2.7.8 Definition (Saturation)

A set N of clauses is called *saturated up to redundancy*, if any
superpos.
inference from non-redundant clauses in N yields a redundant clause with respect to N or is already contained in N.

Superposition Reduction Rules

see Resolution calculus

Subsumption $(N \uplus \{C_1, C_2\}) \Rightarrow_{\mathrm{SUP}} (N \cup \{C_1\})$

provided $C_1 \subsetneq C_2$

Tautology Deletion $(N \uplus \{C \lor P \lor \neg P\}) \Rightarrow_{\mathrm{SUP}} (N)$

Condensation $(N \uplus \{C_1 \lor L \lor L\}) \Rightarrow_{\mathrm{SUP}} (N \cup \{C_1 \lor L\})$

Subsumption Resolution $(N \uplus \{C_1 \lor L, C_2 \lor \mathrm{comp}(L)\}) \Rightarrow_{\mathrm{SUP}}$ $(N \cup \{C_1 \lor L, C_2\})$

where $C_1 \subseteq C_2$

2.7.9 Proposition (Reduction Rules)

All clauses removed by Subsumption, Tautology Deletion, Condensation and Subsumption Resolution are redundant with respect to the kept or added clauses.

2.7.10 Corollary (Soundness)

Superposition is sound.

2.7.11 Theorem (Completeness)

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

partial model operator

2.7.10 Corollary (Soundness)

Superposition is sound.

The superposition calculus is a refinement of the resolution calculus, which is already sound. □

2.7.11 Theorem (Completeness)

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

By contradiction: assume
- N saturated
- $\perp \notin N$
- $N_I \not\models N$

Then there is a minimal $C \in N$ which is false,
so $N_I \not\models C$

C is not redundant, otherwise
$N^{<C} \models C$ and $N_I \models N^{<C}$

$N_I \models N^{<C} \models C$ ↯ $\Rightarrow$ C not redundant ✓

D	N_D	S_D	why?
C_1	$\vdots$	$\vdots$	$\vdots$
C_2			
$\vdots$			
C		$\emptyset$	$N_C \not\models C$, $N_\bot \not\models C$

why didn't C produce?

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

Why is $\delta_C = \emptyset$

Case distinction on the shape of C

(1) $C = C' \vee P^* \vee P^*$

positive

apply Factoring, get $(C' \vee P^*) < C$

$N_I \not\models C \Rightarrow N_I \not\models C' \vee P^*$

N is saturated $\Rightarrow C' \vee P \in N$

false, in N, $< C$

$\Rightarrow$ Contradiction to the minimality of C

(2) $C = C' \vee \neg P^*$ or $\neg P$ is selected $\neg P^+$

(2) $C = C' \vee \neg P^{+,*}$

$\Rightarrow C$ is false, $P \in N_I$

$\Rightarrow$ There is a clause $D \vee P^*$ with $\delta_{D \vee P^*} = \{P\}$

$\Rightarrow N_D \not\models D \underset{2.7.7}{\Longrightarrow} N_I \not\models D$

$(D \vee P^*) < (C' \vee \neg P^*)$

$\Rightarrow$ Superposition Left: $D \vee C' < (C' \vee \neg P^*)$

$N_I \not\models D \vee C'$

$D \vee C'$ cannot be redundant:

If it were, then $N^{<D \vee C'} \models D \vee C'$, but this means

$N_I \not\models N^{<D \vee C'}$ ↯ contradiction to min. of C

false clause in here

We have a clause $D \vee C'$: $D \vee C' \in N$ $D \vee C' < C$

$N_I \not\models D \vee C'$ $\Rightarrow$ contr. to min. of C

A Recipe for Superposition

Input: • a clause set N
• an ordering $\prec$ on atoms

1. Order the clauses, find max. literal in each clause
2. Run the partial model operator N_I (table!)
3. Find the minimal false clause w.r.t. N_I
   - $\bot \in N$: Then N is unsatisfiable, stop.
   - If there is no m.f.c., $N_I \models N$, N is satisfiable, stop.
   - m.f.c. has the shape $C \lor L^* \lor L^*$: factoring
   - m.f.c. has the shape $C' \lor \neg L^*$: $C' \lor \neg L^+$: superposition left with the "productive counterpart"

$$P \prec Q \prec R \prec S$$

$N = \{ R \vee S^*,\ \{\neg P^+ \vee Q,\quad Q \vee R^* \vee R^*,\quad \neg P \vee S^* \}$

