1. A Variant-Based $EVB$-unification (Sousir) Algorithm

By the Completeness Theorem of Fokker Variant Narrowing (Lecture 8, pages 14-15), we know that if $(\Sigma, B, E)$ is a decomposition of $(\Sigma, EVB)$ and $B$ has a finitary $B$-unification algorithm, the set

$$\langle t \rangle_{E,B}^\Sigma = \{ (u, \eta \mid_{\text{var}(t)}) \mid (t \overset{\eta^*}{\rightarrow} u) \in FVN_{E,B} (t) \land u = u^1 \}$$

is a complete set of $E_B$-variants for any term $t \in T_{E \Sigma}(X)$.

In particular, since, by the lemma in page 9 of Lecture 6, we then also know that $(\Sigma^\Sigma, B, E^\Sigma)$ is a decomposition of $(\Sigma, B, E^\Sigma)$, we immediately get that

$$\langle u \equiv v \rangle_{E^\Sigma,B} = \{ (w, \eta \mid_{\text{var}(u \equiv v)}) \mid (u \equiv v \overset{\eta^*}{\rightarrow} w) \in FVN_{E^\Sigma,B} (u \equiv v) \land w = w^1 \}$$

is a complete set of $E_B$-variants for any $u \equiv v \in T_{E^\Sigma}(X)$.

But then, it follows immediately from the Completeness Lemma in pages 11-13 of Lecture 8, and the lemma in pages 10 of Lecture 6 that

$$\text{VarUnif}_{EVB} (u \equiv v) = \{ \eta \mid_{\text{var}(u \equiv v)} \mid (t \mid_{\eta} \overset{\eta^*}{\rightarrow} u \equiv v) \in \langle u \equiv v \rangle_{E^\Sigma,B} \}$$

is a complete set of $EVB$-unifiers of the equation $u \equiv v$.

We call each $\theta \in \text{VarUnif}_{EVB} (u \equiv v)$ a variant unifier of $u \equiv v$. 

Of course, since the unifiers \( O \in \text{VanUnif}_{EuB} (u=v) \) are computed by \textit{narrows}, if we denote by \( \text{NarUnif}_{EuB} (u=v) \) the set of unifiers computed by the unification semi-algorithm defined in Fig. 11 of Lecture 6, we always have:

\[
\text{VanUnif}_{EuB} (u=v) \subseteq \text{NarUnif}_{EuB} (u=v)
\]

So, what's the difference? The difference is \underline{huge} because:

(i) \( \text{VanUnif}_{EuB} (u=v) \) will typically be \underline{much smaller} than \( \text{NarUnif}_{EuB} (u=v) \). This might seem ludicrous to say when \( \text{VanUnif}_{EuB} (u=v) \) is infinite, but it is not, even so. That is typically computed at depth \( \leq k \) in the narrow tree will typically be \underline{much smaller} than the corresponding set.

(ii) The narrow tree, we will be able to generate \underline{many more} unifiers \underline{in the same amount of time}, by not exploring a huge amount of useless dead ends and redundant solutions. Additionally, \( \text{VanUnif}_{EuB} (u=v) \) and \( \text{NarUnif}_{EuB} (u=v) \) can both have cardinality \( \omega \), but the \underline{finite} set \( \text{VanUnif}_{EuB} (u=v) \) of unifiers, computed at depth \( \leq k \) in the narrow tree will typically be \underline{much smaller} than the corresponding set \( \text{NarUnif}_{EuB} (u=v) \).
(ii) \( \text{VanUnf}_{E \cup B}^f (u = v) \) will terminate with either failure of a finite complete set \( E \cup B \)-unifiers in a huge number of cases where \( \text{NamUnf}_{E \cup B}^f (u = v) \) will typically not terminate (including the case where \( \text{NamUnf}_{E \cup B}^f (u = v) = \emptyset \) but the semi-algorithm does not terminate).

Claim (ii) is a fundamental claim, but at the moment seems both vague and unjustified. Can we make good on it? Yes!

But we first need a lemma:

Lemma. Let \((\Sigma, B, \vec{E}^x)\) be an FVP decomposition of \((\Sigma, E \cup B)\) and assume \(B\) has a finitary \(B\)-unification algorithm (extensible by additional free function symbol). Then \((\Sigma^x, B, \vec{E}^x)\) is also FVP.

