object-Oriented Software.* Adisson-Wesley, 1994.

[56] Gerhard Gentzen. Investigations into logical deductions, 1935. In M. E. Szabo, editor, *The Collected Works of Gerhard Gentzen*, pages 68–131. North-Holland, Amsterdam, 1969.

[57] J. Gosling, B. Joy, and G. Steele. *The Java Programming Language.* Addison-Wesley, Reading, MA, 1996.

[58] Armin Haken. The intractability of resolution. *Theor. Comput. Sci.*, 39:297–308, 1985.

[59] Klaus Havelund and Natarajan Shankar. Experiments in theorem proving and model checking for protocol verification. In *FME '96: Proceedings of the Third International Symposium of Formal Methods Europe on Industrial Benefit and Advances in Formal Methods*, pages 662–681. Springer-Verlag, 1996.

[60] J. Hintikka. Form and content in quantification theory. *Acta Philosophica Fennica*, 8:7–55, 1955.

[61] Jan Holub and Borivoj Melichar. Implementation of nondeterministic finite automata for approximate pattern matching. In *WIA '98: Revised Papers from the Third International Workshop on Automata Implementation*, pages 92–99. Springer-Verlag, 1999.

[62] Scott E. Hudson, Frank Flannery, C. Scott Anaian, Dan Wang, and Andrew Appel. *CUP Parser Generator for Java*, 1999. `http://www2.cs.tum.edu/projects/cup`. Last accessed, November 2006.

[63] Ullrich Hustadt and Renate A. Schmidt. Simplification and backjumping in modal tableau. In *Lecture Notes in Computer Science*, volume 1397 of *Lecture Notes in Computer Science*, pages 187–201, 1998.

[64] Gregor Kiczales, Erik Hilsdale, Jim Hugunin, Mik Kersten, Jeffrey Palm, and William G. Griswold. Getting Started with AspectJ. *Communications of the ACM*, 44:59–65, 2001.

[65] Gregor Kiczales, Erik Hilsdale, Jim Hugunin, Mik Kersten, Jeffrey Palm, and William G. Griswold. An overview of AspectJ. *Lecture Notes in Computer Science*, 2072:327–355, 2001.

[66] Gerwin Klein. *JFlex: the fast lexical analyzer generator for Java*, 1998. `http://jflex.de`. Last accessed, November 2006.

[67] S. Kundu and J. Chen. Fuzzy logic or Lukasiewicz logic: A clarification. *Fuzzy Sets and Systems*, 95(3):369–379, 1998.

[68] L. Di Lascio. Analytic fuzzy tableaux. *Soft Computing - A Fusion of Foundations, Methodologies and Applications*, 5(6):434–439, Dec 2001.

[69] D. W. Loveland. Automated deduction: Some achievements and future directions. Technical report, National Science Foundation, 1997. Available at `http://tinyurl.com/y7mb6p`. Last accessed, May 2005.

[70] D. W. Loveland. Automated deduction: achievements and future directions. *Commun. ACM*, 43(11es):10, 2000.

[71] Wendy MacCaull. Tableau method for residuated logic. *Fuzzy Sets Syst.*, 80(3):327–337, 1996.

[72] Alistair Manning, Andrew Ireland, and Alan Bundy. Increasing the Versatility of Heuristic Based Theorem Provers. In *LPAR'93*, 1993.

[73] Heiko Mantel and Jens Otten. lintap: A tableau prover for linear logic. In *TABLEAUX '99: Proceedings of the International Conference on Automated Reasoning with Analytic Tableaux and Related Methods*, pages 217–231, London, UK, 1999. Springer-Verlag.

[74] João Marcos. Personal communication by email, October 2006.

[75] Fabio Massacci. Simplification: A general constraint propagation technique for propositional and modal tableaux. In *TABLEAUX '98: Proceedings of the Inter-*

national Conference on Automated Reasoning with Analytic Tableaux and Related Methods, pages 217–231. Springer-Verlag, 1998.

[76] Fabio Massacci. The proof complexity of analytic and clausal tableaux. *Theor. Comput. Sci.*, 243(1-2):477–487, 2000.