	N_D	δ_D	why?
$\neg P^+ \vee Q$	$\emptyset$	$\emptyset$	$\neg P$ selected, true
$Q \vee R^* \vee R^*$	$\emptyset$	$\emptyset$	R is not strict. max
$\neg P \vee S$	$\emptyset$	$\emptyset$	true
$R \vee S^*$	$\emptyset$	$\{S\}$	false

Factorization:

new clause $Q \vee R^*$

$N_I = \{S\}$

$\Updownarrow$

$A(S) = 1$

$A(P) = A(Q) = A(R) = 0$

$$P \prec Q \prec R \prec S$$

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

D	N_D	δ_D	why
$\neg P^+ \vee Q$	$\emptyset$	$\emptyset$	$\neg P$ sel., true
$Q \vee R^*$	$\emptyset$	$\{R\}$	R strict. max.
$Q \vee R^* \vee R^*$	$\{R\}$	$\emptyset$	true
$\neg P \vee S^{\emptyset}$	$\{R\}$	$\emptyset$	true
$R \vee S^*$	$\{R\}$	$\emptyset$	true

$$N_I = \{R\}$$

$$\updownarrow$$

$$A(R) = 1$$
$$A(P) = A(Q) = A(S) = 0$$

$$N_I \models N$$

Superposition with(out) the Partial Model Operator

Superposition Left $(N \uplus \{C_1 \lor P, C_2 \lor \neg P\}) \Rightarrow_{\text{SUP}}$ $(N \cup \{C_1 \lor P, C_2 \lor \neg P\} \cup \{C_1 \lor C_2\})$

where (i) P is strictly maximal in $C_1 \lor P$ (ii) no literal in $C_1 \lor P$ is selected (iii) $\neg P$ is maximal and no literal selected in $C_2 \lor \neg P$, or $\neg P$ is selected in $C_2 \lor \neg P$

Superposition on minimal false clauses with the partial model operator

▷ complete (see completeness proof), sound

▷ deterministic

▷ slow + inflexible

▷ the result is non-redundant ← always make progress

Practice: trade-off between m.f.c. or trying other inf.

Superposition and CDCL

Using an appropriate ordering, and model construction operator, clauses learned by CDCL are actually non-redundant in sense of superposition.

This section explains why.

2.11.1 Definition (Heuristic-Based Partial Model Construction)

Given a clause set N, a set of propositional variables $M \subseteq \Sigma$, a total ordering $\prec$, and a variable heuristic $\mathcal{H} : \Sigma \to \{0, 1\}$, the (partial) model $N_M^{\mathcal{H}}$ for N with $P, Q \in M$ is inductively constructed as follows:

$$N_P^{\mathcal{H}} := \bigcup_{Q \prec P} \delta_Q^{\mathcal{H}} \qquad N_M^{\mathcal{H}} := \bigcup_{P \in M} \delta_P^{\mathcal{H}}$$

$$\delta_P^{\mathcal{H}} := \begin{cases} \{P\} & \text{if there is a clause } (D \vee P) \in N, \text{ such that} \\ & N_P^{\mathcal{H}} \models \neg D \text{ and either } P \text{ is strictly maximal or} \\ & \mathcal{H}(P) = 1 \text{ and there is no clause} \\ & (D' \vee \neg P) \in N, D' \prec P \text{ such that } N_P^{\mathcal{H}} \models \neg D' \\ \emptyset & \text{otherwise} \end{cases}$$

The heuristic-based model operator $N_M^{\mathcal{H}}$ enjoys many properties of the standard model operator $N_{\mathcal{I}}$ and generalizes it.

Lemma ($N_M^{\mathcal{H}}$ generalizes $N_{\mathcal{I}}$)

If $\mathcal{H}(P) = 0$ for all $P \in \Sigma$ then $N_{\mathcal{I}} = N_{\Sigma}^{\mathcal{H}}$ for any N.

So the new model operator $N_M^{\mathcal{H}}$ is a generalization of $N_{\mathcal{I}}$.

With the help of $N_M^{\mathcal{H}}$ a close relationship between the model assumptions generated by the CDCL calculus and the superposition model operator can be established.

