max planck institut
informatik

Why Automated Reasoning Procedures Work From Propositional SAT to More Expressive Logics

Christoph Weidenbach

Max Planck Institute for Informatics

June 25, 2016

Outline

CDCL as an Abstract Rewrite System

"classical" CDCL see [MSS96, MMZZM01, NOT06, BHMW09]

$$N = \{P \vee Q,\ \neg P \vee Q,\ P \vee \neg Q,\ \neg P \vee \neg Q\}$$

$$\begin{aligned}
&\Rightarrow_{\mathsf{CDCL}} ([], \emptyset, 0, \top)\\
&\Rightarrow_{\mathsf{CDCL}} ([P^1], \emptyset, 1, \top)\\
&\Rightarrow_{\mathsf{CDCL}} ([P^1\, Q^{\neg P \vee Q}], \emptyset, 1, \top)\\
&\Rightarrow_{\mathsf{CDCL}} ([P^1\, Q^{\neg P \vee Q}], \emptyset, 1, \neg P \vee \neg Q)\\
&\Rightarrow_{\mathsf{CDCL}} ([P^1, \emptyset, 1, \neg P)\\
&\Rightarrow_{\mathsf{CDCL}} ([\neg P], \{\neg P\}, 0, \top)\\
&\Rightarrow_{\mathsf{CDCL}} ([\neg P, Q^{P \vee Q}], \{\neg P\}, 0, \top)\\
&\Rightarrow_{\mathsf{CDCL}} ([\neg P, Q^{P \vee Q}], \{\neg P\}, 0, P \vee \neg Q)\\
&\Rightarrow_{\mathsf{CDCL}} ([\neg P], \{\neg P\}, 0, P)\\
&\Rightarrow_{\mathsf{CDCL}} ([], \{\neg P,\ \bot\}, 0, \bot)
\end{aligned}$$

CDCL States

$(\epsilon; \emptyset; 0; \top)$	is the start state for some clause set N
$(M; U; k; \top)$	is a final state, if $M \models N$ and all literals from N are defined in M
$(M; U; k; \bot)$	is a final state, where N has no model
$(M; U; k; \top)$	is an intermediate model search state if $M \not\models N$ or not all literals from N are defined in M
$(M; U; k; D)$	is a backtracking state if $D \notin \{\top, \bot\}$

CDCL Rules I

Model Extension Rules

Propagate

$(M; U; k; \top) \Rightarrow_{\text{CDCL}} (ML^{C \vee L}; U; k; \top)$

provided $C \vee L \in (N \cup U)$, $M \models \neg C$, and L is undefined in M

Decide

$(M; U; k; \top) \Rightarrow_{\text{CDCL}} (ML^{k+1}; U; k+1; \top)$

provided L is undefined in M

Conflict

$(M; U; k; \top) \Rightarrow_{\text{CDCL}} (M; U; k; D)$

provided $D \in (N \cup U)$ and $M \models \neg D$

CDCL Rules II

Backtracking Rules

Skip

$(ML^{C \vee L}; U; k; D) \Rightarrow_{\text{CDCL}} (M; U; k; D)$

provided $D \notin \{\top, \bot\}$ and $\text{comp}(L)$ does not occur in D

Resolve

$(ML^{C \vee L}; U; k; D \vee \text{comp}(L)) \Rightarrow_{\text{CDCL}} (M; U; k; D \vee C)$

provided D is of level k

Backtrack

$(M_1 K^{i+1} M_2; U; k; D \vee L) \Rightarrow_{\text{CDCL}} (M_1 L^{D \vee L}; U \cup \{D \vee L\}; i; \top)$

provided L is of level k and D is of level i.

