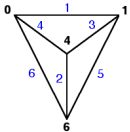
COMPETITIONS AS SCIENTIFIC METHOD

Lecture Series in Artificial Intelligence

JOHANNES KEPLER
UNIVERSITY LINZ

Practical Applications of SAT Solving

formal verification

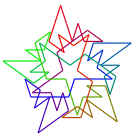


graph theory

bioinformatics

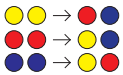
train safety

planning

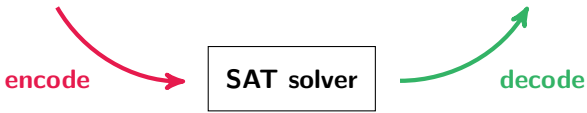


combinatorics

cryptography



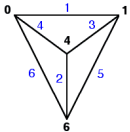
rewrite termination



from http://www.cs.utexas.edu/users/marijn/talks/Ptn-Linz.pdf

Practical Applications of SAT Solving

formal verification



graph theory

bioinformatics

train safety

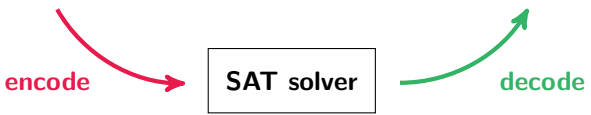
Where do solvers come from?

planning

combinatorics

cryptography

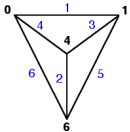
rewrite termination



from http://www.cs.utexas.edu/users/marijn/talks/Ptn-Linz.pdf

Practical Applications of SAT Solving

formal verification



graph theory

bioinformatics

train safety

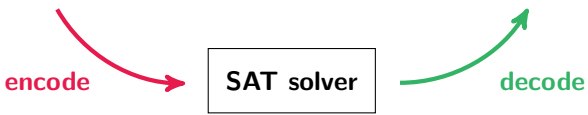
Which solver to use?

planning

combinatorics

cryptography

rewrite termination



from http://www.cs.utexas.edu/users/marijn/talks/Ptn-Linz.pdf

Where Do Solvers Come From?

many solvers come from academia as research output

Where Do Solvers Come From?

many solvers come from academia as research output

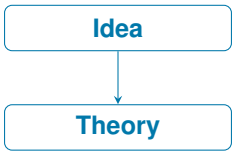
Idea

(simplified/idealized)
scientific process

Where Do Solvers Come From?

many solvers come from academia as research output

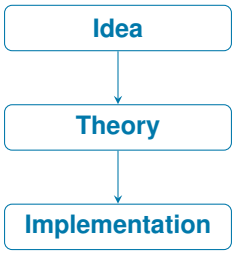
(simplified/idealized) scientific process



Where Do Solvers Come From?

many solvers come from academia as research output

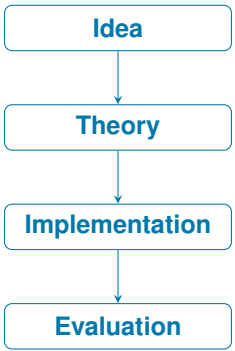
(simplified/idealized) scientific process



Where Do Solvers Come From?

many solvers come from academia as research output

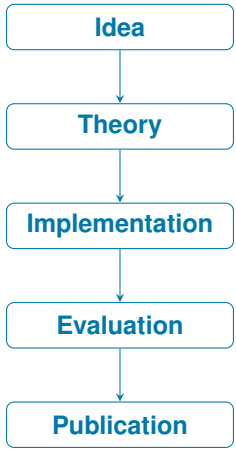
(simplified/idealized) scientific process



Where Do Solvers Come From?

many solvers come from academia as research output

(simplified/idealized) scientific process



Journal of Artificial Intelligence Research 53 (2015) 127-168 Submitted 11/15; published 6/15

