Unification and Matching in Church’s
Original Lambda Calculus

Ullrich Hustadt

MPI-I-92-219 June 1992
Author’s Address

Ullrich Hustadt
Max-Planck-Institut für Informatik
Im Stadtwald
W-6600 Saarbrücken
Germany
E-mail: Ullrich.Hustadt@mpi-sb.mpg.de
Abstract

In current implementations of higher-order logics higher-order unification is used to lift the resolution principle from the first-order case to the higher-order case. Higher-order matching is the core of implementations of higher-order rewriting systems and some systems for program transformation.

In this paper I argue that Church’s original lambda calculus, called non-forgetful lambda calculus, is an appropriate basis for higher-order matching. I provide two correct and complete algorithms for unification in the non-forgetful lambda calculus. Finally, I show how these unification algorithms can be used for matching in the non-forgetful lambda calculus.

Keywords
Non-forgetful lambda-calculus, Permissive unification algorithm, Lawful unification algorithm, Higher-order matching
1 Introduction

In first-order term algebra we interpret a term $M$ containing variables as the set of all ground instances $\sigma(M)$ of $M$.

Example 1.1 Given the following signature $\Sigma_1$

<table>
<thead>
<tr>
<th>sorts</th>
<th>$T_1, T_2, T_3$</th>
</tr>
</thead>
<tbody>
<tr>
<td>constants</td>
<td>$a_1, a_2 : T_1$</td>
</tr>
<tr>
<td>constants</td>
<td>$b_1, b_2 : T_2$</td>
</tr>
<tr>
<td>constant</td>
<td>$f : (T_1 \times T_2 \rightarrow T_3)$</td>
</tr>
<tr>
<td>variable</td>
<td>$X : T_1$</td>
</tr>
<tr>
<td>variable</td>
<td>$Y : T_2$</td>
</tr>
</tbody>
</table>

the term $f(X, b_1)$ can be interpreted as the set $\{f(a_1, b_1), f(a_2, b_1)\}$.

If we consider higher-order term algebras now, the interpretation of a term $M$ to be the set of all ground instances of $M$, which we want to denote $\text{GI}(M)$, gives surprising results.

Example 1.2 Suppose $F$ is a variable of type $(T_1 \times T_2 \rightarrow T_3)$ in signature $\Sigma_1$. Then $F(X, Y)$ can be interpreted as the set of all terms of type $T_3$, i.e.

$$\text{GI}(F(X, Y)) = \{f(a_1, b_1), f(a_2, b_1), f(a_1, b_2), f(a_2, b_2)\}.$$ 

On the other hand, $F(X, b_1)$ has the same interpretation

$$\text{GI}(F(X, b_1)) = \{f(a_1, b_1), f(a_2, b_1), f(a_1, b_2), f(a_2, b_2)\}.$$ 

So there is no difference between $\text{GI}(F(X, b_1))$ and $\text{GI}(F(X, Y))$, although intuitively $F(X, b_1)$ is more specific than $F(X, Y)$.

We want to interpret a term $f(T_1, \ldots, T_m)$ as the set of all ground instances $\sigma(f(T_1, \ldots, T_m))$, where the terms $\sigma(T_1)$, $\ldots$, $\sigma(T_m)$ actually occur in $\sigma(f(T_1, \ldots, T_m))$. We will denote this set by $\text{RGI}(M)$ for a term $M$. This results in a simple restriction to the allowed substitutions $\sigma$: If $\sigma(X) = \lambda x_i : T_n. N$ for some variable $X$, then all the variables in the binder must occur in the matrix $N$, i.e. $x_i \in \text{FV}(N)$ for all $i$, $1 \leq i \leq n$. So $\text{RGI}(F(X, b_1)) = \{f(a_1, b_1), f(a_2, b_1)\}$, for example.

This change of interpretation has impacts on our understanding of unification and matching substitutions. On the other hand, it is not possible to restrict the set of all substitutions to obey the restriction if we want to use some standard algorithm for unification and matching.

