LNCS 4121

Armin Biere
Carla P. Gomes (Eds.)

Theory and Applications of Satisfiability Testing - SAT 2006

9th International Conference
Seattle, WA, USA, August 2006
Proceedings

Springer

Armin Biere Carla P. Gomes (Eds.)

Theory and Applications of Satisfiability Testing – SAT 2006

9th International Conference
Seattle, WA, USA, August 12-15, 2006
Proceedings

Springer

Volume Editors

Armin Biere
Johannes Kepler University Linz, Institute for Formal Models and Verification
Altenbergerstr. 69, 4040 Linz, Austria
E-mail: biere@jku.at

Carla P. Gomes
Cornell University, Department of Computer Science
5133 Upson Hall, Ithaca, NY 14853, USA
E-mail: gomes@cs.cornell.edu

Library of Congress Control Number: 2006930253

CR Subject Classification (1998): F.4.1, I.2.3, I.2.8, I.2, F.2.2, G.1.6

LNCS Sublibrary: SL 1 – Theoretical Computer Science and General Issues

ISSN 0302-9743
ISBN-10 3-540-37206-7 Springer Berlin Heidelberg New York
ISBN-13 978-3-540-37206-6 Springer Berlin Heidelberg New York

Springer is a part of Springer Science+Business Media

springer.com

Printed in Germany

Typesetting: Camera-ready by author, data conversion by Scientific Publishing Services, Chennai, India
Printed on acid-free paper SPIN: 11814948 06/3142 5 4 3 2 1 0

Lecture Notes in Computer Science 4121

Lecture Notes in Computer Science

For information about Vols. 1–4007

please contact your bookseller or Springer

Vol. 4127: E. Damiani, P. Liu (Eds.), Data and Applications Security XX. X, 319 pages. 2006.

Vol. 4121: A. Biere, C.P. Gomes (Eds.), Theory and Applications of Satisfiability Testing – SAT 2006. XII, 438 pages. 2006.

Vol. 4112: D.Z. Chen, D. T. Lee (Eds.), Computing and Combinatorics. XIV, 528 pages. 2006.

Vol. 4108: J.M. Borwein, W.M. Farmer (Eds.), Mathematical Knowledge Management. VIII, 295 pages. 2006. (Sublibrary LNAI).

Vol. 4106: T.R. Roth-Berghofer, M.H. Göker, H. A. Güvenir (Eds.), Advances in Case-Based Reasoning. XIV, 566 pages. 2006. (Sublibrary LNAI).

Vol. 4099: Q. Yang, G. Webb (Eds.), PRICAI 2006: Trends in Artificial Intelligence. XXVIII, 1263 pages. 2006. (Sublibrary LNAI).

Vol. 4098: F. Pfenning (Ed.), Term Rewriting and Applications. XIII, 415 pages. 2006.

Vol. 4097: X. Zhou, O. Sokolsky, L. Yan, E.-S. Jung, Z. Shao, Y. Mu, D.C. Lee, D. Kim, Y.-S. Jeong, C.-Z. Xu (Eds.), Emerging Directions in Embedded and Ubiquitous Computing. XXVII, 1034 pages. 2006.

Vol. 4096: E. Sha, S.-K. Han, C.-Z. Xu, M.H. Kim, L.T. Yang, B. Xiao (Eds.), Embedded and Ubiquitous Computing. XXIV, 1170 pages. 2006.

Vol. 4092: J. Lang, F. Lin, J. Wang (Eds.), Knowledge Science, Engineering and Management. XV, 664 pages. 2006. (Sublibrary LNAI).

Vol. 4090: S. Spaccapietra, K. Aberer, P. Cudré-Mauroux (Eds.), Journal on Data Semantics VI. XI, 211 pages. 2006.

Vol. 4088: Z.-Z. Shi, R. Sadananda (Eds.), Agent Computing and Multi-Agent Systems. XVII, 827 pages. 2006. (Sublibrary LNAI).

