3.12.8 Proposition (Completeness of the Reduction Rules)

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

3.12.9 Theorem (Completeness)

Let N be a, possibly countably infinite, set of ground clauses. If N is saturated up to redundancy and $\bot \notin N$ then N is satisfiable and $N_{\mathcal{I}} \models N$.

3.12.10 Theorem (Compactness of First-Order Logic)

Let N be a, possibly countably infinite, set of first-order logic ground clauses. Then N is unsatisfiable iff there is a finite subset $N' \subseteq N$ such that N' is unsatisfiable.

3.12.11 Corollary (Compactness of First-Order Logic: Classical)

A set N of clauses is satisfiable iff all finite subsets of N are satisfiable.

3.12.12 Theorem (Soundness and Completeness of Ground Superposition)

A first-order Σ-sentence ϕ is valid iff there exists a ground superposition refutation for $\text{ground}(\Sigma, \text{cnf}(\neg\phi))$.

3.12.13 Theorem (Semi-Decidability of First-Order Logic by Ground Superposition)

If a first-order Σ-sentence ϕ is valid then a ground superposition refutation can be computed.

3.12.15 Theorem (Craig's Theorem)

Let ϕ and ψ be two propositional (first-order ground) formulas so that $\phi \models \psi$. Then there exists a formula χ (called the *interpolant* for $\phi \models \psi$), so that χ contains only propositional variables (first-order signature symbols) occurring both in ϕ and in ψ so that $\phi \models \chi$ and $\chi \models \psi$.

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.

3.13.1 Definition (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

$(N \uplus \{C_1 \vee P(t_1,\ldots,t_n), C_2 \vee \neg P(s_1,\ldots,s_n)\}) \ \Rightarrow_{\mathrm{SUP}}$
$(N \cup \{C_1 \vee P(t_1,\ldots,t_n), C_2 \vee \neg P(s_1,\ldots,s_n)\} \cup \{(C_1 \vee C_2)\sigma\})$

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

Factoring

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

where (i) $P(t_1,\ldots,t_n)\sigma$ is maximal in $(C \vee P(t_1,\ldots,t_n) \vee P(s_1,\ldots,s_n))\sigma$ (ii) no literal is selected in $C \vee P(t_1,\ldots,t_n) \vee P(s_1,\ldots,s_n)$ (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.

3.13.2 Definition (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 $C_1\tau_1, \ldots, C_n\tau_n \models C\sigma$.

3.13.3 Definition (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.

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

provided $C_1\sigma \subset C_2$ for some σ

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

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

Condensation $(N \uplus \{C_1 \lor L \lor L'\}) \Rightarrow_{\text{SUP}}$ $(N \cup \{\text{rdup}((C_1 \lor L \lor L')\sigma)\})$

provided $L\sigma = L'$ and $\text{rdup}((C_1 \lor L \lor L')\sigma)$ subsumes $C_1 \lor L \lor L'$ for some σ

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

where $L\sigma = \neg L'$ and $C_1\sigma \subseteq C_2$ for some σ