Example 1.3 Given the following signature

<table>
<thead>
<tr>
<th>sort</th>
<th>$T$</th>
</tr>
</thead>
<tbody>
<tr>
<td>constants</td>
<td>$a, b :$</td>
</tr>
<tr>
<td>constant</td>
<td>$f : (T \rightarrow (T \rightarrow T))$</td>
</tr>
<tr>
<td>variable</td>
<td>$F : (T \rightarrow (T \rightarrow T))$,</td>
</tr>
</tbody>
</table>

we consider the unification problem $U_1 = \{F(a, b) \doteq f(a, g(a, b))\}$. Using the transformation system for higher-order unification given in Snyder and Gallier (1989), we have the following transformation sequences

$$\{F(a, b) \doteq f(a, g(a, b))\}$$

Imitation rule using

$\rho_1 = \{F/\lambda x_1, x_2 : F(H_1(x_1, x_2), H_2(x_1, x_2))\}$

$\{f(H_1(a, b), H_2(a, b)) \doteq f(a, g(a, b))\}$

Decomposition rule
\[ \{ H_1(a, b) \doteq a, H_2(a, b) \doteq g(a, b) \} \]

Imitation rule using
\( \sigma_1 = \{ H_1 / \lambda y_1, y_2, a \} \)

Projection rule using
\( \sigma_2 = \{ H_i / \lambda y_1, y_2, y_1 \} \)

\( \{ a \doteq a, H_2(a, b) \doteq g(a, b) \} \)

Trivial removal

\( \{ H_2(a, b) \doteq g(a, b) \} \)

Imitation rule using
\( \tau = \{ H_2 / \lambda z_1, z_2, g(K_1(z_1, z_2), K_2(z_1, z_2)) \} \)

\( \{ g(K_1(a, b), K_2(a, b)) \doteq g(a, b) \} \)

Decomposition rule

\( \{ K_1(a, b) \doteq a, K_2(a, b) \doteq b \} \)

Imitation rule using
\( \phi_1 = \{ K_1 / \lambda y_1, y_2, a \} \)

Projection rule using
\( \phi_2 = \{ K_1 / \lambda y_1, y_2, y_1 \} \)

\( \{ a \doteq a, K_2(a, b) \doteq b \} \)

Trivial removal

\( \{ K_2(a, b) \doteq b \} \)

Imitation rule using
\( \psi_1 = \{ K_2 / \lambda y_1, y_2, b \} \)

Projection rule using
\( \psi_2 = \{ K_2 / \lambda y_1, y_2, y_2 \} \)

\( \{ b \doteq b \} \)

Trivial removal

\( \{ \} \)
The set of all unifiers of $U_1$ is

\[
\{\psi_i \circ \phi_j \circ \sigma_k \circ \rho_1 \mid 1 \leq i \leq 2 \land 1 \leq j \leq 2 \land 1 \leq k \leq 2\} = \\
\{\{F / \lambda x_1, x_2, f(x_1, g(x_1, x_2))\}, \{F / \lambda x_1, x_2, f(x_1, g(a, x_2))\}, \\
\{F / \lambda x_1, x_2, f(x_1, g(x_1, b))\}, \{F / \lambda x_1, x_2, f(x_1, g(a, x_2))\}, \\
\{F / \lambda x_1, x_2, f(a, g(x_1, x_2))\}, \{F / \lambda x_1, x_2, f(a, g(a, x_2))\}\}
\]

The set of those unifiers obeying our restriction is

\[
\{\psi_2 \circ \phi_2 \circ \sigma_2 \circ \rho_1, \psi_2 \circ \phi_1 \circ \tau_1 \circ \sigma_2 \circ \rho_1, \psi_2 \circ \phi_2 \circ \tau_1 \circ \sigma_1 \circ \rho_1,\} = \\
\{\{F / \lambda x_1, x_2, f(x_1, g(x_1, x_2))\}, \{F / \lambda x_1, x_2, f(a, g(x_1, x_2))\}, \{F / \lambda x_1, x_2, f(a, g(a, x_2))\}\}
\]

It is easy to see that neither $\sigma_1, \sigma_2, \phi_1, \phi_2, \psi_1$ nor $\psi_2$ obeys the restriction, only the appropriate compositions do.

In the following sections we will show that our restriction is necessary and sufficient to solve the problem. We will give a correct and complete transformation system for this context.

## 2 Non-forgetful Lambda-Calculus

### Definition 2.1 (Non-forgetful Terms)

Given a set $T_0$ of base types we define the set of types $T$ inductively as the smallest set containing $T_0$ and if $S, T \in T$ then $(S \rightarrow T) \in T$.

The set $RL^{-}(V, \Sigma)$ of raw terms is defined by the following abstract syntax

\[
RL^{-} = V \mid (\Sigma; T) \mid (V; T) \mid (RL^{-} \cdot RL^{-}) \mid \lambda V; T; RL^{-}
\]

where $V$ is a set of variables and $\Sigma$ a set of constants. We suppose $V, \Sigma$, and $T_0$ to be pairwise disjoint.