Vol. 4079: S. Etalle, M. Truszczyński (Eds.), Logic Programming. XIV, 474 pages. 2006.

Vol. 4077: M.-S. Kim, K. Shimada (Eds.), Advances in Geometric Modeling and Processing. XVI, 696 pages. 2006.

Vol. 4076: F. Hess, S. Pauli, M. Pohst (Eds.), Algorithmic Number Theory. X, 599 pages. 2006.

Vol. 4075: U. Leser, F. Naumann, B. Eckman (Eds.), Data Integration in the Life Sciences. XI, 298 pages. 2006. (Sublibrary LNBI).

Vol. 4074: M. Burmester, A. Yasinsac (Eds.), Secure Mobile Ad-hoc Networks and Sensors. X, 193 pages. 2006.

Vol. 4073: A. Butz, B. Fisher, A. Krüger, P. Olivier (Eds.), Smart Graphics. XI, 263 pages. 2006.

Vol. 4072: M. Harders, G. Székely (Eds.), Biomedical Simulation. XI, 216 pages. 2006.

Vol. 4071: H. Sundaram, M. Naphade, J.R. Smith, Y. Rui (Eds.), Image and Video Retrieval. XII, 547 pages. 2006.

Vol. 4070: C. Priami, X. Hu, Y. Pan, T.Y. Lin (Eds.), Transactions on Computational Systems Biology V. IX, 129 pages. 2006. (Sublibrary LNBI).

Vol. 4069: F.J. Perales, R.B. Fisher (Eds.), Articulated Motion and Deformable Objects. XV, 526 pages. 2006.

Vol. 4068: H. Schärfe, P. Hitzler, P. Øhrstrøm (Eds.), Conceptual Structures: Inspiration and Application. XI, 455 pages. 2006. (Sublibrary LNAI).

Vol. 4067: D. Thomas (Ed.), ECOOP 2006 – Object-Oriented Programming. XIV, 527 pages. 2006.

Vol. 4066: A. Rensink, J. Warmer (Eds.), Model Driven Architecture – Foundations and Applications. XII, 392 pages. 2006.

Vol. 4065: P. Perner (Ed.), Advances in Data Mining. XI, 592 pages. 2006. (Sublibrary LNAI).

Vol. 4064: R. Büschkes, P. Laskov (Eds.), Detection of Intrusions and Malware & Vulnerability Assessment. X, 195 pages. 2006.

Vol. 4063: I. Gorton, G.T. Heineman, I. Crnkovic, H.W. Schmidt, J.A. Stafford, C.A. Szyperski, K. Wallnau (Eds.), Component-Based Software Engineering. XI, 394 pages. 2006.

Vol. 4062: G. Wang, J.F. Peters, A. Skowron, Y. Yao (Eds.), Rough Sets and Knowledge Technology. XX, 810 pages. 2006. (Sublibrary LNAI).

Vol. 4061: K. Miesenberger, J. Klaus, W. Zagler, A. Karshmer (Eds.), Computers Helping People with Special Needs. XXIX, 1356 pages. 2006.

Vol. 4060: K. Futatsugi, J.-P. Jouannaud, J. Meseguer (Eds.), Algebra, Meaning and Computation. XXXVIII, 643 pages. 2006.

Vol. 4059: L. Arge, R. Freivalds (Eds.), Algorithm Theory – SWAT 2006. XII, 436 pages. 2006.

Vol. 4058: L.M. Batten, R. Safavi-Naini (Eds.), Information Security and Privacy. XII, 446 pages. 2006.

Vol. 4057: J.P. W. Pluim, B. Likar, F.A. Gerritsen (Eds.), Biomedical Image Registration. XII, 324 pages. 2006.

Vol. 4056: P. Flocchini, L. Gąsieniec (Eds.), Structural Information and Communication Complexity. X, 357 pages. 2006.

Vol. 4055: J. Lee, J. Shim, S.-g. Lee, C. Bussler, S. Shim (Eds.), Data Engineering Issues in E-Commerce and Services. IX, 290 pages. 2006.

