max planck institut informatik

Automated Reasoning Building Blocks

Part I

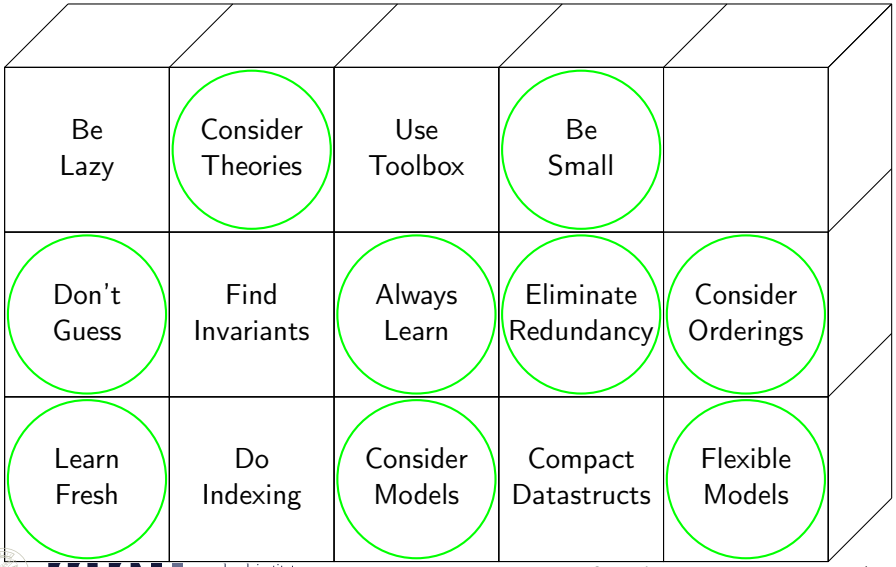
Christoph Weidenbach

Max Planck Institute for Informatics

September 21, 2015

Outline

Automated Reasoning Building Blocks



Propositional Clause Logic

Syntax

- Clauses have the form $P \vee \neg R \vee Q \vee Q$ where $P, Q, R \in \Sigma$
- Clauses denoted by C, D, empty clause denoted by $\bot$
- Clause sets N, M are interpreted as conjunctions of clauses

Semantics

- (Partial) Valuations $\mathcal{A} : \Sigma \to \{0, 1\}$
- Clause Set N satisfiable if $\mathcal{A}(N) = 1$ for some $\mathcal{A}$, $\mathcal{A} \models N$
- Clause Set N unsatisfiable if $\mathcal{A}(N) = 0$ for all $\mathcal{A}$
- Clause Set N valid if $\mathcal{A}(N) = 1$ for all $\mathcal{A}$, $\models N$

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

Analytic Models

Beth 55, Smullyan 68, Fitting 90

Tableau Procedure: Close Branch & Backtrack, Split Disjunction

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

$\Rightarrow_{\mathsf{TAB}}\ ([], \top)$
$\Rightarrow_{\mathsf{TAB}}\ ([P^{P \vee Q}], \top)$
$\Rightarrow_{\mathsf{TAB}}\ ([P^{P \vee Q}\, R^{R \vee S}], \top)$
$\Rightarrow_{\mathsf{TAB}}\ ([P^{P \vee Q}\, R^{R \vee S}\, \neg P^{\neg P \vee Q}], \bot)$
$\Rightarrow_{\mathsf{TAB}}\ ([P^{P \vee Q}\, R^{R \vee S}\, Q], \top)$
$\Rightarrow_{\mathsf{TAB}}\ ([P^{P \vee Q}\, R^{R \vee S}\, Q\, \neg Q], \bot)$
$\Rightarrow_{\mathsf{TAB}}\ ([P^{P \vee Q}\, S], \top) \ldots$

$\Rightarrow_{\mathsf{TAB}}\ ([Q], \top)$
$\Rightarrow_{\mathsf{TAB}}\ ([Q\, \neg Q], \bot)$
$\Rightarrow_{\mathsf{TAB}}\ ([], \bot)$