CDCL again

$$N = \{\neg P \vee Q,\ \neg Q \vee R \vee S,\ \neg Q \vee R \vee U,\ \neg Q \vee R \vee V,\ \neg V \vee \neg S \ldots\}$$

$$\Rightarrow_{\text{CDCL}} ([], \emptyset, 0, \top)$$

$$\Rightarrow^{\text{Decide}}_{\text{CDCL}} ([P^1], \emptyset, 1, \top)$$

$$\Rightarrow^{\text{Propagate}}_{\text{CDCL}} ([P^1\, Q^{\neg P \vee Q}], \emptyset, 1, \top)$$

$$\Rightarrow^{\text{Decide}}_{\text{CDCL}} ([P^1\, Q^{\neg P \vee Q}\, T^2], \emptyset, 2, \top)$$

$$\Rightarrow^{\text{Decide}}_{\text{CDCL}} ([P^1\, Q^{\neg P \vee Q}\, T^2 \neg R^3], \emptyset, 3, \top)$$

$$\Rightarrow^{\text{Propagate}}_{\text{CDCL}} ([P^1\, Q^{\neg P \vee Q}\, T^2 \neg R^3\, S^{\neg Q \vee R \vee S}], \emptyset, 3, \top)$$

$$\Rightarrow^{\text{Propagate}}_{\text{CDCL}} ([P^1\, Q^{\neg P \vee Q}\, T^2 \neg R^3\, S^{\neg Q \vee R \vee S}\, U^{\neg Q \vee R \vee U}], \emptyset, 3, \top)$$

$$\Rightarrow^{\text{Propagate}}_{\text{CDCL}} ([P^1\, Q^{\neg P \vee Q}\, T^2 \neg R^3\, S^{\neg Q \vee R \vee S}\, U^{\neg Q \vee R \vee U}\, V^{\neg Q \vee R \vee V}], \emptyset, 3, \top)$$

$$\Rightarrow^{\text{Conflict}}_{\text{CDCL}} ([P^1 \ldots R^3\, S^{\neg Q \vee R \vee S}\, U^{\neg Q \vee R \vee U}\, V^{\neg Q \vee R \vee V}], \emptyset, 3, \neg V \vee \neg S)$$

CDCL again continued

$$N = \{\neg P \vee Q,\ \neg Q \vee R \vee S,\ \neg Q \vee R \vee U,\ \neg Q \vee R \vee V,\ \neg V \vee \neg S \ldots\}$$

$$\Rightarrow^{\text{Conflict}}_{\text{CDCL}} ([P^1 \ldots R^3 S^{\neg Q \vee R \vee S} U^{\neg Q \vee R \vee U} V^{\neg Q \vee R \vee V}], \emptyset, 3, \neg V \vee \neg S)$$

$$\Rightarrow^{\text{Resolve}}_{\text{CDCL}} ([P^1 Q^{\neg P \vee Q} T^2 \neg R^3 S^{\neg Q \vee R \vee S} U^{\neg Q \vee R \vee U}], \emptyset, 3, \neg Q \vee R \vee \neg S)$$

$$\Rightarrow^{\text{Skip}}_{\text{CDCL}} ([P^1 Q^{\neg P \vee Q} T^2 \neg R^3 S^{\neg Q \vee R \vee S}], \emptyset, 3, \neg Q \vee R \vee \neg S)$$

$$\Rightarrow^{\text{Resolve}}_{\text{CDCL}} ([P^1 Q^{\neg P \vee Q} T^2 \neg R^3], \emptyset, 3, \neg Q \vee R)$$

$$\Rightarrow^{\text{Backtrack}}_{\text{CDCL}} ([P^1 Q^{\neg P \vee Q} R^{\neg Q \vee R}], \{\neg Q \vee R\}, 1, \top)$$

CDCL Properties [BFW2016]

Theorem (CDCL Terminates)

CDCL terminates reasonably.
Proof: show for $S_1 \Rightarrow_{CDCL} S_2$ *that* $\mu(S_1) > \mu(S_2)$ *separately for each rule, where* μ*'s range are the naturals.*

Theorem (CDCL Strong Completeness)

For any model M *of* N*, there is a reasonable sequence of rule applications generating* $(M; U; k; \top)$ *as a final state.*
Proof: inductive argument on the size of M

Theorem (CDCL Soundness)

If $(\epsilon; \emptyset; 0; \top) \Rightarrow^{\downarrow}_{CDCL} (M; U; k; \top)$ *then* $M \models N$.
Proof: all literals defined, rule Conflict not applicable

CDCL & Redundancy [W2015]

Definition (Redundancy)

Let $\prec$ be a total ordering on the propositional variables. Then a clause $C \in N$ is redundant if $N^{\prec C} \models C$.