Vol. 4054: A. Horváth, M. Telek (Eds.), Formal Methods and Stochastic Models for Performance Evaluation. VIII, 239 pages. 2006.

Vol. 4053: M. Ikeda, K.D. Ashley, T.-W. Chan (Eds.), Intelligent Tutoring Systems. XXVI, 821 pages. 2006.

Vol. 4052: M. Bugliesi, B. Preneel, V. Sassone, I. Wegener (Eds.), Automata, Languages and Programming, Part II. XXIV, 603 pages. 2006.

Vol. 4051: M. Bugliesi, B. Preneel, V. Sassone, I. Wegener (Eds.), Automata, Languages and Programming, Part I. XXIII, 729 pages. 2006.

Vol. 4049: S. Parsons, N. Maudet, P. Moraitis, I. Rahwan (Eds.), Argumentation in Multi-Agent Systems. XIV, 313 pages. 2006. (Sublibrary LNAI).

Vol. 4048: L. Goble, J.-J.C.. Meyer (Eds.), Deontic Logic and Artificial Normative Systems. X, 273 pages. 2006. (Sublibrary LNAI).

Vol. 4047: M. Robshaw (Ed.), Fast Software Encryption. XI, 434 pages. 2006.

Vol. 4046: S.M. Astley, M. Brady, C. Rose, R. Zwiggelaar (Eds.), Digital Mammography. XVI, 654 pages. 2006.

Vol. 4045: D. Barker-Plummer, R. Cox, N. Swoboda (Eds.), Diagrammatic Representation and Inference. XII, 301 pages. 2006. (Sublibrary LNAI).

Vol. 4044: P. Abrahamsson, M. Marchesi, G. Succi (Eds.), Extreme Programming and Agile Processes in Software Engineering. XII, 230 pages. 2006.

Vol. 4043: A.S. Atzeni, A. Lioy (Eds.), Public Key Infrastructure. XI, 261 pages. 2006.

Vol. 4042: D. Bell, J. Hong (Eds.), Flexible and Efficient Information Handling. XVI, 296 pages. 2006.

Vol. 4041: S.-W. Cheng, C.K. Poon (Eds.), Algorithmic Aspects in Information and Management. XI, 395 pages. 2006.

Vol. 4040: R. Reulke, U. Eckardt, B. Flach, U. Knauer, K. Polthier (Eds.), Combinatorial Image Analysis. XII, 482 pages. 2006.

Vol. 4039: M. Morisio (Ed.), Reuse of Off-the-Shelf Components. XIII, 444 pages. 2006.

Vol. 4038: P. Ciancarini, H. Wiklicky (Eds.), Coordination Models and Languages. VIII, 299 pages. 2006.

Vol. 4037: R. Gorrieri, H. Wehrheim (Eds.), Formal Methods for Open Object-Based Distributed Systems. XVII, 474 pages. 2006.

Vol. 4036: O. H. Ibarra, Z. Dang (Eds.), Developments in Language Theory. XII, 456 pages. 2006.

Vol. 4035: T. Nishita, Q. Peng, H.-P. Seidel (Eds.), Advances in Computer Graphics. XX, 771 pages. 2006.

Vol. 4034: J. Münch, M. Vierimaa (Eds.), Product-Focused Software Process Improvement. XVII, 474 pages. 2006.

Vol. 4033: B. Stiller, P. Reichl, B. Tuffin (Eds.), Performability Has its Price. X, 103 pages. 2006.

Vol. 4032: O. Etzion, T. Kuflik, A. Motro (Eds.), Next Generation Information Technologies and Systems. XIII, 365 pages. 2006.

Vol. 4031: M. Ali, R. Dapoigny (Eds.), Advances in Applied Artificial Intelligence. XXIII, 1353 pages. 2006. (Sublibrary LNAI).