Only Guessing &
No Model
Consideration

Consider Models

Davis & Logman & Loveland 1962

DPLL Procedure: Find Conflict & Backtrack, Propagate, Guess

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

$\Rightarrow_{\mathsf{DPLL}}\ ([], \top)$
$\Rightarrow_{\mathsf{DPLL}}\ ([\neg Q^{\neg Q}], \top)$
$\Rightarrow_{\mathsf{DPLL}}\ ([\neg Q^{\neg Q} P^{P \vee Q}], \top)$
$\Rightarrow_{\mathsf{DPLL}}\ ([\neg Q^{\neg Q} P^{P \vee Q}], \neg P \vee Q)$
$\Rightarrow_{\mathsf{DPLL}}\ ([], \bot)$

Propagation &
No Guessing

DPLL Minimal Proof Length

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

$\Rightarrow_{\mathsf{DPLL}}\ ([], \top)$
$\Rightarrow_{\mathsf{DPLL}}\ ([P^1], \top)$
$\Rightarrow_{\mathsf{DPLL}}\ ([P^1 Q^{\neg P \vee Q}], \top)$
$\Rightarrow_{\mathsf{DPLL}}\ ([P^1 Q^{\neg P \vee Q}], \neg P \vee \neg Q)$
$\Rightarrow_{\mathsf{DPLL}}\ ([\neg P], \top)$
$\Rightarrow_{\mathsf{DPLL}}\ ([\neg P, Q^{P \vee Q}], \top)$
$\Rightarrow_{\mathsf{DPLL}}\ ([\neg P, Q^{P \vee Q}], P \vee \neg Q)$
$\Rightarrow_{\mathsf{DPLL}}\ ([], \bot)$

No Learning
$O(2^n)$

Always Learn

Robinson 1965

Resolution: $C \vee P,\ \neg P \vee D \Rightarrow_{\mathsf{RES}} C \vee D$

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

$\neg P \vee Q,\ \neg P \vee \neg Q \Rightarrow_{\mathsf{RES}} \neg P$
$P \vee Q,\ P \vee \neg Q \Rightarrow_{\mathsf{RES}} P$
$P,\ \neg P \Rightarrow_{\mathsf{RES}} \bot$

$O(n)$
No Model
Consideration

Eliminate Redundancy

$N = \{P \vee Q,\ \neg P \vee \neg Q\}$
$P \vee Q,\ \neg P \vee \neg Q \Rightarrow_{\mathrm{RES}} Q \vee \neg Q$
May we eliminate $Q \vee \neg Q$?

$N = \{P \vee Q \vee R,\ \neg P \vee Q, P \vee R\}$
$\neg P \vee Q,\ P \vee R \Rightarrow_{\mathrm{RES}} Q \vee R$
May we eliminate $P \vee Q \vee R$?

Number of clauses generated in a typical resolution run:
42, 420, 42000, 42000000, ...

Redundancy: A First-Class Citizen

Boyer 1971

Lock Resolution: $C \vee P_k, \neg P_l \vee D \Rightarrow_{\mathrm{LRES}} C \vee D$
if k, l are maximal indexes, respectively

$N = \{P_5 \vee Q_4,\ \neg P_4 \vee Q_5,\ P_4 \vee \neg Q_5,\ \neg P_5 \vee \neg Q_4\}$
$\Rightarrow_{\mathrm{LRES}} Q_4 \vee \neg Q_4$
$\Rightarrow_{\mathrm{LRES}} \neg P_4 \vee P_4$

Redundancy needs to be considered in the context of a calculus.

Learn Fresh, Eliminate Redundancy

Bachmair & Ganzinger 1990

Superposition: $C \vee P, \neg P \vee D \Rightarrow_{\mathrm{LRES}} C \vee D$
if P, $\neg P$ are maximal in their respective clauses

Ordering

- $\prec$ is a total strict ordering on Σ: $P \prec Q$
- $\prec$ on literals: $P \prec \neg P \prec Q \prec \neg Q$
- $\prec$ on clauses: multiset extension: $\{P, Q\} \prec \{P, Q, Q\} \prec \{\neg Q\}$
- $N^{\prec C} = \{D \in N \mid D \prec C\}$

Superposition Redundancy

Definition (Redundancy)

A clause C is *redundant* with respect to a clause set N if $N^{\prec C} \models C$.