Clause Elimination for SAT and QSAT

Marijn Heule MARIJN@CS.UTEXAS.EDU
Department of Computer Science,
The University of Texas at Austin, USA
Matti Järvisalo MATTI.JARVISALO@CS.HELSINKI.FI
HIIT, Department of Computer Science,
University of Helsinki, Finland
Florian Lonsing FLORIAN.LONSING@TUWIEN.AC.AT
Institute of Information Systems,
Vienna University of Technology, Austria
Martina Seidl MARTINA.SEIDL@JKU.AT
Armin Biere BIERE@JKU.AT
Institute for Formal Models and Verification,
Johannes Kepler University Linz, Austria

Abstract

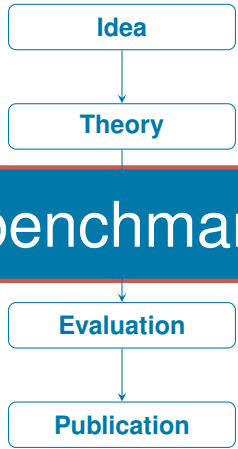
The famous archetypical NP-complete problem of *Boolean satisfiability* (SAT) and its PSPACE-complete generalization of *quantified Boolean satisfiability* (QSAT) have become central declarative programming paradigms through which real-world instances of various computationally hard problems can be efficiently solved. This success has been achieved through several breakthroughs in practical implementations of decision procedures for SAT and QSAT, that is, in SAT and QSAT solvers. Here, simplification techniques for conjunctive normal form (CNF) for SAT and for prenex conjunctive normal form (PCNF) for QSAT—the standard input formats of SAT and QSAT solvers—have recently proven very effective in increasing solver efficiency when applied before (i.e., *in preprocessing*) or during (i.e., *in inprocessing*) satisfiability search.

In this article, we develop and analyze clause elimination procedures for pre- and inprocessing. Clause elimination procedures form a family of (P)CNF formula simplification techniques which remove clauses that have specific (in practice polynomial-time) redundancy properties while maintaining the satisfiability status of the formulas. Extending known procedures such as tautology, subsumption, and blocked clause elimination, we introduce novel elimination procedures based on *asymmetric* variants of these techniques, and also develop a novel family of so-called *covered clause elimination* procedures, as well as natural liftings of the CNF-level procedures to PCNF. We analyze the considered clause elimination procedures from various perspectives. Furthermore, for the variants not preserving logical equivalence under clause elimination, we show how to reconstruct solutions to original CNFs from satisfying assignments to simplified CNFs, which is important for practical applications for the procedures. Complementing the more theoretical analysis, we present results on an empirical evaluation on the practical importance of the clause elimination procedures in terms of the effect on solver runtimes on standard real-world application benchmarks. It turns out that the importance of applying the clause elimination procedures developed in this work is empirically emphasized in the context of state-of-the-art QSAT solving.

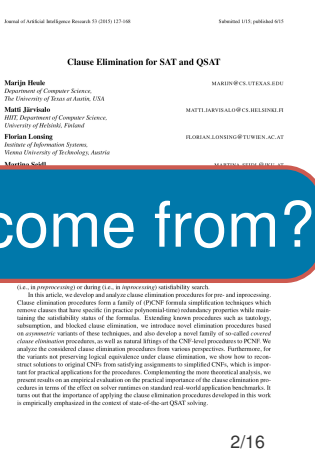
Where Do Solvers Come From?

many solvers come from academia as research output

(simplified/idealized) scientific process



Where do the benchmarks come from?


Journal of Artificial Intelligence Research 53 (2015) 127-168 Submitted 11/15; published 6/15

Clause Elimination for SAT and QSAT

Marijn Heule MARIJN@CS.UTEXAS.EDU
Department of Computer Science,
The University of Texas at Austin, USA
Matti Järvisalo MATTI.JARVISALO@CS.HELSINKI.FI
HIIT, Department of Computer Science,
University of Helsinki, Finland
Florian Lonsing FLORIAN.LONSING@TUWIEN.AC.AT
Institute of Information Systems,
Vienna University of Technology, Austria
Martina Seidl MARTINA.SEIDL@JKU.AT

(i.e., in *preprocessing*) or during (i.e., in *inprocessing*) satisfiability search.