Vol. 4029: L. Rutkowski, R. Tadeusiewicz, L.A. Zadeh, J. Zurada (Eds.), Artificial Intelligence and Soft Computing – ICAISC 2006. XXI, 1235 pages. 2006. (Sublibrary LNAI).

Vol. 4028: J. Kohlas, B. Meyer, A. Schiper (Eds.), Dependable Systems: Software, Computing, Networks. XII, 295 pages. 2006.

Vol. 4027: H.L. Larsen, G. Pasi, D. Ortiz-Arroyo, T. Andreasen, H. Christiansen (Eds.), Flexible Query Answering Systems. XVIII, 714 pages. 2006. (Sublibrary LNAI).

Vol. 4026: P.B. Gibbons, T. Abdelzaher, J. Aspnes, R. Rao (Eds.), Distributed Computing in Sensor Systems. XIV, 566 pages. 2006.

Vol. 4025: F. Eliassen, A. Montresor (Eds.), Distributed Applications and Interoperable Systems. XI, 355 pages. 2006.

Vol. 4024: S. Donatelli, P. S. Thiagarajan (Eds.), Petri Nets and Other Models of Concurrency - ICATPN 2006. XI, 441 pages. 2006.

Vol. 4021: E. André, L. Dybkjær, W. Minker, H. Neumann, M. Weber (Eds.), Perception and Interactive Technologies. XI, 217 pages. 2006. (Sublibrary LNAI).

Vol. 4020: A. Bredenfeld, A. Jacoff, I. Noda, Y. Takahashi (Eds.), RoboCup 2005: Robot Soccer World Cup IX. XVII, 727 pages. 2006. (Sublibrary LNAI).

Vol. 4019: M. Johnson, V. Vene (Eds.), Algebraic Methodology and Software Technology. XI, 389 pages. 2006.

Vol. 4018: V. Wade, H. Ashman, B. Smyth (Eds.), Adaptive Hypermedia and Adaptive Web-Based Systems. XVI, 474 pages. 2006.

Vol. 4017: S. Vassiliadis, S. Wong, T.D. Hämäläinen (Eds.), Embedded Computer Systems: Architectures, Modeling, and Simulation. XV, 492 pages. 2006.

Vol. 4016: J.X. Yu, M. Kitsuregawa, H.V. Leong (Eds.), Advances in Web-Age Information Management. XVII, 606 pages. 2006.

Vol. 4014: T. Uustalu (Ed.), Mathematics of Program Construction. X, 455 pages. 2006.

Vol. 4013: L. Lamontagne, M. Marchand (Eds.), Advances in Artificial Intelligence. XIII, 564 pages. 2006. (Sublibrary LNAI).

Vol. 4012: T. Washio, A. Sakurai, K. Nakajima, H. Takeda, S. Tojo, M. Yokoo (Eds.), New Frontiers in Artificial Intelligence. XIII, 484 pages. 2006. (Sublibrary LNAI).

Vol. 4011: Y. Sure, J. Domingue (Eds.), The Semantic Web: Research and Applications. XIX, 726 pages. 2006.

Vol. 4010: S. Dunne, B. Stoddart (Eds.), Unifying Theories of Programming. VIII, 257 pages. 2006.

Vol. 4009: M. Lewenstein, G. Valiente (Eds.), Combinatorial Pattern Matching. XII, 414 pages. 2006.

Vol. 4008: J.C. Augusto, C.D. Nugent (Eds.), Designing Smart Homes. XI, 183 pages. 2006. (Sublibrary LNAI).

Preface

This volume contains the papers presented at the 9th International Conference on Theory and Applications of Satisfiability Testing (SAT 2006).

The International Conference on Theory and Applications of Satisfiability Testing is the primary annual meeting for researchers studying the propositional satisfiability problem (SAT). SAT 2006 was part of FLoC 2006, the fourth Federated Logic Conference, which hosted, in addition to SAT, LICS, RTA, CAV, ICLP and IJCAR. SAT 2005 was held in St. Andrews, Scotland, and SAT 2004 in Vancouver, BC, Canada. This time SAT featured the SAT Race in spirit of the SAT Competitions, the first competitive QBF Evaluation, an Evaluation of Pseudo-Boolean Solvers and the Workshop on Satisfiability Solvers and Program Verification (SSPV).