$P \prec Q \prec R$
$N = \{P \vee Q,\ R \vee \neg P\}$ clauses $P \vee Q \vee \neg R,\ Q \vee R$ redundant

Superposition Static Model $N_{\mathcal{I}}$

$$
\begin{aligned}
N_C &:= \textstyle\bigcup_{D \prec C} \delta_D \\
\delta_D &:= \begin{cases} \{P\} & \text{if } D = D' \vee P, P \text{ strictly maximal } N_D \not\models D \\ \emptyset & \text{otherwise} \end{cases} \\
N_{\mathcal{I}} &:= \textstyle\bigcup_{C \in N} \delta_C
\end{aligned}
$$

Superposition Results

Theorem (Completeness, Models, Redundancy)

If all superposition inferences in N up to redundancy are performed and $\bot \notin N$ then N is satisfiable and $N_{\mathcal{I}} \models N$.
It is sufficient to consider inferences between a minimal false clause $\neg P \vee C$, $N_{\mathcal{I}} \not\models \neg P \vee C$ and its productive counterpart $P \vee D$.
The result $C \vee D$ of the superposition inference is not redundant.

Model Properties

- fixed by ordering: $P \vee Q$, $P \vee \neg Q$
- minimal: $P \vee Q$, $\neg P \vee R$, where $R \prec Q \prec P$ then $N_{\mathcal{I}} = \{P\}$

Static Ordering & Model

Always Learn Fresh, Flexible Models

Silva, Sakallah, Bayardo, Schrag, Et Al 2000-

CDCL: Find Conflict & Backtrack & Learn, Propagate, Guess

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

$\Rightarrow_{\text{CDCL}}$ $([], \emptyset, \top)$
$\Rightarrow_{\text{CDCL}}$ $([P^1], \emptyset, \top)$
$\Rightarrow_{\text{CDCL}}$ $([P^1 Q^{\neg P \vee Q}], \emptyset, \top)$
$\Rightarrow_{\text{CDCL}}$ $([P^1 Q^{\neg P \vee Q}], \emptyset, \neg P \vee \neg Q)$
$\Rightarrow_{\text{CDCL}}$ $([\neg P], \{\neg P\}, \top)$
$\Rightarrow_{\text{CDCL}}$ $([\neg P, Q^{P \vee Q}], \{\neg P\}, \top)$
$\Rightarrow_{\text{CDCL}}$ $([\neg P, Q^{P \vee Q}], \{\neg P\}, P \vee \neg Q)$
$\Rightarrow_{\text{CDCL}}$ $([], \{\neg P, \bot\}, \bot)$

$O(n)$
No Redundancy

Consider Theories

while Program Analysis

1	$n = 0$;	$\neg P_1(n,x,y) \vee P_2(0,x,y)$
2	while $(x > 0)$ {	$\neg x > 0 \vee \neg P_2(n,x,y) \vee P_3(n,x,y)$
		$\neg x \leq 0 \vee \neg P_2(n,x,y) \vee P_6(n,x,y)$
3	$n = n + y$;	$\neg P_3(n,x,y) \vee P_4(n+y,x,y)$
4	$x = x - 1$;	$\neg P_4(n,x,y) \vee P_5(n,x-1,y)$
5	}	$\neg P_5(n,x,y) \vee P_2(n,x,y)$
6	return n;	

Axiomatize Theories

Program Formalization

$\neg x > 0 \vee \neg P_2(n,x,y) \vee P_3(n,x,y)$
$\neg x \leq 0 \vee \neg P_2(n,x,y) \vee P_6(n,x,y)$
$\neg P_3(n,x,y) \vee P_4(n+y,x,y)$
$\neg P_4(n,x,y) \vee P_5(n,x-1,y)$

Theory Axioms

$x = 0 \vee x > 0 \wedge s(x) > x$
$s(x) + y = s(x+y) \wedge 0 + y = y \wedge x + y = y + x$
$\ldots$