The set $L^{-}(V, \Sigma) \subseteq RL^{-}(V, \Sigma)$ of well-typed terms is defined using the following inference rules:

**Bound variable:**

\[
\frac{x : T \in \Gamma}{\Gamma \vdash x : T}, \quad \text{if } x \in V \land T \in T
\]

**Free variable:**

\[
\frac{\Gamma \vdash F : T : T}{\Gamma \vdash F : T : T}, \quad \text{if } F \in V \land T \in T
\]

**Constant:**

\[
\frac{\Gamma \vdash (c : T) : T}{\Gamma \vdash (c : T) : T}, \quad \text{if } c : T \in \Sigma \land T \in T
\]

**Application:**

\[
\frac{\Gamma \vdash M : (S \rightarrow T) \quad \Gamma \vdash N : S}{\Gamma \vdash (M \cdot N) : T}
\]

**Abstraction:**

\[
\frac{\Gamma \oplus x : S \vdash M : T}{\Gamma \vdash \lambda x : S ; N : (S \rightarrow T)}
\]

The set $L^{-}(V, \Sigma)$ is the set of all $M \in RL^{-}(V, \Sigma)$ such that $\epsilon \vdash M : T$ can be deduced for some $T \in T$. The function type $\cdot : L^{-} \rightarrow T$ is defined as $\text{type}(M) = T \iff \epsilon \vdash M : T$.

We simply write $L^{-}$ if $V$ and $\Sigma$ are obvious from the context.

The set of all free variables is $FV$. With a set $Z$ of free variables we associate the set $\text{symbols}(Z) \subseteq V$ of all variables occurring in $Z$. The set of free variables of a term $M$ with respect to $V$ is $FV_{V}(M)$. The set of bound variables of a term $M$ with respect to $V$ is $BV_{V}(M)$. If the set of variables $V$ is obvious from the context we write $FV(M)$ or $BV(M)$ for simplicity.

Church (1941) defines a $\lambda$-term $M$ to be well-formed iff

\[
\triangle
\]

Church (1941) defines a $\lambda$-term $M$ to be well-formed iff
• $M$ is either a constant or a variable,
• $M$ is an application of the form $(M_1 \cdot M_2)$ and $M_1$ and $M_2$ are well-formed,
• $M$ is an abstraction of the form $\lambda x : T . M_1$ such that $M_1$ is well-formed and contains at least an occurrence of $x$ in the scope of this binder.

We will use the phrase non-forgetful for these terms instead. The set of all such terms is $L_{nf}$. The terms in $L \rightarrow L_{nf}$ are called forgetful.

The set of all free variables of a term $M$ is $FV(M)$. The rules of lambda conversion are defined as usual. It is important to note that the set of non-forgetful $\lambda$-terms is closed under $\alpha$, $\beta$, and $\eta$-conversion. The $\beta$-normal form of a term $M$ is denoted $M \downarrow$, its $\eta$-expanded form is denoted $\eta[M]$. The set of all terms in $\eta$-expanded form is $L_{\eta}$. The set of non-forgetful substitutions $\mathcal{SUB}(L_{nf})$ is the set of all substitutions $\sigma : V \rightarrow L_{nf}$. This set is closed under composition and union of substitutions. The substitutions in $\mathcal{SUB}(L_{nf} V) \setminus \mathcal{SUB}(L_{nf} V)$ are called forgetful.

### Definition 2.2 (Substitution)
A substitution $\sigma$ in $L^\rightarrow$ is a mapping $\sigma : V \rightarrow L^\rightarrow$ such that the domain of $\sigma$, defined $\text{DOM}(\sigma) = \{ x \in V | \sigma(x) \neq x \}$, is finite. The identity substitution is denoted $\iota$. The set of all substitutions in $L^\rightarrow V$ is denoted $\mathcal{SUB}(L_{nf} V)$.

The domain of a substitution $\sigma$ is

$$\text{DOM}(\sigma) = \{ F \in FV | \sigma_2(F) \neq F \}.$$ 

The set of variables introduced by $\sigma$ is

$$\text{I}(\sigma) = \bigcup_{F \in \text{DOM}(\sigma)} FV(\sigma(F)).$$

A substitution $\sigma$ is normalized if $\sigma(F) = \sigma(F) \downarrow_\beta$ for all $F \in \text{DOM}(\sigma) \cap FV$.

