Proof. Translate ϕ and $\neg\psi$ into CNF. let N and M, respectively, denote the resulting clause set. Choose an atom ordering $\succ$ for which the propositional variables that occur in ϕ but not in ψ are maximal. Saturate N into N^* w.r.t. $Sup^{\succ}_{sel}$ with an empty selection function sel. Then saturate $N^* \cup M$ w.r.t. $Sup^{\succ}_{sel}$ to derive $\bot$. As N^* is already saturated, due to the ordering restrictions only inferences need to be considered where premises, if they are from N^*, only contain symbols that also occur in ψ. The conjunction of these premises is an interpolant χ. The theorem also holds for first-order formulas. For universal formulas the above proof can be easily extended. In the general case, a proof based on superposition technology is more complicated because of Skolemization. □

3.13 First-Order Superposition

Now the result for ground superposition are lifted to superposition on first-order clauses with variables, still without equality. The completeness proof of ground superposition above talks about (strictly) maximal literals of *ground* clauses. The non-ground calculus considers those literals that correspond to (strictly) maximal literals of ground instances.

The used ordering is exactly the ordering of Definition 3.12.1 where clauses with variables are projected to their ground instances for ordering computations.

Definition 3.13.1 (Maximal Literal)**.** A literal L is called *maximal* in a clause C if and only if there exists a grounding substitution σ so that $L\sigma$ is maximal in $C\sigma$, i.e., there is no different $L' \in C$: $L\sigma \prec L'\sigma$. The literal L is called *strictly maximal* if there is no different $L' \in C$ such that $L\sigma \preceq L'\sigma$.

Note that the orderings KBO and LPO cannot be total on atoms with variables, because they are stable under substitutions. Therefore, maximality can also be defined on the basis of absence of greater literals. A literal L is called *maximal* in a clause C if $L \not\prec L'$ for all other literals $L' \in C$. It is called *strictly maximal* in a clause C if $L \not\preceq L'$ for all other literals $L' \in C$.

Superposition Left $(N \uplus \{C_1 \lor P(t_1,\ldots,t_n), C_2 \lor \neg P(s_1,\ldots,s_n)\}) \Rightarrow_{\text{SUP}}$ $(N \cup \{C_1 \lor P(t_1,\ldots,t_n), C_2 \lor \neg P(s_1,\ldots,s_n)\} \cup \{(C_1 \lor C_2)\sigma\})$

where (i) $P(t_1,\ldots,t_n)\sigma$ is strictly maximal in $(C_1 \lor P(t_1,\ldots,t_n))\sigma$ (ii) no literal in $C_1 \lor P(t_1,\ldots,t_n)$ is selected (iii) $\neg P(s_1,\ldots,s_n)\sigma$ is maximal and no literal selected in $(C_2 \lor \neg P(s_1,\ldots,s_n))\sigma$, or $\neg P(s_1,\ldots,s_n)$ is selected in $(C_2 \lor \neg P(s_1,\ldots,s_n))\sigma$ (iv) σ is the mgu of $P(t_1,\ldots,t_n)$ and $P(s_1,\ldots,s_n)$

Factoring $(N \uplus \{C \lor P(t_1,\ldots,t_n) \lor P(s_1,\ldots,s_n)\}) \Rightarrow_{\text{SUP}}$ $(N \cup \{C \lor P(t_1,\ldots,t_n) \lor P(s_1,\ldots,s_n)\} \cup \{(C \lor P(t_1,\ldots,t_n))\sigma\})$

where (i) $P(t_1,\ldots,t_n)\sigma$ is maximal in $(C \lor P(t_1,\ldots,t_n) \lor P(s_1,\ldots,s_n))\sigma$ (ii) no literal is selected in $C \lor P(t_1,\ldots,t_n) \lor P(s_1,\ldots,s_n)$ (iii) σ is the mgu of $P(t_1,\ldots,t_n)$ and $P(s_1,\ldots,s_n)$

Note that the above inference rules Superposition Left and Factoring are generalizations of their respective counterparts from the ground superposition calculus above. Therefore, on ground clauses they coincide. Therefore, we can safely overload them in the sequel.

