Let $Q = \ ?- G_1, G_2, \ldots, G_m$ be a query. We say that a clause $H : - B_1, \ldots, B_k$ applies to $Q$ if and only if $G_1$ (the leftmost goal) unifies with a variant of $H$.

The SLD-resolution step is illustrated below:

$$
\begin{align*}
\text{\textbullet } & Q = \ ?- G_1, \ldots, G_m \\
& \quad \quad \quad \quad \quad H' : - B_1', \ldots, B_k' \\
& \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \theta = mgu(G_1, H') \\
\text{\textbullet } & \ ?- (B_1', \ldots, B_k', G_2, \ldots, G_m)\theta \\
\text{\textbullet } & \text{variant of } H : - B_1, \ldots, B_k \\
\text{\textbullet } & \text{with fresh new variables}
\end{align*}
$$
For instance, consider the query \(- add(X, X, Y), add(Y, Z, s(s(0))) \) and the rule
\[
add(s(U), V, s(W)) :- add(U, V, W).
\]
The variant \(add(s(U_1), V_1, s(W_1))\) unifies with \(add(X, X, Y)\) with the most general unifier
\[
\theta_1 = [s(U_1)/X, s(W_1)/Y, s(U_1)/V_1]
\]
So the next query (the "SLD-resolvent") is
\[
?- add(U_1, s(U_1), W_1), add(s(W_1), Z, s(s(0)))
\]
Creating a variant, i.e., renaming variables in a clause with new (unused) variables, is important. For instance, if the query is \(- add(U, Y), add(Y, Z, s(s(0))) \) and we use the above clause without renaming variables, then we will fail since the unification problem
\[
\{add(U, U, Y) =? add(s(U), V, s(W))\}
\]
has no unifier.