Incomplete & No Decision Procedure

Combine Theories

Program Formalization

$\neg x > 0 \vee \neg P_2(n,x,y) \vee P_3(n,x,y)$
$\neg x \leq 0 \vee \neg P_2(n,x,y) \vee P_6(n,x,y)$
$\neg P_3(n,x,y) \vee P_4(n+y,x,y)$
$\neg P_4(n,x,y) \vee P_5(n,x-1,y)$

Combination

$x > 0 \quad \| \neg P_2(n,x,y) \vee P_3(n,x,y)$
$x \leq 0 \quad \| \neg P_2(n,x,y) \vee P_6(n,x,y)$
$z = n+y \| \neg P_3(n,x,y) \vee P_4(z,x,y)$
$z = x-1 \| \neg P_4(n,x,y) \vee P_5(n,z,y)$

Expressive Logic & Difficult Automation

Summary

Open Questions

- Can redundancy and completeness be combined?
- Can redundancy and ordering restrictions be combined?
- Can model building and inferences be combined?
- How does reasoning in combination of theories work?

Answers

- Superposition [BachmairGanzinger90]
- CDCL [BayardoSchrag96, SilvaSakallah96, NieuwenhuisEtAl06]
- CDCL(T)/SMT [NieuwenhuisEtAl06]
- Hierarchic Superposition [BachmairEtAl94, KruglovW12, FietzkeW12]

Disclaimer

Simplified Presentation

- Many Technical Details Omitted
- Almost no Equality
- See References for Further Reading

Goal

Get the Ideas/Intuition

References Tableau

E.W. Beth.
Semantic entailment and formal derivability.
Mededelingen van de Koninklijke Nederlandse Akademie van Wetenschappen, Afdeling Letterkunde, 18(13):309–342, 1955.

Raymond M. Smullyan.
First-Order Logic.
Ergebnisse der Mathematik und ihrer Grenzgebiete. Springer, 1968.

Melvin Fitting.
First-Order Logic and Automated Theorem Proving.
Texts and Monographs in Computer Science. Springer, 1990.

References Superposition

- Leo Bachmair and Harald Ganzinger.
  On restrictions of ordered paramodulation with simplification.
  In CADE-10, LNCS 449, pages 427–441. Springer, 1990.
- R.S. Boyer.
  Locking: A Restriction of Resolution.
  PhD thesis, University of Texas at Austin, August 1971.
- John Alan Robinson.
  A machine-oriented logic based on the resolution principle.
  Journal of the ACM, 12(1):23–41, January 1965.

References CDCL

- Roberto J. Bayardo Jr. and Robert Schrag.
  Using CSP look-back techniques to solve exceptionally hard SAT instances.
  In Eugene C. Freuder, editor, CP 1996, Cambridge, Massachusetts, USA, August 19-22, LNCS 1118, pages 46–60. Springer, 1996.
- João P. Marques Silva and Karem A. Sakallah.
  Grasp - a new search algorithm for satisfiability.
  In ICCAD 1996, pages 220–227. IEEE Press, 1996.
- 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, 53:937–977, November 2006.

References Combination

- Leo Bachmair, Harald Ganzinger, and Uwe Waldmann.
  Refutational theorem proving for hierarchic first-order theories.
  AAECC, 5(3/4):193–212, 1994.
- Evgeny Kruglov and Christoph Weidenbach.
  Superposition decides the first-order logic fragment over ground theories.
  MCS, 6(4):427–456, 2012.
- Arnaud Fietzke and Christoph Weidenbach.
  Superposition as a decision procedure for timed automata.
  MCS, 6(4):409–425, 2012.

Propositional Resolution

Resolution [Robinson65]

Resolution
$(N \uplus \{C \vee P, D \vee \neg P\}) \Rightarrow_{\text{RES}} (N \cup \{C \vee P, D \vee \neg P, C \vee D\})$

Factoring
$(N \uplus \{C \vee L \vee L\}) \Rightarrow_{\text{RES}} (N \cup \{C \vee L \vee L\} \cup \{C \vee L\})$

Theorem (Resolution is Sound and Complete)

