The above deterministic, linear resolution refutation, Example 2.6.4, cannot be simulated by the tableau calculus without generating an exponential overhead, see also the comment on page 37. At first, it looks strange to have the same rule, namely Factorization and Condensation, both as a reduction rules and as an inference rule. On the propositional level there is obviously no difference and it is possible to get rid of one of the two. In Section 3.10 the resolution calculus is lifted to first-order logic. In first-order logic Factorization and Condensation are actually different, i.e., a Factorization inference is no longer a Condensation simplification, in general. They are separated here to eventually obtain the same set of rules propositional and first-order logic. This is needed for a proper lifting proof of first-order completeness that us actually reduced to the ground fragment of first-order logic that can be considered as a variant of propositional logic.

Proposition 2.6.5. The reduction rules Subsumption, Tautology Deletion, Condensation and Subsumption Resolution are sound.

Proof. This is obvious for Tautology Deletion and Condensation. For Subsumption we have to show that $C_1 \models C_2$, because this guarantees that if $N \cup \{C_1\}$ has a model, $N \uplus \{C_1, C_2\}$ has a model too. So assume $\mathcal{A}(C_1) = 1$ for an arbitrary $\mathcal{A}$. Then there is some literal $L \in C_1$ with $\mathcal{A}(L) = 1$. Since $C_1 \subseteq C_2$, also $L \in C_2$ and therefore $\mathcal{A}(C_2) = 1$. Subsumption Resolution is the combination of a Resolution application followed by a Subsumption application. □

Theorem 2.6.6 (Resolution Termination)**.** If reduction rules are preferred over inference rules and no inference rule is applied twice to the same clause(s), then $\Rightarrow^{+}_{\text{RES}}$ is well-founded.

Proof. If reduction rules are preferred over inference rules, then the overall length if a clause cannot exceed n, where n is the number of variables. Multiple occurrences of the same literal are removed by rule Condensation, multiple occurrences of the same variable with different sign result in an application of rule Tautology Deletion. The number of such clauses can be overestimated by 3^n because every variable occurs at most once positively, negatively or not at all in clause. Hence, there are at most $2n3^n$ different resolution applications. □

Of course, what needs to be shown is that the strategy employed in Theorem 2.6.6 is still complete. This is not completely trivial. This result becomes a particular instance of superposition completeness. Exercise **??** contains the completeness part when the reduction rules are preferred over inference rules.

2.7 Propositional Superposition

Superposition was originally developed for first-order logic with equality [5]. Here I introduce its projection to propositional logic. Compared to the resolution calculus superposition adds (i) ordering and selection restrictions on inferences, (ii) an abstract redundancy notion, (iii) the notion of a partial model, based on the ordering for inference guidance, and (iv) a *saturation* concept.

Definition 2.7.1 (Clause Ordering)**.** Let $\prec$ be a total strict ordering on Σ. Then $\prec$ can be lifted to a total ordering on literals by $\prec \subseteq \prec_L$ and $P \prec_L \neg P$ and $\neg P \prec_L Q$, $\neg P \prec_L \neg Q$ for all $P \prec Q$. The ordering $\prec_L$ can be lifted to a total ordering on clauses $\prec_C$ by considering the multiset extension of $\prec_L$ for clauses.

For example, if $P \prec Q$, then $P \prec_L \neg P \prec_L Q \prec_L \neg Q$ and $P \vee Q \prec_C P \vee Q \vee Q \prec_C \neg Q$ because $\{P, Q\} \prec_L^{\text{mul}} \{P, Q, Q\} \prec_L^{\text{mul}} \{\neg Q\}$.

Proposition 2.7.2 (Properties of the Clause Ordering)**.** (i) The orderings on literals and clauses are total and well-founded.
(ii) Let C and D be clauses with $P = \text{atom}(\max(C))$, $Q = \text{atom}(\max(D))$, where $\max(C)$ denotes the maximal literal in C.

1. If $Q \prec_L P$ then $D \prec_C C$.
2. If $P = Q$, P occurs negatively in C but only positively in D, then $D \prec_C C$.