Theorem (CDCL & Redundancy)

A learned CDCL clause is not redundant.

If $(\epsilon; \emptyset; 0; \top) \Rightarrow^*_{\mathsf{CDCL}} ([L_1, \ldots, L_n]; U; k; C)$ then $\mathsf{atom}(L_1) \prec \ldots \prec \mathsf{atom}(L_n)$ and $N^{\prec C} \not\models C$

Corollary (CDCL Terminates)

CDCL terminates because it does not learn redundant clauses.

Delete Redundant Clauses

$N = \{P \vee Q,\ R \vee \neg P, \ldots\}$

$\Rightarrow^*_{\mathrm{CDCL}} ([\ldots, \neg R, \neg Q, \neg P, \ldots], \ldots)$ ordering $R \prec Q \prec P$

learns $Q \vee R$ that it not redundant

restart $\Rightarrow^*_{\mathrm{CDCL}} ([\ldots, \neg P, \neg Q, \neg R, \ldots], \ldots)$ ordering $P \prec Q \prec R$

now $Q \vee R$ becomes redundant

- flexible ordering not compatible with abstract redundancy
- abstract redundancy is not compatible with CDCL reasoning
- CDCL supports only weaker notions of redundancy: subset order
- still: at any point in time, learned CDCL clauses are not redundant

Fixed Ordering Problem Solving

$$\begin{aligned} N \quad = \{ & \neg P_1 \vee P_2 \vee \ldots \vee P_n,\ P_1 \vee P_2 \vee \ldots \vee P_n \\ & \neg P_2 \vee P'_2,\ \neg P_2 \vee \neg P'_2, \\ & \ldots \\ & \neg P_n \vee P'_n,\ \neg P_n \vee \neg P'_n \} \end{aligned}$$

$O(n)$ refutation: $P_n \prec P_{n-1} \ldots \prec P_1 \prec P'_n \prec P'_{n-1} \ldots \prec P'_2$

$O(2^n)$ refutation: $P_n \succ P_{n-1} \ldots \succ P_1 \succ P'_n \succ P'_{n-1} \ldots \succ P'_2$

$$\begin{aligned} N \quad = \{ & Q \vee \neg P_1 \vee P_2 \vee \ldots \vee P_n,\ Q \vee P_1 \vee P_2 \vee \ldots \vee P_n \\ & Q \vee \neg P_2 \vee P'_2,\ Q \vee \neg P_2 \vee \neg P'_2, \\ & \ldots \\ & Q \vee \neg P_n \vee P'_n,\ Q \vee \neg P_n \vee \neg P'_n \} \end{aligned}$$

Consider $N \cup N\{P_i \mapsto P'_i, P'_i \mapsto P_i, Q \mapsto \neg Q\}$

CDCL Summary

- CDCL only learns non-redundant clauses
- removing redundant clauses bashes CDCL completeness
- weaker notions, e.g., based on subset order are fine
- if no good fixed ordering is known, learn it dynamically
- dynamic ordering learning is superior to strong redundancy

The Bernays-Schönfinkel Fragment [BS28]

- first-order clause logic without non-constant function symbols
- decidable, finite model property, NEXPTIME complete
- $R(x, y) \wedge R(y, z) \rightarrow R(x, z)$
- $\mathrm{Bird}(\mathrm{tweety}), \mathrm{Bird}(x) \rightarrow \mathrm{Animal}(x)$
- can be reduced to propositional logic with exponential overhead
- dedicated calculi, decision procedures [GK03, BFT06, PMB10, HW13]
- here Non Redundant Clause Learning (NRCL) [AW15]

