

3.11 Orderings

Propositional superposition is based on an ordering on the propositional variables, Section 2.7. The ordering is total and well-founded. Basically, propositional variables correspond to ground atoms in first-order logic. This section generalizes the ideas of the propositional superposition ordering to first-order logic. In first-order logic the ordering has to also consider terms and variables and operations on terms like the application of a substitution.

Definition 3.11.1 (Σ -Operation Compatible Relation). A binary relation \sqsupset over $T(\Sigma, \mathcal{X})$ is called *compatible with Σ -operations*, if $s \sqsupset s'$ implies $f(t_1, \dots, s, \dots, t_n) \sqsupset f(t_1, \dots, s', \dots, t_n)$ for all $f \in \Omega$ and $s, s', t_i \in T(\Sigma, \mathcal{X})$.

Lemma 3.11.2 (Σ -Operation Compatible Relation). A relation \sqsupset is compatible with Σ -operations iff $s \sqsupset s'$ implies $t[s]_p \sqsupset t[s']_p$ for all $s, s', t \in T(\Sigma, \mathcal{X})$ and $p \in \text{pos}(t)$.

In the literature *compatible with Σ -operations* is sometimes also called *compatible with contexts*.

Definition 3.11.3 (Substitution Stable Relation, Rewrite Relation). A binary relation \sqsupset over $T(\Sigma, \mathcal{X})$ is called *stable under substitutions*, if $s \sqsupset s'$ implies $s\sigma \sqsupset s'\sigma$ for all $s, s' \in T(\Sigma, \mathcal{X})$ and substitutions σ . A binary relation \sqsupset is called a *rewrite relation*, if it is compatible with Σ -operations and stable under substitutions.

A *rewrite ordering* is then an ordering that is a rewrite relation.

Definition 3.11.4 (Subterm Ordering). The *proper subterm ordering* $s > t$ is defined by $s > t$ iff $s|_p = t$ for some position $p \neq \epsilon$ of s .

Definition 3.11.5 (Simplification Ordering). A rewrite ordering \succ over $T(\Sigma, \mathcal{X})$ is called *simplification ordering*, if it enjoys the *subterm property* $s \succ t$ implies $s > t$ for all $s, t \in T(\Sigma, \mathcal{X})$ of the same sort.

Definition 3.11.6 (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. $t = x \in \mathcal{X}$, $x \in \text{vars}(s)$ and $s \neq t$ or
2. $s = f(s_1, \dots, s_n)$, $t = g(t_1, \dots, t_m)$ and
 - (a) $s_i \succeq_{lpo} t$ for some $i \in \{1, \dots, n\}$ or
 - (b) $f \succ g$ and $s \succ_{lpo} t_j$ for every $j \in \{1, \dots, m\}$ or
 - (c) $f = g$, $s \succ_{lpo} t_j$ for every $j \in \{1, \dots, m\}$ and $(s_1, \dots, s_n) (\succ_{lpo})_{lex} (t_1, \dots, t_m)$.

Theorem 3.11.7 (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.

Example 3.11.8. Consider the terms $g(x)$, $g(y)$, $g(g(a))$, $g(b)$, $g(a)$, b , a . With respect to the precedence $g \succ b \succ a$ the ordering on the ground terms is $g(g(a)) \succ_{lpo} g(b) \succ_{lpo} g(a) \succ_{lpo} b \succ_{lpo} a$. The terms $g(x)$ and $g(y)$ are not comparable. Note that the terms $g(g(a))$, $g(b)$, $g(a)$ are all instances of both $g(x)$ and $g(y)$.

With respect to the precedence $b \succ a \succ g$ the ordering on the ground terms is $g(b) \succ_{lpo} b \succ_{lpo} g(g(a)) \succ_{lpo} g(a) \succ_{lpo} a$.

Definition 3.11.9 (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 $w : \Omega \cup \mathcal{X} \rightarrow \mathbb{R}^+$ be a *weight function*, so that the following admissibility 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:

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

or alternatively

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

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) $s = f(s_1, \dots, s_m)$, $t = g(t_1, \dots, t_n)$, and $f \succ g$, or

$$(b) \ s = f(s_1, \dots, s_m), \ t = f(t_1, \dots, t_m), \text{ and } (s_1, \dots, s_m) \succ_{kbo} \text{lex}(t_1, \dots, t_m).$$

- Theorem 3.11.10** (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 KBO ordering can be extended to contain unary function symbols with weight zero. This was motivated by completion of the group axioms, see Chapter 4.

Definition 3.11.11 (The Knuth-Bendix Ordering Extended). The additional requirements added to Definition 3.11.9 are

1. Extend w to $w : \Omega \cup \mathcal{X} \rightarrow \mathbb{R}_0^+$
2. If $w(f) = 0$ for some $f \in \Omega$ with $\text{arity}(f) = 1$, then $f \succeq g$ for all $g \in \Omega$.
3. As a first case to the disjunction of 3.11.9-2.
 - (a') $t = x, s = f^n(x)$ for some $n \geq 1$

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.