[77] William McCune and Larry Wos. Otter - The CADE-13 Competition Incarnations. *J. Autom. Reason.*, 18(2):211–220, 1997.

[78] Elliott Mendelson. *Introduction to Mathematical Logic*. Chapman & Hall, London, UK, fourth edition, 1997.

[79] Paulo Blauth Menezes. *Linguagens Formais e Autômatos*. Instituto de Informática da UFRGS : Editora Sagra Luzzatto, Porto Alegre, 232p., 2005.

[80] Sun Microsystems. Java Runtime Environment (JRE) 5.0 Installation Notes, 2006. `http://java.sun.com/j2se/1.5.0/jre/install.html`. Last accessed, November 2006.

[81] S. H. Mirian and M. Mousavi. Nondeterminism in set-theoretic specifications (in persian). In *Proceedings of Iranian Computer Society Annual Conference (CSICC'02)*, feb 2002.

[82] David G. Mitchell, Bart Selman, and Hector J. Levesque. Hard and easy distributions for SAT problems. In Paul Rosenbloom and Peter Szolovits, editors, *Proceedings of the Tenth National Conference on Artificial Intelligence*, pages 459–465, Menlo Park, California, 1992. AAAI Press.

[83] M. Moskewicz, C. Madigan, Y. Zhao, L. Zhang, and S. Malik. Chaff: Engineering an Efficient SAT Solver. In *Proceedings of the 38th Design Automation Conference (DAC'01)*, June 2001.

[84] John K. Myers. An introduction to planning and meta-decision-making with uncertain nondeterministic action using 2nd-order probabilities. In *Proceedings of the first*

international conference on Artificial intelligence planning systems, pages 297–298. Morgan Kaufmann Publishers Inc., 1992.

[85] Adolfo Neto. An Object-Oriented Implementation of a KE Tableau Prover, nov 2003. Avaliable at `http://tinyurl.com/y3qmkx`. Last accessed, November 2006.

[86] Adolfo Neto. Modifications on the implementation of a framework for tableau methods, jul 2003. Avaliable at `http://tinyurl.com/yjgjnq`. Last accessed, November 2006.

[87] Adolfo Neto and Marcelo Finger. A Multi-Strategy Tableau Prover. In *I Simpósio de Iniciação Científica e Pós-Graduação do IME-USP*. University of São Paulo, 2005. Available at `http://tinyurl.com/tbdd6`. Last accessed, November 2006.

[88] Adolfo Neto and Marcelo Finger. A Multi-Strategy Tableau Prover. In *SeMe-2005. Workshop "Semantics and Meaning"*, IFIP International Federation for Information Processing. Unicamp. Campinas-SP., 2005. Available at `http://tinyurl.com/yzx8ve`. Last accessed, November 2006.

[89] Adolfo Neto and Marcelo Finger. Implementing a multi-strategy theorem prover. In Ana Cristina Bicharra Garcia and Fernando Santos Osório, editors, *Proceedings of the V ENIA (Encontro Nacional de Inteligência Artificial), held in São Leopoldo-RS, Brazil, July 22-29 2005*, 2005. Available at `http://tinyurl.com/yd6n6n`. Last accessed, November 2006.

[90] Adolfo Neto and Marcelo Finger. Using Aspect-Oriented Programming in the Development of a Multi-Strategy Theorem Prover. In *Anais da II Jornada do Conhecimento e da Tecnologia do Univem*, Marília-SP, 2005. Available at `http://www.ime.usp.br/~adolfo/trabalhos/jornada2005.pdf`. Last accessed, November 2006.

[91] Adolfo Neto and Marcelo Finger. Effective Prover for Minimal Inconsistency Logic. In *Artificial Intelligence in Theory and Practice*, IFIP International Federation for Information Processing, pages 465–474. Springer Verlag, 2006. Available at `http:`

//www.springerlink.com/content/b80728w7m6885765. Last accessed, November 2006.

[92] Adolfo Neto and Marcelo Finger. *KEMS - A* **KE** *Multi-Strategy Tableau Prover*, 2006. http://kems.iv.fapesp.br. Last accessed, November 2006.