N is unsatisfiable iff $N \Rightarrow^*_{RES} N' \cup \{\bot\}$

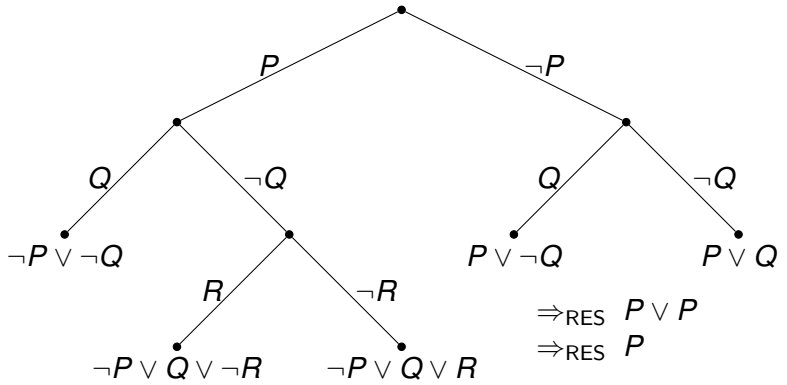
Soundness

Resolution
$(N \uplus \{C \vee P, D \vee \neg P\}) \Rightarrow_{\text{RES}} (N \cup \{C \vee P, D \vee \neg P, C \vee D\})$
if $\mathcal{A}((C \vee P) \wedge (D \vee \neg P)) = 1$ then $\mathcal{A}(C \vee D) = 1$

Factoring
$(N \uplus \{C \vee L \vee L\}) \Rightarrow_{\text{RES}} (N \cup \{C \vee L \vee L\} \cup \{C \vee L\})$
if $\mathcal{A}(C \vee L \vee L) = 1$ then $\mathcal{A}(C \vee L) = 1$

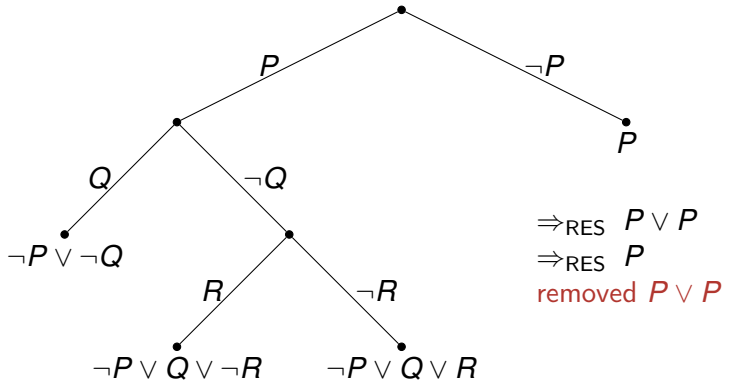
Completeness: Semantic Trees

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



Semantic Tree Redundancy

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



Eliminate Redundancy

Well-Founded Semantic Tree Ordering
A clause C is redundant if $D \subseteq C$ for some D
Any clause C not occurring in any semantic tree is redundant

Subsumption
$(N \uplus \{C, D\}) \Rightarrow_{\text{RES}} (N \cup \{C\})$
provided $C \subset D$

Condensation
$(N \uplus \{C \vee L \vee L\}) \Rightarrow_{\text{RES}} (N \cup \{C \vee L\})$

Tautology Deletion
$(N \uplus \{C \vee P \vee \neg P\}) \Rightarrow_{\text{RES}} (N)$

Can redundancy and completeness be combined?

Termination

$N_0 \Rightarrow_{\mathsf{RES}} N_1 \Rightarrow_{\mathsf{RES}} \ldots$

How Many Different Clause Sets?

at most 3^n different clauses, $n = |\Sigma|$, modulo condensation
at most $2^{(3^n)}$ different clause sets

Examples

$N = \{P \vee C_1, P \vee C_2, \ldots\}$ obviously satisfiable
$N = \{P \vee Q,\ P \vee \neg Q\}$
under $\mathcal{A}(P) = 1$ why consider $P \vee \neg Q$ for resolution?

Can model building and inferences be combined?