NRCL (Simplified)

$$N = \{\neg P(c),\ \neg P(x) \vee \neg P(y) \vee Q(x,y), \neg P(y) \vee \neg Q(a,y), \ldots\}$$

$$\begin{aligned}
&\Rightarrow_{\mathsf{NRCL}} ([], \emptyset, 0, \top)\\
&\Rightarrow^{\mathsf{Propagate}}_{\mathsf{NRCL}} ([\neg P(c)], \emptyset, 0, \top)\\
&\Rightarrow^{\mathsf{Decide}}_{\mathsf{NRCL}} ([\neg P(c), (P(x); x \neq c)^1], \emptyset, 1, \top)\\
&\Rightarrow^{\mathsf{Propagate}}_{\mathsf{NRCL}} ([\neg P(c), (P(x); x \neq c)^1, (\neg Q(a,y); y \neq c)], \emptyset, 1, \top)\\
&\Rightarrow^{\mathsf{Conflict}}_{\mathsf{NRCL}} ([\neg P(c), (P(x); x \neq c)^1, (\neg Q(a,y); y \neq c)], \emptyset, 1,\\
&\qquad\qquad (\neg P(x) \vee \neg P(y) \vee Q(x,y); \{x \mapsto a\}; y \neq c))\\
&\Rightarrow^{\mathsf{Resolve}}_{\mathsf{NRCL}} ([\neg P(c), (P(x); x \neq c)^1], \emptyset, 1, (\neg P(a) \vee \neg P(y); y \neq c))\\
&\Rightarrow^{\mathsf{Backtrack}}_{\mathsf{NRCL}} ([\neg P(c), \neg P(a)], \{\neg P(a)\}, 0, \top)
\end{aligned}$$

NRCL one Rule

CDCL Backtrack

Backtrack

$(M_1 K^{i+1} M_2; U; k; D \vee L) \Rightarrow_{\text{CDCL}} (M_1 L^{D \vee L}; U \cup \{D \vee L\}; i; \top)$

provided L is of level k and D is of level i.

NRCL Backtrack

Backtrack

$(M_1 M_2; U; k; (C; \sigma; \pi)) \Rightarrow_{\text{NRCL}} (M_1; U \cup \{C\}; i; \top)$

if $0 \leq i \leq k$, $i = \text{lvl}(M_1)$, and if: (1) $k = 0$, and $C = \bot$; or
(2) $k > 0$, $(C\sigma; \pi)$ is assertive, and C has no false instance under M_1; or
(3) $k > 0$, the right-most element of M_2 is the top-level decision, $(C\sigma; \pi)$ is not assertive, *Factorize* cannot be applied, and C has no false instance under M_1.

Blocked Decisions

$$N = \{\neg P(x) \vee \neg P(y) \vee Q(x, y), \ldots\}$$

$\Rightarrow_{\text{NRCL}} \ ([], \emptyset, 0, \top)$

$\Rightarrow^{\text{Decide}}_{\text{NRCL}} \ ([\neg Q(x, y)], \emptyset, 0, \top)$

then a decision $P(x)$ is blocked as it would falsify the above clause

it can be shown that there is always a non-blocked decision

NRCL Properties

Theorem (NRCL Terminates)

NRCL terminates reasonably.
Proof: show for $S_1 \Rightarrow_{NRCL} S_2$ *that* $\mu(S_1) > \mu(S_2)$ *separately for each rule, where* μ*'s range are the naturals.*

Theorem (NRCL Strong Completeness)

For any model M *of* N*, there is a reasonable sequence of rule applications generating* $(M; U; k; \top)$ *as a final state.*
Proof: inductive argument on the size of M

Theorem (NRCL Soundness)

If $(\epsilon; \emptyset; 0; \top) \Rightarrow^{\downarrow}_{NRCL} (M; U; k; \top)$ *then* $M \models N$.
Proof: all literals defined, Conflict not applicable

NRCL & Redundancy (Simplified)

Definition (Redundancy)

Let $\prec$ be a total ordering on first-order ground atoms. Then a clause $C \in N$ is redundant if $N^{\prec C} \models C$ modulo ground instantiation.

Theorem (NRCL & Redundancy)

A learned NRCL clause is not redundant.

If $(\epsilon; \emptyset; 0; \top) \Rightarrow^*_{\mathsf{NRCL}} ([L_1, \ldots, L_n]; U; k; C)$ then $\mathsf{atom}(L_1)\sigma_1 \prec \ldots \prec \mathsf{atom}(L_n)\sigma_n$ and $N^{\prec C} \not\models C$

Corollary (NRCL Terminates)

NRCL terminates because it does not learn redundant clauses.