Many hard combinatorial problems can be formulated as Boolean Satisfiability (SAT) problems. In fact, given the tremendous advances in the state of the art of SAT solvers in the last decade, many real-world applications are now being encoded as SAT problems. For example, many practical verification problems can be rephrased as SAT problems. This applies to verification problems in hardware and software. SAT is therefore becoming one of the most important core technologies to verify secure and dependable systems: Improvements in the theoretical and practical aspects of SAT will consequently apply to a range of real-world problems.

The topics of the conference spanned practical and theoretical research on SAT and its applications and included but were not limited to proof systems, proof complexity, search algorithms, heuristics, analysis of algorithms, hard instances, randomized formulae, problem encodings, industrial applications, solvers, simplifiers, tools, case studies and empirical results. SAT is interpreted in a rather broad sense: besides propositional satisfiability, it includes the domain of Quantified Boolean Formulae (QBF), Constraint Programming Techniques (CP) for word-level problems and their propositional encoding and particularly Satisfiability Modulo Theories (SMT).

There were 80 submissions including 75 regular papers with a page limit of 14 pages and 15 short papers with a page limit of 6 pages. Each submission was reviewed by at least three Programme Committee members. The committee decided to accept 26 regular papers and 11 short papers. Out of the 15 papers submitted as short papers, two papers were accepted. The other nine papers accepted as short papers had been submitted as regular paper.

The program also includes invited talks by Fahiem Bacchus and Karem Sakallah, the presentations of the results of the SAT Race, and the Evaluations of QBF, Pseudo-Boolean and MAX-SAT Solvers.

We would like to thank the organizers of FLoC for coordinating the different conferences. We thank Andrei Voronkov for his excellent EasyChair system.

It helped us streamline the reviewing progress tremendously and meet our decision deadlines. Last but not least we thank the Programme Committee and the additional external reviewers for their careful and thorough work, without which it would not have been possible for us to put together such an outstanding conference programme.

We would also like to acknowledge the support of our sponsors: Cadence, IBM, Microsoft Research, NEC, John von Neumann Minerva Center for the Development of Reactive Systems, and the Intelligent Information Systems Institute at Cornell University.

August 2006 Armin Biere and Carla P. Gomes

Organization

Programme Chairs

Armin Biere
Carla P. Gomes

FLoC Representative

Henry Kautz

Programme Committee

Dimitris Achlioptas
Carlos Ansótegui
Fahiem Bacchus
Paul Beame
Alessandro Cimatti
Niklas Eén
Enrico Giunchiglia
Holger Hoos
Henry Kautz
Hans Kleine Büning
James Kukula
Daniel Le Berre
Inês Lynce
Hans van Maaren
Sharad Malik
João Marques-Silva
Cristopher Moore
Jussi Rintanen
Ashish Sabharwal
Bart Selman
Carsten Sinz
Ewald Speckenmeyer
Ofer Strichman
Stefan Szeider
Allen Van Gelder
Miroslav Velev
Toby Walsh
Riccardo Zecchina
Lintao Zhang