Proof. By the extensibility assumption, the finitary \(B\)-unification algorithm with symbols \(\Sigma\) extends to one with symbols \(\Sigma^x\) and in fact coincides with it for all \(\Sigma\) except the new \(E_{\Sigma^x}\) symbols. By the Completeness Theorem for FVP in Lecture 8, pgs. 14-15, we will be done if we prove that

\[
\langle f(x_1, \ldots, x_n) \rangle_{E = B}^x \quad \text{is finite for each } f \in \Sigma^x.
\]

But since for each \( f \in \Sigma\) we have \(\langle f(x_1, \ldots, x_n) \rangle_{E = B}^x = \langle f(x_1, \ldots, x_n) \rangle_{E, B}^x\) and \((\Sigma, B, \vec{E}^x)\) is FVP, we only need to prove that

\[
\langle f(x_1, \ldots, x_n) \rangle_{E = B}^x \quad \text{is finite for each } f \in \Sigma^x.
\]
\( \langle x \equiv y \rangle \) for each \( x : i, y : j \) with \( i \neq j \) in \( \Sigma \). But this follows trivially from the fact that the mapping tree for \( x \equiv y \) is finite, namely:

\[
\begin{align*}
\begin{cases}
{x \mapsto x''} \\
{y \mapsto x''} \\
{z \mapsto x''}
\end{cases}
\end{align*}
\]

This is so because the only rule that can \( B \)-unify at a non-variable position is \( x' : i \equiv x' : j \rightarrow \text{tt} \), and we have

\[
\operatorname{Unif}_B (x \equiv y, x' \equiv x'') = \left\{ \{x \mapsto x'', y \mapsto x'', x' \mapsto x''\} \right\}
\]

We just need to check that, indeed, this singleton \( B \)-unifier is in a complete set of \( B \)-unifiers. Let \( \Theta \) be any such \( B \)-unifier \( x' \), so that \( \Theta(x) \equiv \Theta(y) =_B \Theta(x') \equiv \Theta(x'') \).

Since \( \equiv \) is a free function symbol not involved in the axioms in \( B \), then holds if and only if

\[
\Theta(x) =_B \Theta(x'), \quad \text{and} \quad \Theta(y) =_B \Theta(x'').
\]

But then

\[
\Theta =_B \left( \{x \mapsto x'', y \mapsto x'', x' \mapsto x''\} \cup \left\{ x'' \mapsto \Theta(x) \cup \Theta\right|_{\operatorname{dom}(\Theta) - \{x, y, x''\}} \right)
\]

since we have:

\[
\begin{align*}
\begin{cases}
{\color{green}{x'} \mapsto \color{red}{\Theta(x')}} =_B \Theta(x) \leftarrow x'' \leftarrow x' \\
{\color{green}{x} \mapsto \Theta(x)} = \Theta(x) \leftarrow x'' \leftarrow x \\
{\color{green}{y} \mapsto \Theta(y)} =_B \Theta(y) \leftarrow x'' \leftarrow y \\
\operatorname{dom}(\Theta) - \{x, y, x''\} \ni \Theta(\varepsilon) =_B \Theta(x) \leftarrow x' \leftarrow \varepsilon
\end{cases}
\end{align*}
\]

show that \( (\Sigma, B, E, \vec{E}) \) is FVP, as desired. \( \square \)
Here is now the main theorem about variant unification:

**Termination Theorem.** Let \( (\Sigma, B, \vec{E}) \) be a decomposition of \( (\Sigma, EUB) \) and assume \( B \) has a unification algorithm for \( \Sigma \)-terms extensible with free function symbols. Then \( \text{VarUnif}_{EUB}^\Sigma (u = v) \) terminates with either failure or a finite set of \( EUB \)-unifiers iff \( u \equiv_B v \) has a finite complete set of \( \vec{E}_\equiv^\Sigma_B \)-variants.

**Proof.** Obviously, \( \text{VarUnif}_{EUB}^\Sigma (u = v) \) terminates iff \( FVN_{\vec{E}_\equiv^\Sigma_B} (u = v) \) terminates iff (by the Completeness Theorem in Lecture 8, pp. 14-15) \( u \equiv_B v \) has a finite complete set of variants. \(\square\)

**Corollary.** Let \( (\Sigma, B, \vec{E}) \) be as in the above Termination Theorem. Then \( \text{VarUnif}_{EUB}^\Sigma (u = v) \) terminates for all inputs \( u = v \) iff \( (\Sigma, B, \vec{E}) \) is in FVP.

