Clausal Proofs of Mutilated Chessboards

Marijn J.H. Heule, Benjamin Kiesl, and Armin Biere

UT Austin, CISPA — Helmholtz Center, and JKU Linz

NASA Formal Methods May 8, 2019

Certificates

What makes a problem hard?

Certificate angle: can one efficiently check an alleged solution?

Consider chess: does white begin and win?

A winning strategy will be very costly to check.

Certificates

What makes a problem hard?

Certificate angle: can one efficiently check an alleged solution?

Consider chess: does white begin and win?

A winning strategy will be very costly to check.

Consider the sudoku on the right:
Is searching for the solution harder than verifying a given solution?

	4		3					
						7	9	
			6					
			1	4		5		
9							1	
2								6
				7	2			
	5					8		
				9				

Certificates

What makes a problem hard?

Certificate angle: can one efficiently check an alleged solution?

Consider chess: does white begin and win?

A winning strategy will be very costly to check.

Consider the sudoku on the right:
Is searching for the solution harder than verifying a given solution?

Intuition: yes!

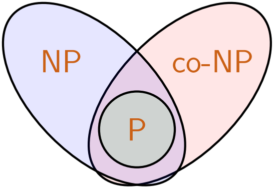
However, many problems for which we can efficiently check a solution turn out to be easy in practice.

1	4	*7*	3	*8*	*9*	*2*	*6*	*5*
5	*8*	*6*	*2*	*1*	*4*	7	9	*3*
3	*9*	*2*	6	*5*	*7*	*1*	*8*	*4*
8	*7*	*3*	1	4	*6*	5	*2*	*9*
9	*6*	*4*	*7*	*2*	*5*	*3*	1	*8*
2	*1*	*5*	*9*	*3*	*8*	*4*	*7*	6
6	*3*	*8*	*5*	7	2	*9*	*4*	*1*
7	5	*9*	*4*	*6*	*1*	8	*3*	*2*
4	*2*	*1*	*8*	9	*3*	*6*	*5*	*7*

Certificates and Complexity

Complexity classes of decision problems:

P : efficiently computable answers.
NP : efficiently checkable yes-answers.
co-NP : efficiently checkable no-answers.



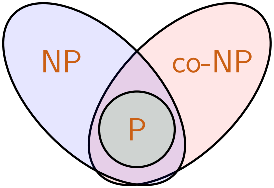
Cook-Levin Theorem [1971]: SAT is NP-complete.

Solving the $P \stackrel{?}{=} NP$ question is worth $1,000,000 [Clay MI '00].

Certificates and Complexity

Complexity classes of decision problems:

P : efficiently computable answers.
NP : efficiently checkable yes-answers.
co-NP : efficiently checkable no-answers.



Cook-Levin Theorem [1971]: SAT is NP-complete.

Solving the P $\stackrel{?}{=}$ NP question is worth $1,000,000 [Clay MI '00].

The beauty of NP: guaranteed short solutions.

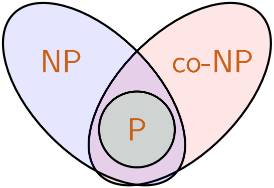
The effectiveness of SAT solving: fast solutions in practice.

"NP is the new P!"

Certificates and Complexity

Complexity classes of decision problems:

P : efficiently computable answers.
NP : efficiently checkable yes-answers.
co-NP : efficiently checkable no-answers.



Cook-Levin Theorem [1971]: SAT is NP-complete.

Solving the P $\stackrel{?}{=}$ NP question is worth $1,000,000 [Clay MI '00].

The beauty of NP: guaranteed short solutions.

The effectiveness of SAT solving: fast solutions in practice.

"NP is the new P!"

What about co-NP?

How to find short proofs for interesting problems efficiently?

Original Motivation for Producing and Validating Proofs

Automated reasoning tools may give incorrect answers.

- Documented bugs in SAT, SMT, and QSAT solvers;
- Implementation errors often imply conceptual errors;
- Proofs now mandatory in some competitive events;
  [Balyo, Heule, and Järvisalo '17]
- Mathematical results require a stronger justification than a simple yes/no by a tool. Answers must be verifiable.

Original Motivation for Producing and Validating Proofs

Automated reasoning tools may give incorrect answers.