Eventually, I overload $\prec$ with $\prec_L$ and $\prec_C$. So if $\prec$ is applied to literals it denotes $\prec_L$, if it is applied to clauses, it denotes $\prec_C$. Note that $\prec$ is a total ordering on literals and clauses as well. Eventually we will restrict inferences to maximal literals with respect to $\prec$. For a clause set N, I define $N^{\prec C} = \{D \in N \mid D \prec C\}$.

Example 2.7.3 (Propositional Clause Ordering)**.** Let $P \prec Q \prec R \prec S$ and consider the clause set

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

then

$$\begin{array}{ll} N^{\prec C} = \emptyset & \text{if} \quad C = P \vee \neg Q \\ N^{\prec C} = \{P \vee \neg Q, Q \vee \neg R\} & \text{if} \quad C = S \\ N^{\prec C} = \{P \vee \neg Q, Q \vee \neg R, P \vee Q \vee S\} & \text{if} \quad C = \neg S \end{array}$$

Definition 2.7.4 (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. Duplicate clauses are anyway eliminated quietly because the calculus operates on sets of clauses.

Note that for finite N, and any $C \in N$ redundancy $N^{\prec C} \models C$ can be decided but is as hard as testing unsatisfiability for a clause set N. So the goal is to invent redundancy notions that can be efficiently decided and that are useful.

C

Definition 2.7.5 (Selection Function)**.** The selection function sel maps clauses to one of its negative literals or $\bot$. If $\mathrm{sel}(C) = \neg P$ then $\neg P$ is called *selected* in C. If $\mathrm{sel}(C) = \bot$ then no literal in C is *selected*.

The selection function is, in addition to the ordering, a further means to restrict superposition inferences. If a negative literal is selected on a clause, any superposition inference must be on the selected literal.

Definition 2.7.6 (Partial Model Construction)**.** Given a clause set N and an ordering $\prec$ we can construct a (partial) Herbrand model $N_{\mathcal{I}}$ for N inductively as follows:

$$
\begin{array}{rcl}
N_C & := & \bigcup_{D \prec C} \delta_D \\[1em]
\delta_D & := & \begin{cases} \{P\} & \text{if } D = D' \vee P, P \text{ strictly maximal, no literal} \\ & \text{selected in } D \text{ and } N_D \not\models D \\ \emptyset & \text{otherwise} \end{cases} \\
N_{\mathcal{I}} & := & \bigcup_{C \in N} \delta_C
\end{array}
$$

Clauses C with $\delta_C \neq \emptyset$ are called *productive*.

Proposition 2.7.7. Some properties of the partial model construction.

1. For every D with $(C \vee \neg P) \prec D$ we have $\delta_D \neq \{P\}$.
2. If $\delta_C = \{P\}$ then $N_C \cup \delta_C \models C$.
3. If $N_C \models D$ and $D \prec C$ then for all C' with $C \prec C'$ we have $N_{C'} \models D$ and in particular $N_{\mathcal{I}} \models D$.
4. There is no clause C with $P \vee P \prec C$ such that $\delta_C = \{P\}$.

Please properly distinguish: N is a set of clauses interpreted as the conjunction of all clauses. $N^{\prec C}$ is of set of clauses from N strictly smaller than C with respect to $\prec$. $N_{\mathcal{I}}$, N_C are sets of atoms, often called *Herbrand Interpretations*. $N_{\mathcal{I}}$ is the overall (partial) model for N, whereas N_C is generated from all clauses from N strictly smaller than C. Validity is defined by $N_{\mathcal{I}} \models P$ if $P \in N_{\mathcal{I}}$ and $N_{\mathcal{I}} \models \neg P$ if $P \notin N_{\mathcal{I}}$, accordingly for N_C.

Given some clause set N, the partial model $N_{\mathcal{I}}$ can be extended to a valuation $\mathcal{A}$ by defining $\mathcal{A}(N_{\mathcal{I}}) := N_{\mathcal{I}} \cup \{\neg P \mid P \notin N_{\mathcal{I}}\}$. For some Herbrand interpretation $N_{\mathcal{I}}$ (N_C) I define $N_{\mathcal{I}} \models \phi$ if $\mathcal{A}(N_{\mathcal{I}})(\phi) = 1$.