External Reviewers

Johan Alfredsson
Anbulagan
Gilles Audemard
Ramón Béjar
Marco Benedetti
Wolfgang Blochinger
Maria Luisa Bonet
Marco Bozzano
Hajo Broersma
Uwe Bubeck
Amin Coja-Oghlan
Sylvie Coste-Marquis
Stefan Dantchev
Ivan Dotú
Anders Franzén
Zhaohui Fu
Roman Gershman
Eugene Goldberg
Dan Goldwasser
Emmanuel Hebrard
Marijn Heule
Jinbo Huang
Frank Hutter
Gabriel Istrate
Maya Koifman
Andrei Krokhin
Oliver Kullmann
Theodor Lettmann
Chu-Min Li
Jean Christophe Madre
Vasco Manquinho
Felip Manyà
Marco Maratea
Pierre Marquis
Massimo Narizzano
Arlindo Oliveira
Stefan Porschen
Steven Prestwich
Bert Randerath
Silvio Ranise
Marco Roveri
Marko Samer
Tian Sang
Vishal Sanwalani
Roberto Sebastiani
Andrew Slater
Greg Sorkin
Ted Stanion
Dominik Stoffel
Peter Stuckey
Christian Szegedy
Niklas Sörensson
John Thornton
Dave Tompkins
Michael Veksler
Dong Wang
Karen Yorav
Yinlei Yu

Table of Contents

Invited Talks

Session 1. Proofs and Cores

Session 2. Heuristics and Algorithms

Session 3. Applications

Session 4. SMT

Session 5. Structure

Session 6. MAX-SAT

Session 7. Local Search and Survey Propagation

Session 8. QBF

Session 9. Counting and Concurrency

From Propositional Satisfiability to Satisfiability Modulo Theories

Hossein M. Sheini and Karem A. Sakallah

University of Michigan, Ann Arbor MI 48109, USA
{hsheini, karem}@umich.edu

Abstract. In this paper we present a review of SAT-based approaches for building scalable and efficient decision procedures for quantifier-free first-order logic formulas in one or more decidable theories, known as Satisfiability Modulo Theories (SMT) problems. As applied to different system verification problems, SMT problems comprise of different theories including fragments of elementary theory of numbers, the theory of arrays, the theory of list structures, etc. In this paper we focus on different DPLL-style satisfiability procedures for decidable fragments of the theory of integers. Leveraging the advances made in SAT solvers in the past decade, we introduce several SAT-based SMT solving methods that in many applications have outperformed classical decision methods. Aside from the classical method of translating the SMT formula to a purely Boolean problem, in recent methods, a SAT solver is utilized to serve as the "glue" that ties together the different theory atoms and forms the basis for reasoning and learning within and across them. Several methods have been developed to provide a combination framework for implications to flow through the theory solvers and to possibly activate other theory atoms based on the current assignments. Similarly, conflict-based learning is also extended to enable the creation of learned clauses comprising of the combination of theory atoms. Additional methods unique to one or more types of theory atoms have also been proposed that learn more expressive constraints and significantly increase the pruning power of these combination schemes. We will describe several combination strategies and their impact on scalability and performance of the overall solver in different settings and applications.

1 Introduction

The decision problem for quantifier-free first-order logic formulas arises quite naturally in a wide variety of applications including software and hardware verification, scheduling and planning [6,2,31,9,27]. Such formulas typically consist of logical combinations of atoms from different theories such as the theory of integer linear arithmetic, the theory of arrays, the theory of equality with uninterpreted functions, set theory, etc. Systematic procedures for deciding such formulas, based on equality propagation among the different theory solvers, were first described by Nelson and Oppen [20]. More recently, interest in this problem has increased dramatically, sparked in part by the phenomenal progress in the

A. Biere and C.P. Gomes (Eds.): SAT 2006, LNCS 4121, pp. 1–9, 2006.

capacity and speed of modern DPLL-based Boolean satisfiability (SAT) solvers. A significant number of researchers around the globe are actively exploring a variety of approaches for solving this problem by leveraging the computational power of modern SAT solvers. In this new incarnation, the problem has been dubbed Satisfiability Modulo Theories (SMT) [29] with particular emphasis on integrating the theory of linear (real and integer) arithmetic within a DPLL backtrack search framework[1].

In this paper we provide a brief survey of the SAT-based approaches for solving SMT problems. Specifically, after covering some preliminaries in Section 2, we describe methods for direct translation of SMT instances to SAT (Section 3), methods based on Abstraction/refinement (Section 4), online approaches (Section 5) and hybrid solutions methods (Section 6).