2.11.3 Theorem (Completeness with $N_M^{\mathcal{H}}$)

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

2.11.4 Theorem ()

Let $(M, N, U, k, C \vee K)$ be a CDCL state generated by rule Conflict and a reasonable strategy where $M = L_1, \ldots, L_n$. Let $\mathcal{H}(\text{atom}(L_m)) = 1$ for any positive decision literal L_m^i occurring in M and $\mathcal{H}(\text{atom}(L_m)) = 0$ otherwise. Furthermore, I assume that if CDCL can propagate both P and $\neg P$ in some state, then it propagates P. The superposition precedence is $\text{atom}(L_1) \prec \text{atom}(L_2) \prec \ldots \prec \text{atom}(L_n)$. Let K be maximal in $C \vee K$ and $C \vee K$ be the minimal false clause with respect to $\prec$. Then

1. L_n is a propagated literal and $K = \text{comp}(L_n)$.
2. The clause generated by $C \vee K$ and the clause propagating L_n is the result of a Superposition Left inference between the clauses and it is not redundant.
3. $N^{\mathcal{H}}_{\{L_1,\ldots,L_n\}} = \{P \mid P \in M\}$

Theorem 2.11.4 is actually a nice explanation for the efficiency of the CDCL procedure: a learned clause is never redundant. Recall that redundancy here means that the learned clause C is not entailed by smaller clauses in $N \cup U$.

Furthermore, the ordering underlying Theorem 2.11.4 is based on the trail, i.e., it changes during a CDCL run. For superposition it is well known that changing the ordering is not compatible with the notion of redundancy, i.e., superposition is incomplete when the ordering may be changed infinitely often and the superposition redundancy notion is applied.

Theorem 2.11.4 is actually a nice explanation for the efficiency of the CDCL procedure: a learned clause is never redundant. Recall that redundancy here means that the learned clause C is not entailed by smaller clauses in $N \cup U$.

Furthermore, the ordering underlying Theorem 2.11.4 is based on the trail, i.e., it changes during a CDCL run. For superposition it is well known that changing the ordering is not compatible with the notion of redundancy, i.e., superposition is incomplete when the ordering may be changed infinitely often and the superposition redundancy notion is applied.

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

$(\epsilon; N; \emptyset; 0; \top)$

$\Rightarrow_{\text{Decide}} (P^1; N; \emptyset; 1; \top)$

$\Rightarrow_{\text{Decide}} (P^1 \neg Q^2; N; \emptyset; 2; \top)$

$\Rightarrow_{\text{Propagate}} (P^1 \neg Q^2 R^{\neg P \vee Q \vee R}; N; \emptyset; 2; \top)$

$\Rightarrow_{\text{Conflict}} (P^1 \neg Q^2 R^{\neg P \vee Q \vee R}; \text{"}; \emptyset; 2; \neg P \vee Q \vee \neg R)$

$\Rightarrow_{\text{Resolve}} (P^1 \neg Q^2; N; \emptyset; 2; \neg P \vee Q \vee \cancel{Q \vee \neg P})$

$\Rightarrow_{\text{Backtrack}} (P^1; N; \{\neg P \vee Q\}; 1; \top)$

$P \prec Q \prec R$ $\quad H(P) = \neg$ $\quad H(Q) = 0$

$H(R) = \neg$

$\neg P \vee Q \vee \neg R^* \longleftrightarrow \neg P \vee Q \vee R^*$

superposition

Furthermore, the ordering underlying Theorem 2.11.4 is based on the trail, i.e., it changes during a CDCL run. For superposition it is well known that changing the ordering is not compatible with the notion of redundancy, i.e., superposition is incomplete when the ordering may be changed infinitely often and the superposition redundancy notion is applied.

2.11.7 Example (Superposition diverges under changed ordering)

Consider the superposition left inference between the clauses $P \lor Q$ and $R \lor \neg Q$ with ordering $P \prec R \prec Q$ resulting in $P \lor R$. Changing the ordering to $Q \prec P \prec R$ the inference $P \lor R$ becomes redundant. So flipping infinitely often between $P \prec R \prec Q$ and $Q \prec P \prec R$ is already sufficient to prevent any saturation progress.

$Q < P < R$: $P \lor R$ redundant? $N^{< P \lor R} \models P \lor R$

$\{P \lor Q, R \lor \neg Q\} \not\models P \lor R$