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

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

Factoring $(N \uplus \{C \lor P \lor P\}) \Rightarrow_{\text{SUP}} (N \cup \{C \lor P \lor P\} \cup \{C \lor P\})$
where (i) P is maximal in $C \lor P \lor P$ (ii) no literal is selected in $C \lor P \lor P$

Note that the superposition factoring rule differs from the resolution factoring rule in that it only applies to positive literals. Abstract redundancy can also be lifted to inferences, in the propositional case to Superposition Left applications. A Superposition Left inference

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

is redundant if either one of the clauses $C_1 \lor P, C_2 \lor \neg P$ is redundant, or if $N^{\prec C_2 \lor \neg P} \models C_1 \lor C_2$. For a Factoring inference, the conclusion $C \lor P$ makes the premise $C \lor P \lor P$, so it is sufficient to require that $C \lor P \lor P$ is not redundant in order to guarantee $C \lor P$ to be non-redundant.

Definition 2.7.8 (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 already contained in N.

Alternatively, saturation can be defined on the basis of redundant inferences. An superposition inference is called *redundant* if the inferred clause is redundant with respect to all clauses smaller than the maximal premise of the inference. Then a set N is saturated up to redundancy if all inferences from clauses from N are redundant.

Examples for specific redundancy rules that can be efficiently decided and are already well-known from the resolution calculus, Section 2.6, are

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

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

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

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

A clause C where Condensation is not applicable is called *condensed*.

Proposition 2.7.9. All clauses removed by Subsumption, Tautology Deletion, Condensation and Subsumption Resolution are redundant with respect to the kept or added clauses.

Corollary 2.7.10 (Soundness)**.** Superposition is sound.

Superposition is a refinement of resolution, so soundness is a consequence of the soundness part of Theorem 2.6.1.

Theorem 2.7.11 (Completeness)**.** If N is saturated up to redundancy and $\bot \notin N$ then N is satisfiable and $N_\mathcal{I} \models N$.

Proof. The proof is by contradiction. So I assume: (i) for any clause D derived by Superposition Left or Factoring from N that D is redundant, i.e., $N^{\prec D} \models D$, (ii) $\bot \notin N$ and (iii) $N_\mathcal{I} \not\models N$. Then there is a minimal, with respect to $\prec$, clause $C \vee L \in N$ such that $N_\mathcal{I} \not\models C \vee L$ and L is a selected literal in $C \vee L$ or no literal in $C \vee L$ is selected and L is maximal. This clause must exist because $\bot \notin N$.

The clause $C \vee L$ is not redundant. For otherwise, $N^{\prec C \vee L} \models C \vee L$ and hence $N_\mathcal{I} \models C \vee L$, because $N_\mathcal{I} \models N^{\prec C \vee L}$, a contradiction.

I distinguish the case L is a positive and no literal selected in $C \vee L$ or L is a negative literal. Firstly, assume L is positive, i.e., $L = P$ for some propositional variable P. Now if P is strictly maximal in $C \vee P$ then actually $\delta_{C \vee P} = \{P\}$ and hence $N_\mathcal{I} \models C \vee P$, a contradiction. So P is not strictly maximal. But then actually $C \vee P$ has the form $C'_1 \vee P \vee P$ and Factoring derives $C'_1 \vee P$ where $(C'_1 \vee P) \prec (C'_1 \vee P \vee P)$. Now $C'_1 \vee P$ is not redundant, strictly smaller than $C \vee L$, we have $C'_1 \vee P \in N$ and $N_\mathcal{I} \not\models C'_1 \vee P$, a contradiction against the choice that $C \vee L$ is minimal.

Secondly, let us assume L is negative, i.e., $L = \neg P$ for some propositional variable P. Then, since $N_\mathcal{I} \not\models C \vee \neg P$ we know $P \in N_\mathcal{I}$. So there is a clause $D \vee P \in N$ where $\delta_{D \vee P} = \{P\}$ and P is strictly maximal in $D \vee P$ and $(D \vee P) \prec (C \vee \neg P)$. So Superposition Left derives $C \vee D$ where $(C \vee D) \prec (C \vee \neg P)$. The derived clause $C \vee D$ cannot be redundant, because for otherwise either $N^{\prec D \vee P} \models D \vee P$ or $N^{\prec C \vee \neg P} \models C \vee \neg P$. So $C \vee D \in N$ and $N_\mathcal{I} \not\models C \vee D$, a contradiction against the choice that $C \vee L$ is the minimal false clause. □

So the proof actually tells us that at any point in time we need only to consider either a superposition left inference between a minimal false clause and a productive clause or a factoring inference on a minimal false clause.

