3.11.6 Definition (Lexicographical Path Ordering (LPO))

Let $\Sigma = (\mathcal{S}, \Omega, \Pi)$ be a signature and let $\succ$ be a strict partial ordering on operator symbols in Ω, called *precedence*. The *lexicographical path ordering* $\succ_{lpo}$ on $T(\Sigma, \mathcal{X})$ is defined as follows: if s, t are terms in $T_S(\Sigma, \mathcal{X})$ then $s \succ_{lpo} t$ iff

1. $t = x \in \mathcal{X}$, $x \in vars(s)$ and $s \neq t$ or
2. $s = f(s_1, \ldots, s_n)$, $t = g(t_1, \ldots, t_m)$ and
   - 2.1 $s_i \succeq_{lpo} t$ for some $i \in \{1, \ldots, n\}$ or
   - 2.2 $f \succ g$ and $s \succ_{lpo} t_j$ for every $j \in \{1, \ldots, m\}$ or
   - 2.3 $f = g$, $s \succ_{lpo} t_j$ for every $j \in \{1, \ldots, m\}$ and $(s_1, \ldots, s_n)(\succ_{lpo})_{lex}(t_1, \ldots, t_m)$.

3.11.7 Theorem (LPO Properties)

1. The LPO is a rewrite ordering.
2. LPO enjoys the subterm property, hence is a simplification ordering.
3. If the precedence $\succ$ is total on Ω then $\succ_{lpo}$ is total on the set of ground terms $T(\Sigma)$.
4. If Ω is finite then $\succ_{lpo}$ is well-founded.

3.11.9 Definition (The Knuth-Bendix Ordering)

Let $\Sigma = (\mathcal{S}, \Omega, \Pi)$ be a finite signature, let $\succ$ be a strict partial ordering *("precedence")* on Ω, let $w : \Omega \cup \mathcal{X} \to \mathbb{R}^+$ be a *weight function*, so that the following condition is satisfied: $w(x) = w_0 \in \mathbb{R}^+$ for all variables $x \in \mathcal{X}$; $w(c) \geq w_0$ for all constants $c \in \Omega$.

Then, the weight function w can be extended to terms recursively:

$$w(f(t_1, \ldots, t_n)) = w(f) + \sum_{1 \leq i \leq n} w(t_i)$$

3.11.9 Definition (The Knuth-Bendix Ordering Ctd.)

or alternatively

$$\sum w(t) = \sum_{x \in vars(t)} w(x) \cdot \#(x,t) + \sum_{f \in \Omega} w(f) \cdot \#(f,t)$$

where $\#(a,t)$ is the number of occurrences of a in t.
The *Knuth-Bendix ordering* $\succ_{kbo}$ on $T(\Sigma, \mathcal{X})$ induced by $\succ$ and admissible w is defined by: $s \succ_{kbo} t$ iff

1 $\#(x,s) \geq \#(x,t)$ for all variables x and $w(s) > w(t)$, or
2 $\#(x,s) \geq \#(x,t)$ for all variables x, $w(s) = w(t)$, and
   (a) $s = f(s_1, \ldots, s_m)$, $t = g(t_1, \ldots, t_n)$, and $f \succ g$, or
   (b) $s = f(s_1, \ldots, s_m)$, $t = f(t_1, \ldots, t_m)$, and $(s_1, \ldots, s_m)(\succ_{kbo})_{lex}(t_1, \ldots, t_m)$.

3.11.10 Theorem (KBO Properties)

1. The KBO is a rewrite ordering.
2. KBO enjoys the subterm property, hence is a simplification ordering.
3. If the precedence $\succ$ is total on Ω then $\succ_{kbo}$ is total on the set of ground terms $T(\Sigma)$.
4. If Ω is finite then $\succ_{kbo}$ is well-founded.

The LPO ordering as well as the KBO ordering can be extended to atoms in a straightforward way. The precedence $\succ$ is extended to Π. For LPO atoms are then compared according to Definition 3.11.6-2. For KBO the weight function w is also extended to atoms by giving predicates a non-zero positive weight and then atoms are compared according to terms.

Actually, since atoms are never substituted for variables in first-order logic, an alternative to the above would be to first compare the predicate symbols and let $\succ$ decide the ordering. Only if the atoms share the same predicate symbol, the argument terms are considered, e.g., in a lexicographic way and are then compared with respect to KBO or LPO, respectively.

First-Order Ground Superposition

Propositional clauses and ground clauses are essentially the same, as long as equational atoms are not considered. This section deals only with ground clauses and recalls mostly the material from Section 2.7 for first-order ground clauses. The main difference is that the atom ordering is more complicated, see Section 3.11.

From now on let N be a possibly infinite set of ground clauses.

3.12.1 Definition (Ground Clause Ordering)

Let $\prec$ be a strict rewrite ordering total on ground terms and ground atoms. Then $\prec$ can be lifted to a total ordering $\prec_L$ on literals by its multiset extension $\prec_{\text{mul}}$ where a positive literal $P(t_1, \ldots, t_n)$ is mapped to the multiset $\{P(t_1, \ldots, t_n)\}$ and a negative literal $\neg P(t_1, \ldots, t_n)$ to the multiset $\{P(t_1, \ldots, t_n), P(t_1, \ldots, t_n)\}$.