Definition 3.13.2 (Abstract Redundancy)**.** A clause C is *redundant* with respect to a clause set N if for all ground instances $C\sigma$ there are clauses $\{C_1, \ldots, C_n\} \subseteq N$ with ground instances $C_1\tau_1, \ldots, C_n\tau_n$ such that $C_i\tau_i \prec C\sigma$ for all i and $C_1\tau_1, \ldots, C_n\tau_n \models C\sigma$.

Definition 3.13.3 (Saturation)**.** A set N of clauses is called *saturated up to redundancy*, if any inference from non-redundant clauses in N yields a redundant clause with respect to N or is contained in N.

In contrast to the ground case, the above abstract notion of redundancy is not effective, i.e., it is undecidable for some clause C whether it is redundant, in general. Nevertheless, the concrete ground redundancy notions carry over to the non-ground case. Note also that a clause C is contained in N modulo renaming of variables.

Let rdup be a function from clauses to clauses that removes duplicate literals, i.e., $\mathrm{rdup}(C) = C'$ where $C' \subseteq C$, C' does not contain any duplicate literals, and for each $L \in C$ also $L \in C'$.

Subsumption $\quad (N \uplus \{C_1, C_2\}) \ \Rightarrow_{\mathrm{SUP}} \ (N \cup \{C_1\})$
provided $C_1\sigma \subset C_2$ for some σ

Tautology Deletion $\quad (N \uplus \{C \vee P(t_1, \ldots, t_n) \vee \neg P(t_1, \ldots, t_n)\}) \ \Rightarrow_{\mathrm{SUP}} \ (N)$

Condensation $\quad (N \uplus \{C_1 \vee L \vee L'\}) \ \Rightarrow_{\mathrm{SUP}} \ (N \cup \{\mathrm{rdup}((C_1 \vee L \vee L')\sigma)\})$
provided $L\sigma = L'$ and $\mathrm{rdup}((C_1 \vee L \vee L')\sigma)$ subsumes $C_1 \vee L \vee L'$ for some σ

Subsumption Resolution $\quad (N \uplus \{C_1 \vee L, C_2 \vee L'\}) \ \Rightarrow_{\mathrm{SUP}} \ (N \cup \{C_1 \vee L, C_2\})$
where $L\sigma = \neg L'$ and $C_1\sigma \subseteq C_2$ for some σ

Lemma 3.13.4. All reduction rules are instances of the abstract redundancy criterion.

Proof. Do it □

Lemma 3.13.5 (Subsumption is NP-complete)**.** Subsumption is NP-complete.

Proof. Let C_1 subsume C_2 with substitution σ Subsumption is in NP because the size of σ is bounded by the size of C_2 and the subset relation can be checked in time at most quadratic in the size of C_1 and C_2.

Propositional SAT can be reduced as follows. Assume a 3-SAT clause set N. Consider a 3-place predicate R and a unary function g and a mapping from propositional variables P to first order variables x_P. □

Lemma 3.13.6 (Lifting)**.** Let $D \vee L$ and $C \vee L'$ be variable-disjoint clauses and σ a grounding substitution for $C \vee L$ and $D \vee L'$. If there is a superposition left inference

$$(N \uplus \{(D \vee L)\sigma, (C \vee L')\sigma\}) \Rightarrow_{\text{SUP}} (N \cup \{(D \vee L)\sigma, (C \vee L')\sigma\} \cup \{D\sigma \vee C\sigma\})$$

and if $\text{sel}((D \vee L)\sigma) = \text{sel}((D \vee L))\sigma$, $\text{sel}((C \vee L')\sigma) = \text{sel}((C \vee L'))\sigma$, then there exists a mgu τ such that

$$(N \uplus \{D \vee L, C \vee L'\}) \Rightarrow_{\text{SUP}} (N \cup \{D \vee L, C \vee L'\} \cup \{(D \vee C)\tau\}).$$