2 Preliminaries

Satisfiability Modulo Theories (SMT) is the problem of determining the satisfiability of a quantifier-free first-order logic (FOL) formula in one or more decidable theories. Quantifier-free first-order logic extends propositional logic with *terms*, *function symbols*, and *predicate symbols* defined according to the following rules:

- A variable is a term.
- $f(t_1, \cdots, t_n)$ is a *term*, where f is an n-arity function symbol (with $n \geq 0$) and $t_1, \cdots, t_n$ are terms. 0-arity functions are *constants.*
- $P(t_1, \cdots, t_n)$ is an *atom*, where P is an n-arity predicate symbol (with $n \geq 0$) and $t_1, \cdots, t_n$ are terms. 0-arity predicates are *propositional variables.*
- Quantifier-free FOL formulas are constructed by combining atoms according to the rules of propositional logic. In particular, a quantifier-free FOL formula in *conjunctive normal form* (CNF) is the conjunction of a set of *clauses* each of which is the disjunction of a set of *literals*, where a literal is either an atom or the negation of an atom.

Using quantifier-free FOL as an organizing framework, we can define several specialized "logics" or "theories" by specifying the domains of their variables, as well as the functions (terms) and predicates (atoms) they admit. We list below some of the most commonly-used theories in hardware and software verification applications. Note that in these theories all variables and constant are assumed to be integer-valued.

- **Propositional Logic ($\mathcal{P}$):** In this theory, there are no function symbols, and only 0-arity predicate symbols, i.e., propositional variables.
- **Equality Logic ($\mathcal{E}$):** This theory has no function symbols and only the equality predicate $t_i = t_j$ where t_i and t_j are terms.
- **Equality Logic with Successors ($\mathcal{ES}$):** This logic extends $\mathcal{E}$ logic by introducing the function $succ(t) = t + 1$ where t is a term.

[1] Independently, Hooker et al [13] and the OR community refer to similar problems as mixed logical linear programs.

- **Equality Logic with Uninterpreted Functions ($\mathcal{EUF}$):** This logic extends $\mathcal{E}$ with n-arity uninterpreted function and predicate symbols and enforces functional consistency.
- **Equality Logic with Successors and Uninterpreted Functions ($\mathcal{ESUF}$):** This logic extends $\mathcal{ES}$ with n-arity uninterpreted function and predicate symbols and enforces functional consistency.
- **Difference Logic ($\mathcal{DL}$):** This logic extends $\mathcal{ES}$ with the interpreted predicate of the form $t_i - t_j \leq d$ where t_i and t_j are terms and d is an integer constant.
- **Counter Arithmetic Logic with Lambda Expressions and Uninterpreted Functions ($\mathcal{CLU}$):** The set of functions and predicates are respectively the union of the functions and predicates of $\mathcal{EUF}$ and $\mathcal{DL}$.
- **Integer Unit-Two-Variable-Per-Inequality (UTVPI) Logic ($\mathcal{TVL}$):** This logic generalizes the interpreted predicates of $\mathcal{DL}$ to the form $a_i t_i + a_j t_j \leq d$ where $a_i, a_j \in \{\pm 1, 0\}$, t_i and t_j are terms and d is an integer constant.
- **Linear Integer Arithmetic ($\mathcal{LIA}$):** This is essentially the logic of integer linear inequalities. Note that $\mathcal{DL}$ and $\mathcal{TVL}$ are restrictions of $\mathcal{LIA}$.

The choice of which logic to apply in a particular situation depends on the expressiveness of the logic as well as the existence of efficient procedures for checking the satisfiability of conjunctions of atoms in that logic. Such procedures are referred to as *theory solvers* include congruence-closure for the logic of equality and its extensions [22], transitive-closure for the $\mathcal{DL}$ and $\mathcal{TVL}$ theories [15] and Simplex-based Branch-and-Bound algorithms for $\mathcal{LIA}$.