Comparison NRCL & CDCL

	CDCL	NRCL
Satisfiability	NP-complete	NEXPTIME-complete
Model Size	$O(n)$	$O(k^i)$
Propagation	$O(\|C\| + \|M\|)$	NP-complete
Falsehood	$O(\|C\| + \|M\|)$	NP-complete
Theory	15 pages	45 pages

potential gain	Propositional	Bernays-Schönfinkel
Problem Size	$O(2^n)$	$O(n)$
Model Size	$O(2^n)$	$O(k^i)$
Proof Size	$O(2^m)$	$O(m)$

BS and QBF [SLB12]

- $\exists A \forall B, C \exists D\, [(A \vee \neg D \vee C) \ \wedge \ (A \vee \neg B \vee D)]$
- $\exists x \forall y, z \exists u\, [(A(x) \vee \neg D(u) \vee C(z)) \ \wedge \ (A(x) \vee \neg B(y) \vee D(u))]$
- $\forall y, z\, [(A(a) \vee \neg D(f(y,z)) \vee C(z)) \ \wedge \ (A(a) \vee \neg B(y) \vee D(f(y,z)))]$
- $\forall y, z\, [(A(a) \vee \neg D_f(y,z) \vee C(z)) \ \wedge \ (A(a) \vee \neg B(y) \vee D_f(y,z))]$
- $\forall y, z\, [(A \vee \neg D_f(y,z) \vee C(z)) \ \wedge \ (A \vee \neg B(y) \vee D_f(y,z))]$
- 2-element domain, simplifying greatly NRCL constraint handling

BS Summary

- CDCL & Superposition ideas can be lifter to BS
- roughly same structure, but far more complicated
- non-redundant clause learning
- naturally get calculus for QBF
- potential gains but less robustness

LIA Constraint Solving

$$\begin{aligned} 5x - 3y &\leq 7 \\ 2z + 3x - y &= 0 \\ 37x + 11z &> 42 \end{aligned}$$

Find values for x, y, z in $\mathbb{Z}$ such that the above disequations hold.

How does it work in $\mathbb{R}$?

LRA Constraint Solving

$$\begin{aligned} y &\geq 4x \\ 3x &\geq z \\ 8z &\geq 2y \end{aligned}$$

$3y \geq 4z \geq y$

$y = 2,\ z = 1,\ \frac{1}{2} \geq x \geq \frac{1}{3}$

LIA Constraint Solving

$$\begin{aligned} y &\geq 4x \\ 3x &\geq z \end{aligned}$$

$3y \geq 4z$ in $\mathbb{R}$ with $x = \frac{z}{3}$ but $x = \lceil \frac{z}{3} \rceil$ in $\mathbb{Z}$

if $z \in \mathbb{Z}$ then $\lceil \frac{z}{3} \rceil \in \{\frac{z}{3}, \frac{z+1}{3}, \frac{z+2}{3}\}$

$$\bigvee_{k=0}^{2} (3|z + k \wedge 3y \geq 4z + 4k \wedge z + k \geq z) \text{ in } \mathbb{Z}$$

LIA Constraint Solving Complexity: NP-hard

Papadimitriou [P81]

A LIA constraint solving problem has a solution in $-n(ma)^{2m+1} \leq x \leq n(ma)^{2m+1}$.

Example

$$\begin{aligned} 4x - 4y &\geq 1 \\ 4x - 4y &\leq 3 \end{aligned}$$

$-65536 \leq x, y \leq 65536$

Today LIA Constraint Solving

Branch & Bound [G12, BW16]

- do fast tests getting an integer solution
- solve the relaxation in $\mathbb{R}$
- guess the found solution in to $\mathbb{R}$ an integer solution
- select a solution $c \in \mathbb{R}$ of x and branch on $x \geq \lceil c \rceil$ and $x \leq \lfloor c \rfloor$

LIA Partial Model Building

Consider: $1 + y \leq x \quad x \leq y$

Decide: $x \geq 5 \quad y \geq 5$

Propagate: $x \geq 6$

Propagate: $y \geq 6$

Propagate: $\ldots$

Open Problem: find propagation terminating in number of variables.