Let $C \vee L \vee L'$ be a clause and σ a grounding substitution for $C \vee L \vee L'$. If there is a factoring inference

$$(N \uplus \{(C \vee L \vee L')\sigma\}) \Rightarrow_{\text{SUP}} (N \cup \{(C \vee L \vee L')\sigma\} \cup \{(C \vee L)\sigma\})$$

and if $\text{sel}((C \vee L \vee L')\sigma) = \text{sel}((C \vee L \vee L'))\sigma$, then there exists a mgu τ such that

$$(N \uplus \{C \vee L \vee L'\}) \Rightarrow_{\text{SUP}} (N \cup \{C \vee L \vee L'\} \cup \{(C \vee L)\tau\})$$

Note that in the above lemma the clause $D\sigma \vee C\sigma$ is an instance of the clause $(D \vee C)\tau$. The reduction rules cannot be lifted in the same way as the following example shows.

Example 3.13.7 (First-Order Reductions are not Liftable)**.** Consider the two clauses $P(x) \vee Q(x)$, $P(g(y))$ and grounding substitution $\{x \mapsto g(a), y \mapsto a\}$. Then $P(g(y))\sigma$ subsumes $(P(x) \vee Q(x))\sigma$ but $P(g(y))$ does not subsume $P(x) \vee Q(x)$. For all other reduction rules similar examples can be constructed.

Lemma 3.13.8 (Soundness and Completeness)**.** First-Order Superposition is sound and complete.

Proof. Soundness is obvious. For completeness, Theorem 3.12.12 proves the ground case. Now by applying Lemma 3.13.6 to this proof it can be lifted to the first-order level, as argued in the following.

Let N be a an unsatisfiable set of first-order clauses. By Theorem 3.5.5 and Lemma 3.6.10 there exist a finite unsatisfiable set N' of ground instances from clauses from N such that for each clause $C\sigma \in N'$ there is a clause $C \in N$. Now ground superposition is complete, Theorem 3.12.12, so there exists a derivation of the empty clause by ground superposition from N': $N' = N'_0 \Rightarrow_{\text{SUP}} \ldots \Rightarrow_{\text{SUP}} N'_k$ and $\bot \in N'_k$. Now by an inductive argument on the length of the derivation k this derivation can be lifted to the first-order level. The invariant is: for any ground clause $C\sigma \in N'_i$ used in the ground proof, there is a clause $C \in N_i$ on the first-order level. The induction base holds for N and N' by construction. For the induction step Lemma 3.13.6 delivers the result. □

There are questions left open by Lemma 3.13.8. It just says that a ground refutation can be lifted to a first-order refutation. But what about abstract redundancy, Definition 3.13.2? Can first-order redundant clauses be deleted without harming completeness? And what about the ground model operator with respect to clause sets N saturated on the first-order level. Is in this case $\mathrm{grd}(\Sigma, N)_{\mathcal{I}}$ a model? The next two lemmas answer these questions positively.

Lemma 3.13.9 (Redundant Clauses are Obsolete)**.** If a clause set N is unsatisfiable, then there is a derivation $N \Rightarrow^{*}_{\mathrm{SUP}} N'$ such that $\bot \in N'$ and no clause in the derivation of $\bot$ is redundant.

Proof. If N is unsatisfiable then there is a ground superposition refutation of $\mathrm{grd}(\Sigma, N)$ such that no ground clause in the refutation is redundant. Now according to Lemma 3.13.8 this proof can be lifted to the first-order level. Now assume some clause C in the first-order proof is redundant that is the lifting of some clause $C\sigma$ from the ground proof with respect to a grounding substitution σ. The clause C is redundant by Definition 3.13.2 if all its ground instances are, in particular, $C\sigma$. But this contradicts the fact that the lifted ground proof does not contain redundant clauses. □

Lemma 3.13.10 (Model Property)**.** If N is a saturated clause set and $\bot \notin N$ then $\mathrm{grd}(\Sigma, N)_{\mathcal{I}} \models N$.

