Step 4: Apply steps 2, 3, 4, 5 of $\Rightarrow_{\text{ECNF}}$

Remark: The $\Rightarrow_{\text{OCNF}}$ algorithm is already close to a state of the art algorithm. Missing are further redundancy tests and simplification mechanisms we will discuss later on in this section.

### 2.5 Superposition for $\text{PROP}(\Sigma)$

Superposition for $\text{PROP}(\Sigma)$ is:

- resolution (Robinson 1965) +
- ordering restrictions (Bachmair & Ganzinger 1990) +
- abstract redundancy criterion (B&G 1990) +
- partial model construction (B & G 1990) +
- partial-model based inference restriction (Weidenbach)

### Resolution for $\text{PROP}(\Sigma)$

A calculus is a set of inference and reduction rules for a given logic (here $\text{PROP}(\Sigma)$).

We only consider calculi operating on a set of clauses $N$. Inference rules add new clauses to $N$ whereas reduction rules remove clauses from $N$ or replace clauses by “simpler” ones.

We are only interested in unsatisfiability, i.e., the considered calculi test whether a clause set $N$ is unsatisfiable. So, in order to check validity of a formula $\phi$ we check unsatisfiability of the clauses generated from $\neg\phi$.

For clauses we switch between the notation as a disjunction, e.g., $P \lor Q \lor P \lor \neg R$, and the notation as a multiset, e.g., $\{P, Q, P, \neg R\}$. This makes no difference as we consider $\lor$ in the context of clauses always modulo AC. Note that $\bot$, the empty disjunction, corresponds to $\emptyset$, the empty multiset.

For literals we write $L$, possibly with subscript. If $L = P$ then $\bar{L} = \neg P$ and if $L = \neg P$ then $\bar{L} = P$, so the bar flips the negation of a literal.

Clauses are typically denoted by letters $C, D$, possibly with subscript.

The resolution calculus consists of the inference rules resolution and factoring:

\[
\begin{array}{c}
\text{Resolution} \\
\frac{C_1 \lor P \quad C_2 \lor \neg P}{C_1 \lor C_2}
\end{array}
\quad
\begin{array}{c}
\text{Factoring} \\
\frac{C \lor L \lor L}{C \lor L}
\end{array}
\]
where \( C_1, C_2, C \) always stand for clauses, all inference/reduction rules are applied with respect to AC of \( \lor \). Given a clause set \( N \) the schema above the inference bar is mapped to \( N \) and the resulting clauses below the bar are then added to \( N \).

and the reduction rules subsumption and tautology deletion:

\[
\begin{array}{ccc}
\text{Subsumption} & \text{Tautology Deletion} \\
R & \frac{C_1 \quad C_2}{C_1} & \frac{C \lor P \lor \neg P}{\mathcal{R}}
\end{array}
\]

where for subsumption we assume \( C_1 \subseteq C_2 \). Given a clause set \( N \) the schema above the reduction bar is mapped to \( N \) and the resulting clauses below the bar replace the clauses above the bar in \( N \).

Clauses that can be removed are called redundant.

So, if we consider clause sets \( N \) as states, \( \uplus \) is disjoint union, we get the rules

\[
\begin{align*}
\text{Resolution} & \quad (N \uplus \{C_1 \lor P, C_2 \lor \neg P\}) \quad \Rightarrow \quad (N \cup \{C_1 \lor P, C_2 \lor \neg P\} \cup \{C_1 \lor C_2\}) \\
\text{Factoring} & \quad (N \uplus \{C \lor L \lor L\}) \quad \Rightarrow \quad (N \cup \{C \lor L \lor L\} \cup \{C \lor L\}) \\
\text{Subsumption} & \quad (N \uplus \{C_1, C_2\}) \quad \Rightarrow \quad (N \cup \{C_1\}) \\
\text{Tautology Deletion} & \quad (N \uplus \{C \lor P \lor \neg P\}) \quad \Rightarrow \quad (N)
\end{align*}
\]

provided \( C_1 \subseteq C_2 \)

We need more structure than just \( (N) \) in order to define a useful rewrite system. We fix this later on.

**Theorem 2.11** The resolution calculus is sound and complete:

\( N \) is unsatisfiable iff \( N \Rightarrow^* \{\bot\} \)

**Proof.** Will be a consequence of soundness and completeness of superposition. \( \square \)
Ordering restrictions

Let $\prec$ be a total ordering on $\Sigma$.

We lift $\prec$ to a total ordering on literals by $\prec \subseteq \prec_L$ and $P \prec_L \neg P$ and $\neg P \prec_L Q$ for all $P \prec Q$.

We further lift $\prec_L$ to a total ordering on clauses $\prec_C$ by considering the multiset extension of $\prec_L$ for clauses.

Eventually, we overload $\prec$ with $\prec_L$ and $\prec_C$.

We define $N^{\prec C} = \{ D \in N \mid D \prec C \}$.

Eventually we will restrict inferences to maximal literals with respect to $\prec$.

Abstract Redundancy

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

Tautologies are redundant. Subsumed clauses are redundant if $\subseteq$ is strict.

Remark: Note that for finite $N$, $N^{\prec C} \models C$ can be decided for $PROP(\Sigma)$ but is as hard as testing unsatisfiability for a clause set $N$.

Partial Model Construction

Given a clause set $N$ and an ordering $\prec$ we can construct a (partial) model $N_I$ for $N$ as follows:

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

$\delta_D := \begin{cases} \{P\} & \text{if } D = D' \lor P \text{ and } P \text{ maximal and } N_D \not= D \\ \emptyset & \text{otherwise} \end{cases}$

$N_I := \bigcup_{C \in N} \delta_C$
Superposition

The superposition calculus consists of the inference rules superposition left and factoring:

Superposition Left

\[(N \uplus \{C_1 \lor P, C_2 \lor \neg P\}) \implies (N \cup \{C_1 \lor P, C_2 \lor \neg P\} \cup \{C_1 \lor \neg P\})\]

where \(P\) is strictly maximal in \(C_1 \lor P\) and \(\neg P\) is maximal in \(C_2 \lor \neg P\)

Factoring

\[(N \uplus \{C \lor P \lor P\}) \implies (N \cup \{C \lor P \lor P\} \cup \{C \lor P\})\]

where \(P\) is maximal in \(C \lor P \lor P\)

examples for specific redundancy rules are

Subsumption

\[(N \uplus \{C_1, C_2\}) \implies (N \cup \{C_1\})\]

provided \(C_1 \subseteq C_2\)

Tautology

\[(N \uplus \{C \lor P \lor \neg P\}) \implies (N)\]

Deletion

\[(N \uplus \{C \lor P \lor \neg P\}) \implies (N)\]

Subsumption

\[(N \uplus \{C_1 \lor L, C_2 \lor \neg L\}) \implies (N \cup \{C_1 \lor L, C_2\})\]

where \(C_1 \subseteq C_2\)

Theorem 2.12 If from a clause set \(N\) all possible superposition inferences are redundant and \(\bot \notin N\) then \(N\) is satisfiable and \(N_T \models N\).