[93] Michel Odent. *The Caesarean.* Free Association Books, 2004.

[94] Lawrence C. Paulson. *Handbook of logic in computer science (vol. 2): background: computational structures*, chapter Designing a theorem prover, pages 415–475. Oxford University Press, Inc., 1992.

[95] Francis Jeffry Pelletier. Seventy-five problems for testing automatic theorem provers. *J. Autom. Reason.*, 2(2):191–216, 1986.

[96] J. V. Pitt and R. J. Cunningham. Theorem proving and model building with the calculus ke. *Journal of the IGPL*, 4(1):129–150, 1996.

[97] Awais Rashid and Lynne Blair. Editorial: Aspect-oriented Programming and Separation of Crosscutting Concerns. *The Computer Journal*, 46(5):527–528, 2003.

[98] Alexandre Riazanov and Andrei Voronkov. Vampire 1.1 (system description). In *IJCAR '01: Proceedings of the First International Joint Conference on Automated Reasoning*, pages 376–380. Springer-Verlag, 2001.

[99] M. Richardson and P. Domingos. Markov logic networks. *Machine Learning*, 62(1):107–136, 2006.

[100] J. A. Robinson. A machine-oriented logic based on the resolution principle. *J. ACM*, 12(1):23–41, 1965.

[101] Satisfiability suggested format, 1993. http://www.satlib.org. Last accessed, March 22, 2005.

[102] Satisfiability library, 2003. http://www.satlib.org. Last accessed, March 22, 2005.

[103] N. Scharli, S. Ducasse, O. Nierstrasz, and A.P. Black. Traits: Composable units of behaviour. *Proc. of ECOOP*, 2743:248–274, 2003.

[104] J. Schumann. Tableau-based theorem provers: Systems and implementations. *Journal of Automated Reasoning*, 13(3):409–421, 1994. `http://www.springerlink.com/content/k182u80451306371`. Last accessed, November 4th, 2006.

[105] Bart Selman, Hector J. Levesque, and D. Mitchell. A New Method for Solving Hard Satisfiability Problems. In Paul Rosenbloom and Peter Szolovits, editors, *Proceedings of the Tenth National Conference on Artificial Intelligence*, pages 440–446, Menlo Park, California, 1992. AAAI Press.

[106] Raymond M. Smullyan. *First-Order Logic.* Springer-Verlag, 1968.

[107] Sérgio Soares and Paulo Borba. AspectJ - Programação orientada a aspectos em Java. *Tutorial no SBLP 2002, 6o. Simpósio Brasileiro de Linguagens de Programação. 5 a 7 de Junho, PUC-Rio, Rio de Janeiro, Brasil*, pages 39–55, 2002.

[108] R. Statman. Bounds for proof-search and speed-up in the predicate calculus. *Annals of Mathematical Logic*, pages 225–287, 1978.

[109] Friedrich Steimann. The paradoxical success of aspect-oriented programming. *SIGPLAN Not.*, 41(10):481–497, 2006.

[110] Geoff Sutcliffe. An overview of automated theorem proving, 2001. `http://www.cs.miami.edu/~tptp/OverviewOfATP.html`. Last accessed, March 2005.

[111] Geoff Sutcliffe. Thousands of problems for theorem provers, 2001. `http://www.cs.miami.edu/~tptp`. Last accessed, March 2005.

[112] Geoff Sutcliffe and Christian Suttner. The CADE ATP System Competition, 2003. `http://www.cs.miami.edu/~tptp/CASC`. Last accessed, March 2005.

[113] Alasdair Urquhart. Hard examples for resolution. *J. ACM*, 34(1):209–219, 1987.

[114] Marian Vittek. A compiler for nondeterministic term rewriting systems. In *RTA '96: Proceedings of the 7th International Conference on Rewriting Techniques and Applications*, pages 154–167. Springer-Verlag, 1996.

[115] Wikipedia. *Nondetermnistic algorithm*, 2007. `http://en.wikipedia.org/wiki/Nondeterministic`. Last accessed, February 2007.