Propositional Superposition

Ordering

- $\prec$ is a total strict ordering on Σ: $P \prec Q$
- $\prec$ on literals: $P \prec \neg P \prec Q \prec \neg Q$
- $\prec$ on clauses: multiset extension: $\{P, Q\} \prec \{P, Q, Q\} \prec \{\neg Q\}$
- $N^{\prec C} = \{D \in N \mid D \prec C\}$

Definition (Redundancy)

A clause C is *redundant* with respect to a clause set N if $N^{\prec C} \models C$.

$P \prec Q \prec R$
$N = \{P \vee Q,\ R \vee \neg P\}$ clauses $P \vee Q \vee \neg R,\ Q \vee R$ redundant

Models

Definition (Selection Function)

- sel maps clauses to one of its negative literals or $\bot$
- if $\mathsf{sel}(C) = \neg P$ then $\neg P$ is called *selected* in C
- if $\mathsf{sel}(C) = \bot$ then no literal in C is *selected*

Partial Herbrand Model Construction

$$N_C := \bigcup_{D \prec C} \delta_D$$

$$\delta_D := \begin{cases} \{P\} & \text{if } D = D' \vee P, P \text{ strictly maximal, no literal selected in } D \text{ and } N_D \not\models D \\ \emptyset & \text{otherwise} \end{cases}$$

$$N_{\mathcal{I}} := \bigcup_{C \in N} \delta_C$$

Partial Herbrand Model Properties

$$N_C := \bigcup_{D \prec C} \delta_D$$

$$\delta_D := \begin{cases} \{P\} & \text{if } D = D' \vee P, P \text{ strictly maximal, no literal selected in } D \text{ and } N_D \not\models D \\ \emptyset & \text{otherwise} \end{cases}$$

$$N_{\mathcal{I}} := \bigcup_{C \in N} \delta_C$$

Properties

- $N_{\mathcal{I}}$ is minimal with respect to set inclusion
- if $D \prec C$ and $N_C \models D$ then $N_{\mathcal{I}} \models D$
- if $P \vee P \prec C$ then $\delta_C \neq \{P\}$
- if $\delta_C = \{P\}$ then $N_C \cup \delta_C \models C$

Propositional Superposition

Superposition

Superposition Left

$(N \uplus \{C \vee P, D \vee \neg P\}) \Rightarrow_{\mathsf{SUP}} (N \cup \{C \vee P, D \vee \neg P\} \cup \{C \vee D\})$

where (i) P is strictly maximal in $C \vee P$
(ii) no literal in $C \vee P$ is selected
(iii) $\neg P$ is maximal and no literal selected in $D \vee \neg P$, or
$\neg P$ is selected in $D \vee \neg P$

Factoring

$(N \uplus \{C \vee P \vee P\}) \Rightarrow_{\mathsf{SUP}} (N \cup \{C \vee P \vee P\} \cup \{C \vee P\})$

where (i) P is maximal in $C \vee P \vee P$
(ii) no literal is selected in $C \vee P \vee P$

Saturation and Completeness

Definition (Saturation)

A set N of clauses is called *saturated up to redundancy*, if any clause generated by Superposition Left or Factoring from non-redundant clauses in N is redundant with respect to N or contained in N.

Theorem (Superposition Completeness)

If N is saturated up to redundancy and $\bot \notin N$ then N is satisfiable and $N_{\mathcal{I}} \models N$.

Superposition Completeness Proof

Theorem (Superposition Completeness)

If N is saturated up to redundancy and $\bot \notin N$ then N is satisfiable and $N_{\mathcal{I}} \models N$.

Proof.

By contradiction. I assume:
(i) if $N \Rightarrow_{\mathsf{SUP}} N \cup D$ then $N^{\prec D} \models D$ or $D \in N$, (ii) $\bot \notin N$ and
(iii) $N_{\mathcal{I}} \not\models N$.
Then there is a minimal clause $C \vee L \in N$ such that $N_{\mathcal{I}} \not\models C \vee L$ and L is selected or nothing selected and L maximal. This clause must exist because $\bot \notin N$.
$C \vee L$ is not redundant and by Superposition Left or Factoring we can derive a non-redundant clause that is smaller than $C \vee L$ and false in $N_{\mathcal{I}}$, a contradiction. □

Proof: More Details.