- Documented bugs in SAT, SMT, and QSAT solvers;
- Implementation errors often imply conceptual errors;
- Proofs now mandatory in some competitive events;
  [Balyo, Heule, and Järvisalo '17]
- Mathematical results require a stronger justification than a simple yes/no by a tool. Answers must be verifiable.

Major challenge:

- Some "simple" problems have exponentially large proofs in the resolution proof system [Urquhart '87, Buss and Pitassi '98];
- While some dedicated techniques can quickly solve them.

Original Motivation for Producing and Validating Proofs

Automated reasoning tools may give incorrect answers.

- Documented bugs in SAT, SMT, and QSAT solvers;
- Implementation errors often imply conceptual errors;
- Proofs now mandatory in some competitive events;
  [Balyo, Heule, and Järvisalo '17]
- Mathematical results require a stronger justification than a simple yes/no by a tool. Answers must be verifiable.

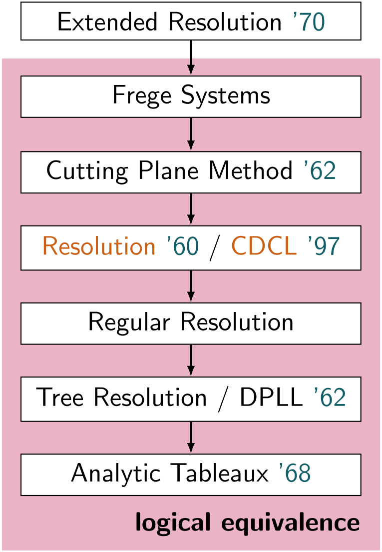
Major challenge:

- Some "simple" problems have exponentially large proofs in the resolution proof system [Urquhart '87, Buss and Pitassi '98];
- While some dedicated techniques can quickly solve them.

Requires a proof system to compactly express all techniques.

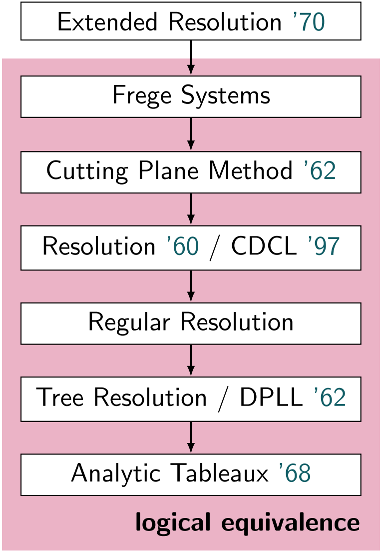
Proof Search in Strong Proof Systems

Existence of Short Proofs

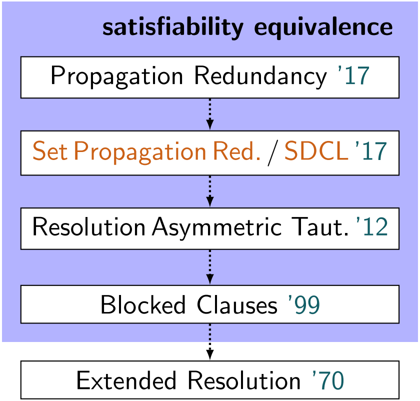


Proof Search in Strong Proof Systems

Existence of Short Proofs



Finding Short Proofs



Express solving techniques compactly
[Järvisalo, Heule, and Biere '12]

Short proofs without new variables
[Heule, Kiesl, and Biere '17A]

Satisfaction-Driven Clause Learning [Heule, Kiesl, Biere '17B]

SDCL generalizes CDCL and finds proofs in the SPR proof system.

CDCL in a nutshell:

1. Main loop combines efficient problem simplification with cheap, but effective decision heuristics; ($>$ 90% of time)
2. Reasoning kicks in if the current state is conflicting;
3. The current state is analyzed and turned into a constraint;
4. The constraint is added to the problem, the heuristics are updated, and the algorithm (partially) restarts.

Satisfaction-Driven Clause Learning [Heule, Kiesl, Biere '17B]

SDCL generalizes CDCL and finds proofs in the SPR proof system.

SDCL in a nutshell:

1. Main loop combines efficient problem simplification with cheap, but effective decision heuristics; ($>$ 90% of time)
2. Reasoning kicks in if the current state is conflicting;
2. Reasoning kicks in if there exists a state that is at least as satisfiable as the current state; (*NP-complete check*)
3. The current state is analyzed and turned into a constraint;
4. The constraint is added to the problem, the heuristics are updated, and the algorithm (partially) restarts.