Any substitution $\sigma$ can be uniquely extended to a mapping $\hat{\sigma} : L^\rightarrow \rightarrow L^\rightarrow$. The composition of two substitutions $\sigma$ and $\tau$ is written $\tau \circ \sigma$ and defined $\tau \circ \sigma(x) = \hat{\tau}(\sigma(x))$ for all $x \in V$. The union of $\sigma$ and $\tau$, denoted $\sigma \cup \tau$, is defined by

$$\sigma \cup \tau(x) = \begin{cases} 
\sigma(x), & \text{if } x \in \text{DOM}(\sigma) \\
\tau(x), & \text{if } x \in \text{DOM}(\tau) \\
x, & \text{otherwise}
\end{cases}$$

The set of non-forgetful substitutions $\mathcal{SUB}(L_{nf} V)$ is the set of all substitutions $\sigma : V \rightarrow L_{nf}$. This set is closed under composition and union of substitutions. The substitutions in $\mathcal{SUB}(L_{nf} V) \setminus \mathcal{SUB}(L_{nf} V)$ are called forgetful.

The restriction $\sigma|_V$ of a substitution $\sigma$ and the equality $\sigma = \tau|_V$ of two substitutions $\sigma$ and $\tau$ over a set of variables $V$ is defined in the usual way.

### Lemma 2.3
Let $\sigma$ be a forgetful substitution and $\tau$ some arbitrary substitution. Then the composition $\tau \circ \sigma$ is again a forgetful substitution.

### 3 Unification Problems in Lambda-Calculus
We use the following representation of unification problems.

### Definition 3.1 (Unification problems in $L^\rightarrow$)
An equation in $L^\rightarrow$ is a multiset of terms $M$ and $N$ in $L_{\eta}^\rightarrow$ of the same type. We will use the notation $M \simeq N$ for equations. A system $D$ in $L^\rightarrow$ is a multiset of equations in $L^\rightarrow$. A unification problem in $L^\rightarrow$ is an ordered pair $(D, V)$, written $(D | V)$, such that $D$ is a system and $V$ is a set of variables.
For a system

\[ D = \{ M_1 \triangleright N_1, \ldots, M_n \triangleright N_n \} \]

we write \( D \downarrow \) instead of \( \{ \eta[M_1] \triangleright \eta[N_1], \ldots, \eta[M_n] \triangleright \eta[N_n] \} \). \( D \downarrow \) is unique up to renaming of bound variables.

**Definition 3.2 (Unifier)**

Let \( \text{SU} \) be a set of substitutions. A substitution \( \theta \) in \( \text{SU} \) is called *unifier in* \( \text{SU} \) of two terms \( M \) and \( N \) from \( \mathcal{L} \) if \( \theta(M) \overset{*}{\rightarrow}_{\beta} \theta(N) \) holds.

A substitution \( \theta \) in \( \text{SU} \) is a *unifier of a unification problem* \( \langle D \mid V \rangle \) in \( \text{SU} \) iff \( \theta_V \) is a unifier for every equation in \( D \).

If \( \text{SU} \) is the set of all normalized substitutions in \( \text{SU}(\mathcal{L} \rightarrow V) \) then the set of all unifiers of a unification problem \( U \) is denoted \( \text{SU}(U) \). If \( \text{SU} \) is the set of all normalized substitutions in \( \text{SU}(\mathcal{L} \rightarrow V) \) then a unifier is called *non-forgetful* and the set of all non-forgetful unifier of \( U \) is written \( \text{SU}_{nf}(U) \).

\[ \Box \]

**Definition 3.3 (Complete set of unifiers)**

Let \( \text{SU} \) be a set of substitutions, \( U \) be a unification problem and \( Z \) and a finite set of variables, called the set of *protected variables*. A set \( \text{CSU}(U)[Z] \) of substitutions in \( \text{SU} \) is a *complete set of unifiers for* \( U \) in \( \text{SU} \) separated on \( FV(U) \) away from \( Z \) if

1. \( \text{CSU}(U)[Z] \subseteq \text{SU}(U) \subseteq \text{SU} \)
2. \( \forall \phi \in \text{SU}(U)[Z]: \exists \theta \in \text{CSU}(U)[Z]: \theta \subseteq_{\beta} \phi[FV(U)]\)
3. \( \forall \theta \in \text{CSU}(U)[Z]: \text{DOM}(\theta) \subseteq FV(U) \) and \( \text{I}(\theta) \cap (Z \cup \text{DOM}(\theta)) = \emptyset \).

If \( Z \) is not significant, we drop the \( [Z] \). If \( \text{CSU}(U) \) consists of a single substitution we call this substitution a *most general unifier*.