Proof. As usual we assume that selection on the ground and respective non-ground clauses is identical. Assume $\mathrm{grd}(\Sigma, N)_{\mathcal{I}} \not\models N$. Then there is a minimal ground clause $C\sigma$, $C \neq \bot$, $C \in N$ such that $\mathrm{grd}(\Sigma, N)_{\mathcal{I}} \not\models C\sigma$. Note that $C\sigma$ is not redundant as for otherwise $\mathrm{grd}(\Sigma, N)_{\mathcal{I}} \models C\sigma$. So $\mathrm{grd}(\Sigma, N)$ is not saturated. If $C\sigma$ is productive, i.e., $C\sigma = (C' \vee L)\sigma$ such that L is positive, $L\sigma$ strictly maximal in $(C' \vee L)\sigma$ then $L\sigma \in \mathrm{grd}(\Sigma, N)_{\mathcal{I}}$ and hence $\mathrm{grd}(\Sigma, N)_{\mathcal{I}} \models C\sigma$ contradicting $\mathrm{grd}(\Sigma, N)_{\mathcal{I}} \not\models C\sigma$.

If $C\sigma = (C' \vee L \vee L')\sigma$ such that L is positive, $L\sigma$ maximal in $(C' \vee L \vee L')\sigma$ then, because N is saturated, there is a clause $(C' \vee L)\tau \in N$ such that $(C' \vee L)\tau\sigma = (C' \vee L)\sigma$. Now $(C' \vee L)\tau$ is not redundant, $\mathrm{grd}(\Sigma, N)_{\mathcal{I}} \not\models (C' \vee L)\tau$, contradicting the minimal choice of $C\sigma$.

If $C\sigma = (C' \vee L)\sigma$ such that L is selected, or negative and maximal then there is a clause $(D' \vee L') \in N$ and grounding substitution ρ, such that $L'\rho$ is a strictly maximal positive literal in $(D' \vee L')\rho$, $L'\rho \in \mathrm{grd}(\Sigma, N)_{\mathcal{I}}$ and $L'\rho = \neg L\sigma$. Again, since N is saturated, there is variable disjoint clause $(C' \vee D')\tau \in N$ for some unifier τ, $(C' \vee D')\tau\sigma\rho \prec C\sigma$, and $\mathrm{grd}(\Sigma, N)_{\mathcal{I}} \not\models (C' \vee D')\tau\sigma\rho$ contradicting the minimal choice of $C\sigma$. □

Dynamic stuff: a clause C is called *persistent* in a derivation $N \rightarrow^{*}_{\mathrm{SUP}} N'$ if there is some i such that $C \in N_i$ for $N \rightarrow^{i}_{\mathrm{SUP}} N_i$ and for all $j > i$, $N \rightarrow^{j}_{\mathrm{SUP}} N_j$ then $C \in N_j$. A derivation $N \rightarrow^{*}_{\mathrm{SUP}} N'$ is called *fair* if any two persistent clauses C, D and any superposition inference C' out of the two clauses, there is an index j such with $N \rightarrow^{j}_{\mathrm{SUP}} N_j \rightarrow^{*}_{\mathrm{SUP}} N'$ such that $C' \in N_j$.

Definition 3.13.11 (Persistent Clause)**.** Let $N_0 \Rightarrow_{\mathrm{SUP}} N_1 \Rightarrow_{\mathrm{SUP}} \ldots$ be a, possibly infinite, superposition derivation. A clause C is called *persistent* in this derivation if $C \in N_i$ for some i and for all $j > i$ also $C \in N_j$.