- if $C \vee L$ redundant, then $N_{C \vee L} \models C \vee L$, a contradiction
- L positive, L not strictly maximal, do Factoring, a contradiction
- L positive, L strictly maximal, $L \in N_{\mathcal{I}}$, a contradiction
- L negative, do Superposition Left, a contradiction

□

Completeness Summary

Superposition Completeness

- does inferences with respect to a candidate model $N_{\mathcal{I}}$
- only inferences on false clauses needed: see proof
- supports ordering restrictions: $\prec$
- supports redundancy: $N^C \models C$

Superposition Results

Theorem (Completeness, Models, Redundancy)

If all superposition inferences in N up to redundancy are performed and $\bot \notin N$ then N is satisfiable and $N_{\mathcal{I}} \models N$.
It is sufficient to consider inferences between a minimal false clause $\neg P \vee C$, $N_{\mathcal{I}} \not\models \neg P \vee C$ and its productive counterpart $P \vee D$.
The result $C \vee D$ of the superposition inference is not redundant.

Propositional Specialities

- $N_{\mathcal{I}}$ can be effectively constructed
- $N_{\mathcal{I}} \not\models \neg P \vee C$ easy to decide
- unsatisfiability is co-NP, saturation always terminates

Concrete Redundancy

Subsumption
$(N \uplus \{C, D\}) \Rightarrow_{\mathsf{SUP}} (N \cup \{C\})$
provided $C \subset D$

Condensation
$(N \uplus \{C \vee L \vee L\}) \Rightarrow_{\mathsf{SUP}} (N \cup \{C \vee L\})$

Tautology Deletion
$(N \uplus \{C \vee P \vee \neg P\}) \Rightarrow_{\mathsf{SUP}} (N)$

Subsumption Resolution
$(N \uplus \{C_1 \vee L, C_2 \vee \neg L\}) \Rightarrow_{\mathsf{SUP}} (N \cup \{C_1 \vee L, C_2\})$
where $C_1 \subseteq C_2$

Extension I: Saturation

Example (N saturated vs. $N_{\mathcal{I}} \models N$)

$N = \{P, \neg P \vee \neg Q\}$ with ordering $Q \prec P$.
Then $N_{\mathcal{I}} \models N$ but N is not saturated.

Fix

Always check whether $N_{\mathcal{I}} \models N$.

Extension II: Flexible Model Operator

Example (Inflexible Model Operator)

$N = \{P \vee Q,\ \neg P \vee R\}$ with ordering $R \prec Q \prec P$.
Then $N_{\mathcal{I}} = \{P\}$ and $N_{\mathcal{I}} \not\models N$
But $N_{\mathcal{I}} \cup \{R\} \models N$.

Fix

Use a different model operator.

Flexible Model Operator

H is a decision heuristic: $H\colon \Sigma \to \{0, 1\}$

$$N_P^H := \bigcup_{Q \prec P} \delta_Q^H$$

$$\delta_P^H := \begin{cases} \{P\} & \text{if } (D \vee P) \in N, \text{ with } N_P^H \models \neg D \\ & \text{and } P \text{ strictly maximal, nothing selected} \quad \text{or} \\ & H(P) = 1 \text{ no clause } (D' \vee \neg P) \in N, D' \prec P \\ & \text{such that } N_P^H \models \neg D' \\ \emptyset & \text{otherwise} \end{cases}$$

$$N_\Sigma^H := \bigcup_{P \in \Sigma} \delta_P^H$$

N_Σ^H Properties [W2015]

Theorem (Superposition Completeness)

If N is saturated up to redundancy and $\bot \notin N$ then N is satisfiable and $N_\Sigma^H \models N$.