The ordering $\prec_L$ is further lifted to a total ordering on clauses $\prec_C$ by considering the multiset extension of $\prec_L$ for clauses.

3.12.2 Proposition (Properties of the Ground Clause Ordering)

1. The orderings on literals and clauses are total and well-founded.
2. Let C and D be clauses with $P(t_1, \ldots, t_n) = \mathrm{atom}(\max(C))$, $Q(s_1, \ldots, s_m) = \mathrm{atom}(\max(D))$, where $\max(C)$ denotes the maximal literal in C.
   (a) If $Q(s_1, \ldots, s_m) \prec_L P(t_1, \ldots, t_n)$ then $D \prec_C C$.
   (b) If $P(t_1, \ldots, t_n) = Q(s_1, \ldots, s_m)$, $P(t_1, \ldots, t_n)$ occurs negatively in C but only positively in D, then $D \prec_C C$.

Eventually, as I did for propositional logic, 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. For superposition, inferences are restricted to maximal literals with respect to $\prec$.

For a clause set N, I define $N^{\prec C} = \{D \in N \mid D \prec C\}$.

3.12.3 Definition (Abstract Redundancy)

A ground clause C is *redundant* with respect to a set of ground clauses 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.

3.12.4 Definition (Selection Function)

The selection function sel maps clauses to one of its negative literals or $\bot$. If $\mathrm{sel}(C) = \neg P(t_1, \ldots, t_n)$ then $\neg P(t_1, \ldots, t_n)$ 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 in a clause, any superposition inference must be on the selected literal.

3.12.5 Definition (Partial Model Construction)

Given a clause set N and an ordering $\prec$ we can construct a (partial) model $N_{\mathcal{I}}$ for N inductively as follows:

$$
\begin{array}{rcl}
N_C & := & \bigcup_{D \prec C} \delta_D \\[2em]
\delta_D & := & \begin{cases} \{P(t_1,\ldots,t_n)\} & \text{if } D = D' \vee P(t_1,\ldots,t_n), \\ & P(t_1,\ldots,t_n) \text{ strictly maximal, no literal} \\ & \text{selected in } D \text{ and } N_D \not\models D \\ \emptyset & \text{otherwise} \end{cases} \\
N_{\mathcal{I}} & := & \bigcup_{C \in N} \delta_C
\end{array}
$$

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

3.12.6 Proposition (Propertied of the Model Operator)

Some properties of the partial model construction.

1. For every D with $(C \lor \neg P(t_1, \ldots, t_n)) \prec D$ we have $\delta_D \neq \{P(t_1, \ldots, t_n)\}$.
2. If $\delta_C = \{P(t_1, \ldots, t_n)\}$ 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 $P(t_1, \ldots, t_n) \lor P(t_1, \ldots, t_n) \prec C$ such that $\delta_C = \{P(t_1, \ldots, t_n)\}$.

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 Herbrand interpretations (see Proposition 3.5.3).

$N_{\mathcal{I}}$ is the overall (partial) model for N, whereas N_C is generated from all clauses from N strictly smaller than C.

Superposition Left
$(N \uplus \{C_1 \vee P(t_1, \ldots, t_n), C_2 \vee \neg P(t_1, \ldots, t_n)\}) \ \Rightarrow_{\text{SUP}}$
$(N \cup \{C_1 \vee P(t_1, \ldots, t_n), C_2 \vee \neg P(t_1, \ldots, t_n)\} \cup \{C_1 \vee C_2\})$
where (i) $P(t_1, \ldots, t_n)$ is strictly maximal in $C_1 \vee P(t_1, \ldots, t_n)$
(ii) no literal in $C_1 \vee P(t_1, \ldots, t_n)$ is selected
(iii) $\neg P(t_1, \ldots, t_n)$ is maximal and no literal selected in $C_2 \vee \neg P(t_1, \ldots, t_n)$, or $\neg P(t_1, \ldots, t_n)$ is selected in $C_2 \vee \neg P(t_1, \ldots, t_n)$

Factoring $(N \uplus \{C \vee P(t_1, \ldots, t_n) \vee P(t_1, \ldots, t_n)\}) \ \Rightarrow_{\text{SUP}}$
$(N \cup \{C \vee P(t_1, \ldots, t_n) \vee P(t_1, \ldots, t_n)\} \cup \{C \vee P(t_1, \ldots, t_n)\})$
where (i) $P(t_1, \ldots, t_n)$ is maximal in $C \vee P(t_1, \ldots, t_n) \vee P(t_1, \ldots, t_n)$
(ii) no literal is selected in $C \vee P(t_1, \ldots, t_n) \vee P(t_1, \ldots, t_n)$

3.12.7 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.

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

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

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

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

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 $\mathrm{grd}(\Sigma, \mathrm{cnf}(\neg\phi))$.

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 $\chi \models \psi$.