Propositional Satisfiability (SAT). Modern SAT solvers are based on the DPLL backtrack search algorithm [8] augmented with powerful techniques for search space pruning and efficient Boolean constraint propagation. These techniques include conflict-based learning and non-chronological backtracking [18], watched-literal schemes for Boolean Constraint Propagation and advanced variable ordering heuristics such as VSIDS [19]. For a survey of these methods, the reader is referred to [32].

The efficiency of the SMT solvers we describe on the remainder of this paper drive their power from these modern SAT solvers.

3 Translation to SAT

The earliest use of SAT solvers in the SMT context was through direct translation of an SMT instance to an equi-satisfiable Boolean formula. Such *eager* solution approaches were attractive because they did not require the development of specialized theory solvers or complex combination strategies. Their effectiveness derived from their reliance on the underlying SAT engine.

These approaches were most effective when applied to the $\mathcal{EUF}$ and $\mathcal{CLU}$ logics in [14,16,23,24,7]. Techniques to translate such formulas to propositional form include *Small Domain Instantiation* [23] and *Per-Constraint Encoding* [25].

Small Domain Instantiation. This method is based on the fact that for a formula whose atoms are only equalities between *input* variables, it is enough to give each variable the range $[1 \dots n]$ (where n is the number of input variables) without affecting the satisfiability/unsatisfiability of the formula. Knowing the range enables us to replace each input variable in the formula with a bit-vector of Boolean variables of size $\lceil \log n \rceil$ yielding a purely Boolean instance. To apply this procedure to an $\mathcal{EUF}$ formula, the formula is first converted, using Ackermann reduction [1], to an equi-satisfiable formula that only involves equaliteis. Specifically, each occurrence of an uninterpreted function is replaced with a new variable and the formula is conjoined with constraints to preserve functional consistency.

Further enhancements to this method deal with reducing the $[1 \dots n]$ range associated with each variable taking into account the structure of the formula. These methods include the equality graph and range analysis of [23], coloring the equality graph by analyzing the CNF representation of the formula in [12], positive equality method of [30] and the hybrid method of [24].

Per-Constraint Encoding. In this approach, also known as e_{ij} *encoding* [11], an $\mathcal{EUF}$ formula is first transformed, as above, into an equi-satisfiable formula, φ, whose atoms are only equalities. Each equality atom $t_i = t_j$ is replaced by a fresh unrestricted Boolean variable e_{ij}, yielding a Boolean abstraction, φ^{bool} that can be processed by a propositional SAT solver. To prevent false positives this abstraction is augmented with transitivity constraints of the form $(e_{ij} \wedge e_{jk}) \rightarrow e_{ik}$ for all e_{ij} variables within the formula. In the worst case the number of such transitively constraints grow exponentially [7].

4 Abstraction/Refinement

An SMT CNF formula whose propositional and theory atoms are denoted, respectively, by $\mathscr{P}$ and $\mathscr{T}$ can be generically expressed as

$$\varphi(\mathscr{P}, \mathscr{T}) = \bigwedge_{c\ \in \mathscr{C}} c \tag{1}$$

where $\mathscr{C}$ is a set of clauses whose elements have the form

$$\bigvee_{A \in\ \mathscr{P} \cup \mathscr{T}} A \ \vee \bigvee_{A \in\ \mathscr{P} \cup \mathscr{T}} \neg A \tag{2}$$

A Boolean abstraction of the SMT formula, φ^{bool}, can be constructed by introducing a mapping α that assigns a fresh Boolean *indicator* variable, $\alpha(A)$, to each theory atom, A. Thus the SMT CNF formula can be represented as

$$\varphi(\mathscr{P}, \mathscr{I}, \mathscr{T}) = \varphi^{bool}(\mathscr{P}, \mathscr{I}) \wedge \bigwedge_{A \in \mathscr{T}} (\alpha(A) \leftrightarrow A) \tag{3}$$

where $\mathscr{I}$ is the set of indicator variables. In the abstraction/refinement approach, a SAT solver is initially applied to φ^{bool}. If φ^{bool} is found to be unsatisfiable,