Satisfaction-Driven Clause Learning [Heule, Kiesl, Biere '17B]

SDCL generalizes CDCL and finds proofs in the SPR proof system.

SDCL in a nutshell:

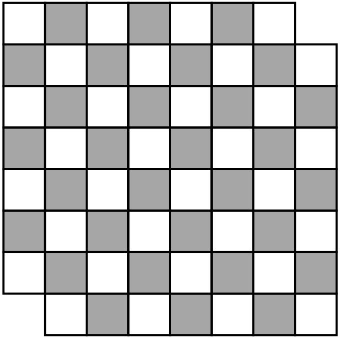
1. Main loop combines efficient problem simplification with cheap, but effective decision heuristics; ($>$ 90% of time)
2. Reasoning kicks in if the current state is conflicting;
2. Reasoning kicks in if there exists a state that is at least as satisfiable as the current state; (*NP-complete check*)
3. The current state is analyzed and turned into a constraint;
4. The constraint is added to the problem, the heuristics are updated, and the algorithm (partially) restarts.

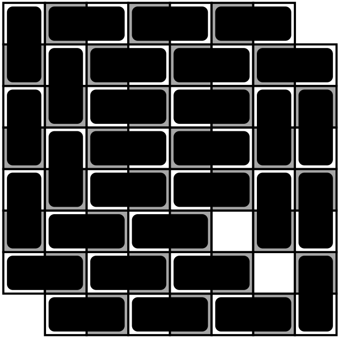
Short proofs for problems that are hard for resolution

including pigeonhole, Tseitin, and mutilated chessboard problems

Mutilated Chessboards: "A Tough Nut to Crack" [McCarthy]

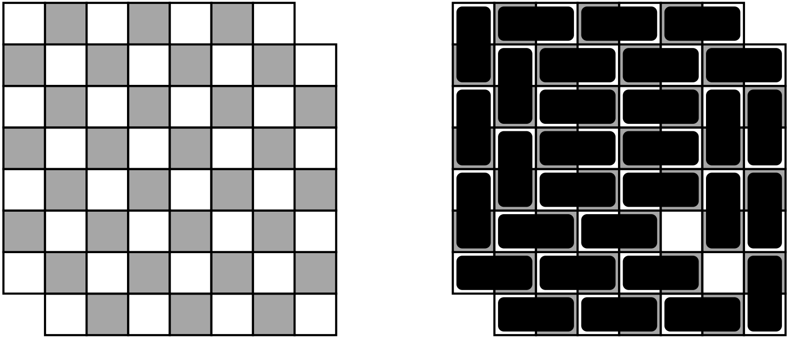
Can a chessboard be fully covered with dominos after removing two diagonally opposite corner squares?





Mutilated Chessboards: "A Tough Nut to Crack" [McCarthy]

Can a chessboard be fully covered with dominos after removing two diagonally opposite corner squares?

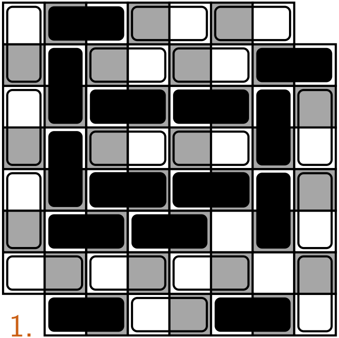


Easy to refute based on the following two observations:

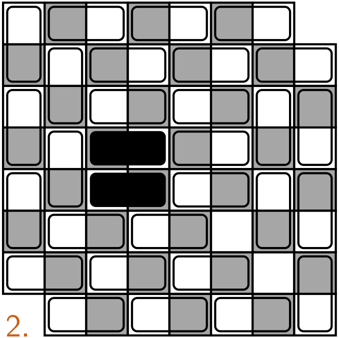
- There are more white squares than black squares; and
- A domino covers exactly one white and one black square.

Without Loss of Satisfaction

One of the crucial techniques in SAT solvers is to generalize a conflicting state and use it to constrain the problem.



1.