In this article, we develop and analyze clause elimination procedures for pre- and inprocessing. Clause elimination procedures form a family of (P)CNF formula simplification techniques which remove clauses that have specific (in practice polynomial-time) redundancy properties while maintaining the satisfiability status of the formulas. Extending known procedures such as tautology, subsumption, and blocked clause elimination, we introduce novel elimination procedures based on *asymmetric* variants of these techniques, and also develop a novel family of so-called *covered clause elimination* procedures, as well as natural liftings of the CNF-level procedures to PCNF. We analyze the considered clause elimination procedures from various perspectives. Furthermore, for the variants not preserving logical equivalence under clause elimination, we show how to reconstruct solutions to original CNFs from satisfying assignments to simplified CNFs, which is important for practical applications for the procedures. Complementing the more theoretical analysis, we present results on an empirical evaluation on the practical importance of the clause elimination procedures in terms of the effect on solver runtimes on standard real-world application benchmarks. It turns out that the importance of applying the clause elimination procedures developed in this work is empirically emphasized in the context of state-of-the-art QSAT solving.

Requirements on Solvers

- fast
- correct
- stable
- configurable
- easy to use / follow standards
- publicly available
- available for recent operating systems
- maintained
- ...

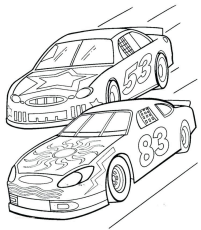
Requirements on Solvers

- fast
- correct
- stable
- configurable
- easy to use / follow standards
- publicly available
- available for recent operating systems
- maintained
- ...

$\Rightarrow$ **Tool Competitions**

Goals of Competitions

- make tools easy to use
  - establish input / output standards
  - establish standard API
  - well working configurations
- promote development of solving tools
- identify challenging benchmarks
- snapshot of current solvers
- archive/documentation of tools
- archive/documentation of benchmarks
- entertainment



The Competition Landscape

Reactive Synthesis Competition

Sparkle SAT Challenge

Hardware Model Checking

SAT Race

MaxSAT Evaluation

Planning Competition

SMT-COMP

Termination Competition

SV-Comp

VerifyThis

QComp

LP/CP Programming Competition

Software Competitions: Key Facts

participants:

open to everybody, some requirements on tools (e.g., open source)

location:

usually virtual, few onsite competition

competition execution:

usually offline, few competitions have live monitor

result presentation:

conference / workshop of research community

General Setup

1. preparatory work
   - setup of rules / organization
   - setup of schedule
   - preparation of tools (e.g., converters)

General Setup

1. preparatory work
   - setup of rules / organization
   - setup of schedule
   - preparation of tools (e.g., converters)
2. call for contribution
   - call for tools
   - call for instances

General Setup

1. preparatory work
   - setup of rules / organization
   - setup of schedule
   - preparation of tools (e.g., converters)
2. call for contribution
   - call for tools
   - call for instances
3. benchmark selection & solver testing

General Setup

1. preparatory work
   - setup of rules / organization
   - setup of schedule
   - preparation of tools (e.g., converters)
2. call for contribution
   - call for tools
   - call for instances
3. benchmark selection & solver testing
4. competition phase

General Setup

1. preparatory work
   - setup of rules / organization
   - setup of schedule
   - preparation of tools (e.g., converters)
2. call for contribution
   - call for tools
   - call for instances
3. benchmark selection & solver testing
4. competition phase
5. result announcement

General Setup

1. preparatory work
   - setup of rules / organization
   - setup of schedule
   - preparation of tools (e.g., converters)
2. call for contribution
   - call for tools
   - call for instances
3. benchmark selection & solver testing
4. competition phase
5. result announcement
6. post processing