LIA Learning

Consider: $3x \leq y \quad z \leq 3x$

Partial Model: $y \geq 1,\ y \leq 1,\ z \geq 1,\ z \leq 1$

Propagate: $x \leq 0$ by $3x \leq y$

Conflict: $z \leq 3x$

Resolve: $z \leq y$ is satisfied.

Open Problem: find powerful learning mechanism.

Summary

- non-redundant clause learning works for propositional SAT: CDCL
- non-redundant clause learning works for BS SAT: NRCL
- non-redundant clause learning works for QBF
- there are more positive cases: NRA [MP13, BK15]
- we do not have a satisfactory result for LIA [JM13, BSW15]

The End

References CDCL

Christoph Weidenbach.
Automated reasoning building blocks.
In Roland Meyer, André Platzer, and Heike Wehrheim, editors, Correct System Design, volume 9360 of LNCS, pages 172–188. Springer, 2015.

Jasmin Christian Blanchette, Mathias Fleury, and Christoph Weidenbach.
A Verified SAT Solver Framework with Learn, Forget, Restart, and Incrementality.
In , Nicola Olivetti, and Ashish Tiwari, Proceedings IJCAR 2016.

Robert Nieuwenhuis, Albert Oliveras, and Cesare Tinelli.
Solving sat and sat modulo theories: From an abstract davis–putnam–logemann–loveland procedure to DPLL(T).
Journal of the ACM, 2006.

References CDCL


João P. Marques Silva and Karem A. Sakallah.
Grasp - a new search algorithm for satisfiability.
In ICCAD-1996.


Armin Biere, Marijn Heule, Hans van Maaren, and Toby Walsh, editors.
Handbook of Satisfiability, IOS Press, 2009.


Matthew W. Moskewicz, Conor F. Madigan, Ying Zhao, Lintao Zhang, and Sharad Malik.
Chaff: Engineering an efficient sat solver.
In DAC-2001.

References BS

Martina Seidl, Florian Lonsing, and Armin Biere.
qbf2epr: A tool for generating EPR formulas from QBF.
In Proceedings PAAR-2012.

Gabor Alagi and Christoph Weidenbach.
NRCL - a model building approach to the Bernays-Schönfinkel fragment.
In Carsten Lutz and Silvio Ranise, Proceedings FroCoS 2015.

Ruzica Piskac, Leonardo Mendonça de Moura, and Nikolaj Bjørner.
Deciding effectively propositional logic using DPLL and substitution sets.
Journal of Automated Reasoning, 2010.

References BS


Harald Ganzinger and Konstantin Korovin.
New directions in instatiation–based theorem proving.
In Samson Abramsky, LICS-2003.


Peter Baumgartner, Alexander Fuchs, and Cesare Tinelli.
Lemma learning in the model evolution calculus.
In LPAR-2006.


Thomas Hillenbrand and Christoph Weidenbach.
Superposition for bounded domains.
In Maria Paola Bonacina and Mark Stickel, editors, *McCune Festschrift*, 2013.

References BS


Paul Bernays and Moses Schönfinkel.
Zum Entscheidungsproblem der mathematischen Logik.
Mathematische Annalen, 1928.

References LIA

Christos H. Papadimitriou.
On the complexity of integer programming.
Journal of the ACM, 1981.

Dejan Jovanovic and Leonardo Mendonça de Moura.
Cutting to the chase - solving linear integer arithmetic.
Journal of Automated Reasoning, 2013.

Martin Bromberger, Thomas Sturm, and Christoph Weidenbach.

Linear integer arithmetic revisited.
In Amy P. Felty and Aart Middeldorp, editors, CADE-2015.

References LIA


Alberto Griggio.
A practical approach to satisability modulo linear integer arithmetic.
Journal on Satisfiability, 2012.


Martin Bromberger and Christoph Weidenbach.
Fast cube tests for lia constraint solving.
In IJCAR-2016.

References NRA


Leonardo Mendonça de Moura and Grant Olney Passmore.
Computation in real closed infinitesimal and transcendental extensions of the rationals.
In Maria Paola Bonacina, editor, CADE-2013.


Christopher W. Brown and Marek Kosta.
Constructing a single cell in cylindrical algebraic decomposition.
Journal of Symbolic Computation, 2015.