**Proof.** \((\Leftarrow)\) Follows from the Termination Theorem and the Lemma in \( p.3 \). We can prove \((\Rightarrow)\) by contradiction. Assume that \( (\Sigma, B, \vec{E}) \) is not FVP. By the Completeness Theorem in Lecture 8, pages 14-15 this means that there is
a term $t \in T_\Sigma(X)$ such that $\langle t \rangle_{E, \beta}$ is an infinite complete set of variants and no finite subset of it is complete set of variants. But this can only happen if $\text{vars}(t) \neq \emptyset$. Let $\phi$ rename all the variables in $\text{vars}(t)$ by fresh new ones. By the above Termination Theorem, we will be done if we show that $t \equiv t_\phi$ does not have a finite complete set of variants. Suppose it did. Since we can compute them by renaming and $t$ and $t_\phi$ share no variables, they must be of the form

$$(u_1 = u_1', \theta_1 \cup \theta_1'), \ldots, (u_m = u_m', \theta_m \cup \theta_m'),$$

$$(tt, \gamma_1), \ldots, (tt, \gamma_m)$$

with $n \geq 1$, $m \geq 1$, and $(\omega_i, \theta_i)$, resp. $(\omega_i', \theta_i')$, $E, \beta$-variants of $t$ resp. $t_\phi$. But this is impossible, because we obviously can find

$$(w, \beta), (w, \beta') \in \langle t \rangle_{E, \beta},$$

such that $\text{ran}(\alpha) \cap \text{ran}(\beta') = \emptyset$,

$$(u_i, \theta) \not\in_B (w, \alpha), 1 \leq i \leq m, (u_i', \theta_i') \not\in_B (w, \beta')$$
But this shows that \((u_i = u_i', \theta; \theta') \neq (v = w, \alpha \cup \beta)\) violates the assumption that \(t \equiv b\) had a finite set of variants. □

In summary, therefore, the variant unification semi-algorithm

\[
\text{VarUnf}_{EB}(u = v)
\]

terminates for all inputs and therefore becomes a \textit{finitary} \(EB\)-unification algorithm if and only if \((\Sigma, B, \widehat{E})\) is FVP.

From the satisfiability point of view this then gives us:

**Theorem.** Let \((\Sigma, B, \widehat{E})\) be an FVP decomposition of \((\Sigma, \text{EUB})\) and let \(B\) have a finitary unification algorithm extensible with free function symbols. Then assume furthermore that \(\Sigma\) has non-empty roots. Then

satisfiability of a quantifier GF formula \(\forall \in QF^+(\Sigma)\) in:

\[
\begin{align*}
1. & \; T_{\Sigma}(X) \\
2. & \; T_{\Sigma}(EUB)
\end{align*}
\]

is decidable. □
One last, useful remark is the following:

**Lemma.** Let \((\Sigma, B, \tilde{E})\) be FVP with \(B\) have a finitary \(B\)-unification algorithm extensible by free function symbols. Then \(\text{VarUnif}^0(u = v) = \{\theta \mid (tt, \theta) \in \{ u \equiv v \}_{\Sigma, B}\} \supset \text{E}, \beta\}^\beta\) is a complete set of most general unifiers.

**Proof.** We already know that \(\text{E}\) is a finite complete set of \(E\)-unifiers, because \(\text{VarUnif}(u = v)\) is obtained from \(\{ u \equiv v \}\) and \(\{ u \equiv v \}\) is \(\exists \beta\) \(\{ u \equiv v \}\).

But we cannot have \(\theta, \theta' \in \text{VarUnif}^0(u = v)\) with \(\theta \not\frac{\beta}{\beta} \theta'\) without \(\theta = \theta'\), since then we would have \((tt, \theta) \not\frac{\beta}{\beta} (tt, \theta')\) without \(\theta = \theta'\) in \(\{ u \equiv v \}_{\Sigma, B}\), which is impossible. \(\square\)

Note that when \((\Sigma, B, \tilde{E})\) is FVP and \(B\) any combination of axoc, axum, axdo, axon, mancl, varify, combd, compute, exactly \(\text{VarUnif}^0(u = v)\).