Benchmark Selection

types of benchmarks

- applications
- crafted
- random

Benchmark Selection

types of benchmarks

- applications
- crafted
- random

challenges

- not too easy
- not too hard
- fair

Benchmark Selection

types of benchmarks

- applications
- crafted
- random

challenges

- not too easy
- not too hard
- fair

approaches

- run pretests
- look at results from previous years
- use structural information

benchmark selection is

- **pretty tricky**
- **a research question on its own**

Execution Environment StarExec

- cross community logic solving service by the University of Iowa for
  - store, manage, and publish benchmark libraries
  - run competitions
- cluster of 200 nodes with Xeon CPU 2.4GH with 260GB of main memory
- web interface
  - upload and manage solvers
  - upload and manage benchmarks
  - manage experiments

Scoring

winning criteria

- most solved instances
- shortest running time
- smallest solution
- ...

Scoring

winning criteria

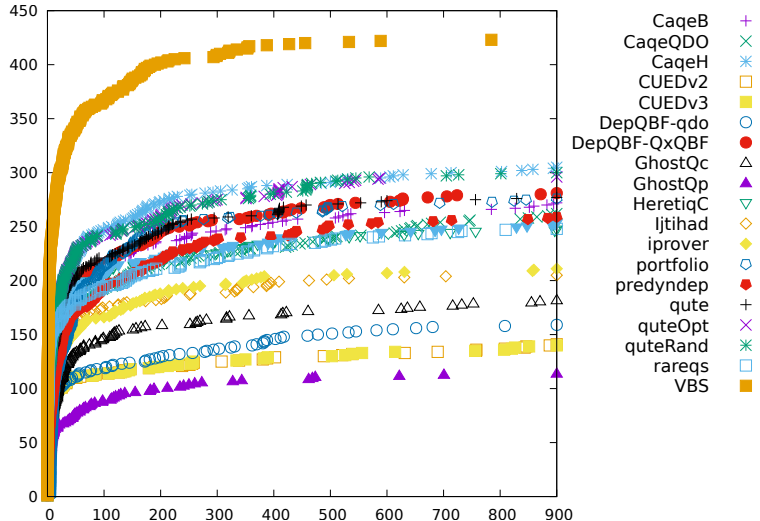
- most solved instances
- shortest running time
- smallest solution
- ...

questions

- How to deal with discrepancies?
- How to deal with timeouts/memory outs/crashes?
- How to deal with incomplete approaches?

usually there are separate awards for the different tracks

Virtual Best Solver (VBS)



Result Validation

How do we know that the results of the solvers are correct?

Result Validation

How do we know that the results of the solvers are correct?

1. by construction of the benchmarks

Result Validation

How do we know that the results of the solvers are correct?

1. by construction of the benchmarks

   ... this information is not always available

Result Validation

How do we know that the results of the solvers are correct?

1. by construction of the benchmarks

   ... this information is not always available
2. by majority vote (for dealing with discrepancies)

Result Validation

How do we know that the results of the solvers are correct?

1. by construction of the benchmarks

   ... this information is not always available
2. by majority vote (for dealing with discrepancies)

   ... what if only one solver finds a solution?

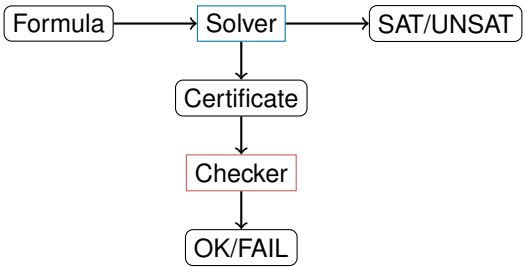
Result Validation

How do we know that the results of the solvers are correct?

1. by construction of the benchmarks

   ... this information is not always available
2. by majority vote (for dealing with discrepancies)

   ... what if only one solver finds a solution?
3. by certificates



Result Presentation and Award Ceremony

usually the results and winners are presented at a conference or workshop like

- Int. Conference on Theory and Applications of Satisfiability Testing
- Int. Conference on Automated Deduction
- Int. Conference on Tools and Algorithms for the Construction and Analysis of Systems
- ...