Example (Flexible Model Operator)

$N = \{P \vee Q,\ \neg P \vee R\}$ with ordering $R \prec Q \prec P$, $H(R) = 1$
Then $N_\Sigma^H = \{P, R\}$ and $N_\Sigma^H \models N$

CDCL States

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

CDCL Rules I

Model Extension Rules

Propagate
$(M; N; U; k; \top) \Rightarrow_{\text{CDCL}} (ML^{C \vee L}; N; U; k; \top)$
provided $C \vee L \in (N \cup U)$, $M \models \neg C$, and L is undefined in M

Decide
$(M; N; U; k; \top) \Rightarrow_{\text{CDCL}} (ML^{k+1}; N; U; k+1; \top)$
provided L is undefined in M

Conflict
$(M; N; U; k; \top) \Rightarrow_{\text{CDCL}} (M; N; U; k; D)$
provided $D \in (N \cup U)$ and $M \models \neg D$

CDCL Rules II

Backtracking Rules

Skip
$(ML^{C \vee L}; N; U; k; D) \Rightarrow_{\text{CDCL}} (M; N; U; k; D)$
provided $D \notin \{\top, \bot\}$ and $\text{comp}(L)$ does not occur in D

Resolve
$(ML^{C \vee L}; N; U; k; D \vee \text{comp}(L)) \Rightarrow_{\text{CDCL}} (M; N; U; k; D \vee C)$
provided D is of level k

Backtrack
$(M_1 K^{i+1} M_2; N; U; k; D \vee L) \Rightarrow_{\text{CDCL}} (M_1 L^{D \vee L}; N; U \cup \{D \vee L\}; i; \top)$
provided L is of level k and D is of level i.

CDCL Properties

Theorem (CDCL Soundness)

CDCL terminates reasonably in two different final states: $(M; N; U; k; \top)$ where $M \models N$ and $(M; N; U; k; \bot)$ where N is unsatisfiable.

Theorem (CDCL Strong Completeness)

For any interpretation M, there is a reasonable sequence of rule applications generating $(M'; N; U; k; \top)$ as a final state, where M and M' only differ in the order of literals.

Theorem (CDCL Termination)

CDCL always terminates reasonably in a state $(M; N; U; k; D)$ with $D \in \{\top, \bot\}$.

Superposition & CDCL

Theorem (N^H_Σ & CDCL)

If $(L_1 \dots L_n; N; U; k; \top)$ is a CDCL state, $\text{atom}(L_i) = P_i$ and $P_1 \prec P_2 \prec \dots \prec P_n$ and $H(P_i) = 1$ if P_i is a decision literal, then $N^H_{P_1, P_2, \dots, P_n}$ contains exactly the atoms from $L_1 \dots L_n$.

CDCL & Redundancy

A learned CDCL clause is the result of a superposition inference and not redundant.

Extension III: Ordering Change

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

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

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

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

Flexible Model & Redundancy

$N = \{P \vee Q,\; R \vee \neg P\}$ ordering $R \prec Q \prec P$ model $N_{\mathcal{I}} = \{P\}$

$\Rightarrow_{\mathsf{SUP}} Q \vee R$ not redundant

change ordering $P \prec Q \prec R$

now $Q \vee R$ redundant

- flexible ordering not compatible with redundancy
- superposition redundancy is not compatible with CDCL reasoning
- flexible models enable only weaker notions of redundancy
- still: at any point in time, learned CDCL clauses are non-redundant

Be Small

Implementing CDCL

- is subject for an independent tutorial
- like resolution CDCL learns many clauses
- not redundant at creation, but become redundant
- about 10% are subsumed by subsequent learned clause
- checking subsumption for every learned clause is too expensive
- greedily throw away learned clauses by activity heuristic
- slowly increase number of overall kept learned clauses

Summary

Propositional Superposition

- combines static models, fresh learning, redundancy elimination
- explicit ordering
- ordering changes conflict with redundancy

CDCL

- combines flexible models, fresh learning
- implicit ordering
- model changes conflict with redundancy

Future Research

Relationship between flexible models and redundancy.

References Propositional Reasoning


Christoph Weidenbach.
Automated Reasoning.
Lecture Script WS14/15.
http://www.mpi-inf.mpg.de/departments/automation-of-logic/teaching/.


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.


Christoph Weidenbach.
Automated Reasoning.
CRC. 201X. To appear.

References Propositional Reasoning


Armin Biere, Marijn Heule, Hans van Maaren, and Toby Walsh, editors.
Handbook of Satisfiability, volume 185 of Frontiers in Artificial Intelligence and Applications. IOS Press, 2009.