The proof relies on the abstract redundancy notion and not on the specific redundancy rules introduced above. However, it also goes through on the basis of the concrete redundancy notions, see Exercise **??**.

According to Theorem 2.7.11 if a clause set N is saturated up to redundancy, the interpretation $N_\mathcal{I}$ is a model for N. This does not hold the other way round. If $N_\mathcal{I}$ is a model for N then N is not saturated, in general, see Exercise **??**.

I mentioned already that the abstract redundancy notion of superposition goes beyond the classical resolution reduction rules tautology deletion, subsumption, subsumption resolution and condensation. For example consider the clause set

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

with ordering $S \prec P \prec Q \prec R$. Then $N^{\prec P \vee Q \vee \neg R} \models P \vee Q \vee \neg R$, i.e., the clause $P \vee Q \vee \neg R$ is redundant and can be deleted. This deletion is not justified by any of the classical resolution reduction rules.

In practice, there is a tradeoff between the unsuccessful testing of a powerful redundancy notion and keeping redundant clauses. Already in propositional logic there are exponentially many resolution or superposition inferences possible for a clause set. Testing the abstract superposition redundancy notion requires exponential run time in the size of the clause set $N^{\prec C}$. Inferences generated with respect to the partial model operator $N_{\mathcal{I}}$ following the proof of Theorem 2.7.11 are provably non-redundant with respect to the abstract superposition redundancy notion. Actually, designing a propositional theorem proving algorithm following the proof of Theorem 2.7.11 results in a deterministic system, without any choices once the atom ordering $\prec$ is fixed. Unfortunately, the resulting system is not very powerful in practice, because it cannot adopt to the problem structure. Please recall that the minimal literals in the ordering are then always highly preferred in the resulting learned clauses.

A calculus nicely demonstrating the tradeoff between restricting inferences and a corresponding redundancy notion preserving completeness is *Lock Resolution* [12]. For lock resolution an ordering is given per literal occurrence in a clause by attaching an index to each individual literal. The literal with the maximal index in a clause is then the maximal literal in that clause. Similar to propositional superposition, inferences are restricted to maximal literals.

Lock Resolution $(N \uplus \{C_1 \lor P^i, C_2 \lor \neg P^j\}) \;\Rightarrow_{\text{LOCK}}\; (N \cup \{C_1 \lor P^i, C_2 \lor \neg P^j\} \cup \{C_1 \lor C_2\})$

where (i) i is a maximal index in $C_1 \lor P^i$ and (ii) j is a maximal index in $C_2 \lor \neg P^j$

Lock Factoring $(N \uplus \{C \lor P^i \lor P^j\}) \;\Rightarrow_{\text{LOCK}}\; (N \cup \{C \lor P^i \lor P^j\} \cup \{C \lor P^j\})$

where j is a maximal index in in $C \lor P^i \lor P^j$

The below Example 2.7.12 demonstrates that for lock resolution there is no compatible redundancy notion in the sense that even tautologies must not be removed.

Example 2.7.12 (Lock Resolution)**.** Consider the unsatisfiable clause set

$$\begin{array}{ll} P^1 \lor Q^2 & \neg P^3 \lor \neg Q^4 \\ \neg Q^5 \lor P^6 & Q^7 \lor \neg P^8 \end{array}$$

over propositional variables P, Q. There are only two possible lock resolution inferences between the two clauses in the first row and the two clauses in the second row, respectively. They lead to the two tautologies $P^1 \lor \neg P^3$ and $\neg Q^5 \lor Q^7$.

Still, lock resolution is complete. It is just that all the well-known redundancy criteria compatible with resolution or superposition are not compatible with lock resolution.