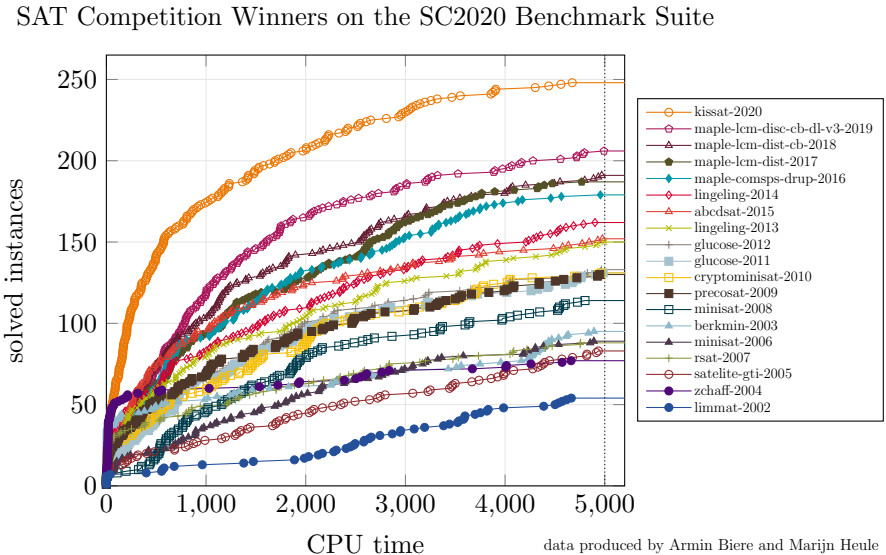
Result Presentation and Award Ceremony

usually the results and winners are presented at a conference or workshop like

- Int. Conference on Theory and Applications of Satisfiability Testing
- Int. Conference on Automated Deduction
- Int. Conference on Tools and Algorithms for the Construction and Analysis of Systems
- ...

some impressions of Olympic Game Ceremony in 2014

Measuring Progress



data produced by Armin Biere and Marijn Heule

“My Competition”: QBFEval

quantified Boolean Formula (QBF) = propositional formula + quantifiers

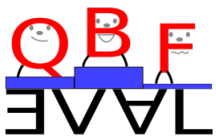
$$\forall x \exists y (x \leftrightarrow y)$$

“My Competition”: QBFEval

quantified Boolean Formula (QBF) = propositional formula + quantifiers

$$\forall x \exists y (x \leftrightarrow y)$$

2003	1st QBFEval
2013, 2014	QBFGallery
2020	13th QBFEval

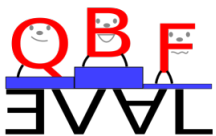


“My Competition”: QBFEval

quantified Boolean Formula (QBF) = propositional formula + quantifiers

$$\forall x \exists y (x \leftrightarrow y)$$

2003	1st QBFEval
2013, 2014	QBFGallery
2020	13th QBFEval



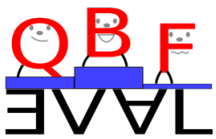
- co-organization with L. Pulina and Ankit Shukla
- affiliated with QBF Workshop & SAT Conference
- QBFGallery 2014 & QBFEval 2018 were part of FLoC Olympic Games

“My Competition”: QBFEval

quantified Boolean Formula (QBF) = propositional formula + quantifiers

$$\forall x \exists y (x \leftrightarrow y)$$

2003	1st QBFEval
2013, 2014	QBFGallery
2020	13th QBFEval



- co-organization with L. Pulina and Ankit Shukla
- affiliated with QBF Workshop & SAT Conference
- QBFGallery 2014 & QBFEval 2018 were part of FLoC Olympic Games

- tracks: PCNF, Prenex non-CNF, DQBF, . . .
- approx. 30 participating tools

Software Competitions

- ... provide an objective evaluation of the state of the art
- ... are an important driver for scientific progress
- ... set standards
- ... increase the visibility of a research community
- ... are fun