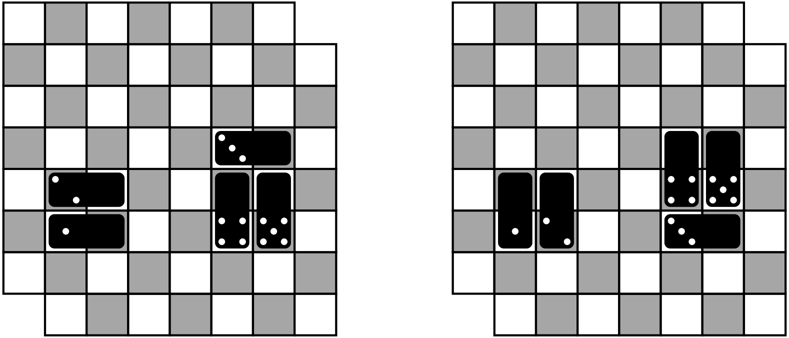
2.

The used proof system can have a big impact on the size:

1. Resolution can only reduce the 30 dominos to 14 (left); and
2. "Without loss of satisfaction" can reduce them to 2 (right).

Mutilated Chessboards: An alternative proof

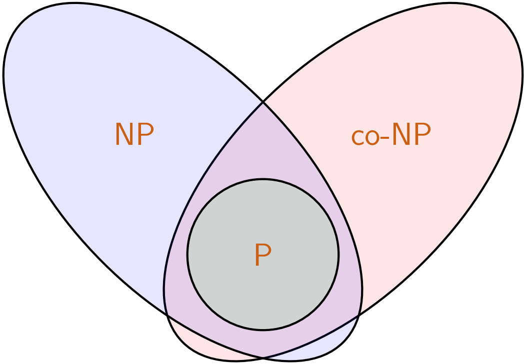
Satisfaction-Driven Clause Learning (SDCL) is a new solving paradigm that finds proofs in the PR proof system [HVC'17]



SDCL can detect that the above two patterns can be blocked

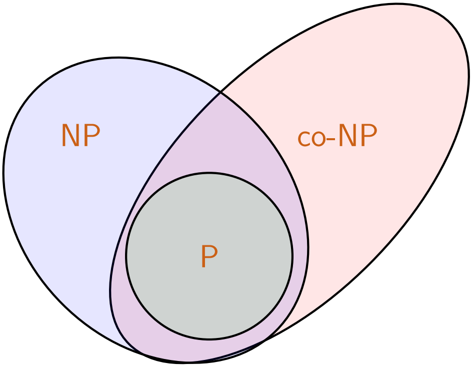
- This reduces the number of explored states exponentially
- We produced SPR proofs that are linear in the formula size

Complexity Classes Closer in Practice



Many NP-complete problems: in NP, but not P (unless P=NP);

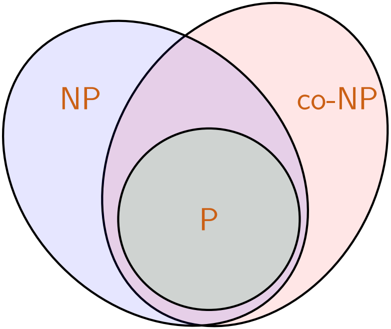
Complexity Classes Closer in Practice



Many NP-complete problems: in NP, but not P (unless P=NP);

yet many NP-complete problem instances easy in practice; and

Complexity Classes Closer in Practice

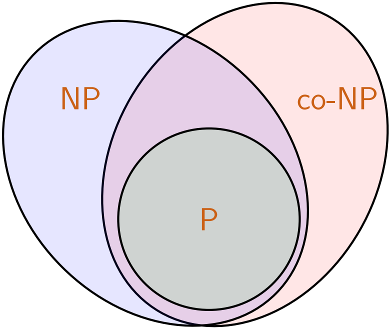


Many NP-complete problems: in NP, but not P (unless P=NP);

yet many NP-complete problem instances easy in practice; and

short proofs exist for many co-NP-complete problem instances.

Complexity Classes Closer in Practice



Many NP-complete problems: in NP, but not P (unless P=NP);

yet many NP-complete problem instances easy in practice; and

short proofs exist for many co-NP-complete problem instances.

"co-NP is the new NP!"

Future Work: Arbitrarily Complex Solvers

Verifying efficient automated reasoning tools is a daunting task:

- Tools are constantly modified and improved; and
- Even top-pier and "experimentally correct" solvers turned out to be buggy. [Järvisalo, Heule, Biere '12]

Verified checkers of certificates in strong proof systems:

- Don't worry about correctness or completeness of tools;
- Facilitates making tools more complex and efficient; while
- Full confidence in results. [Heule, Hunt, Kaufmann, Wetzler '17]

ACL2

Isabelle

Formally verified checkers now also used in industry