Again if \( \text{SU} \) is the set of all normalized substitutions in \( \text{SU}(\mathcal{L} \rightarrow V) \) then a complete set of all unifiers of a unification problem \( U \) is denoted \( \text{CSU}(U) \). If \( \text{SU} \) is the set of all normalized substitutions in \( \text{SU}(\mathcal{L} \rightarrow V) \) then a complete set of all non-forgetful unifier of \( U \) is written \( \text{SU}_{nf}(U) \).

\[ \Box \]

**Definition 3.4 (Solved Form)**

An equation is in *solved form* in a unification problem \( U \) if it is in the form \( \eta[F] \triangleright N \), for some variable \( F \) which occurs only once in \( U \), and \( F \) and \( N \) have the same type. A system \( D \) is *solved* if each of its pairs is solved. A unification problem \( \langle D \mid V \rangle \) is *solved* if \( D \) is solved.

To a system \( S = \{ F_1 \triangleright N_1, \ldots, F_n \triangleright N_n \} \) in solved form we associate a substitution

\[ \{S\}^{\text{un}} = \{ F_1/N_1, \ldots, F_n/N_n \}. \]

This substitution is unique up to variable renaming. To a unification problem \( \langle D \mid V \rangle \) in solved form we associate a system

\[ \langle D \mid V \rangle|_{\text{VAR}} = \{ F \triangleright N \mid F \triangleright N \in D \land F \in V \}. \]

Then we can associate to a unification problem \( U \) in solved form the substitution \( \{U\}^{\text{un}} = \{U\}^{\text{un}}|_{\text{VAR}} \).

\[ \Box \]

**Lemma 3.5**

If

\[ U = \{ \{ F_1 \triangleright N_1, \ldots, F_n \triangleright N_n \} \mid V \} \]

is a unification problem in solved form and \( \{ F_1, \ldots, F_n \} \subseteq V \), then \( \{U\}^{\text{un}} \) is a \( \text{CSU}_{nf}(U)[W] \) for any \( W \) such that \( W \cap FV(F_1 \triangleright N_1, \ldots, F_n \triangleright N_n) = \emptyset \).

**Proof:** \( \{U\}^{\text{un}} \) is obviously a unifier of \( U \), so \( \{U\}^{\text{un}} \subseteq \text{SU}(U) \). If \( \theta \in \text{SU}(U) \) then \( \theta = \beta \circ \theta \) \( \{U\}^{\text{un}} \), since \( \theta(F_i) \overset{*}{\rightarrow}_{\beta} \theta(N_i) = \theta ([U^{\text{un}}(F_i))] \) for 1 \( \leq i \leq n \), and \( \theta(x) = \theta ([U^{\text{un}}(x))] \) otherwise. So \( \{U\}^{\text{un}} \leq_{\beta} \theta \) and \( \{U\}^{\text{un}} \leq_{\beta} \theta \) \( FV(U) \). Because \( \text{DOM}(\{U\}^{\text{un}}) = \{ F_1, \ldots, F_n \} \subseteq FV(U) \) and \( \{U\}^{\text{un}} \) is idempotent the third condition is also fulfilled.
4 The Permissive Unification Algorithm

Snyder and Gallier (1989) give a correct and complete transformation system $\mathcal{H}U$ for unification in $\mathcal{L}^+$. Our first approach to the problem of unification in $\mathcal{L}^+_n$ will be a slight modification of their transformation system.

The central notion for both transformation systems is the partial binding.

**Definition 4.1 (Partial binding)**
A partial binding of type $(\overline{A_n} \rightarrow A_0)$ is a term of the form

$$\lambda \overline{x_n} : A_n. a(\lambda y_{p_m} : B_{p_m}. H_m(\overline{x_n}, y_{p_m}))$$

for some atom $a$ of type $(\overline{B_m} \rightarrow A_0)$ and free variables $H_i$ of type $(\overline{A_i}, \overline{B_{p_i}} \rightarrow B_i)$ for all $i$, $1 \leq i \leq m$.

If $a$ is a constant or a free variable, the partial binding is called an imitation binding. If $a$ is a bound variable $x_i$ for some $i$, $1 \leq i \leq n$, then it is called an $i$th projection binding.

For a variable $F$, a partial binding $M$ is appropriate to $F$ if $\text{type}(F) = \text{type}(M)$.

The transformation system uses sequences of partial bindings to construct unifiers. Partial bindings allow to substitute a term for a free variable that is as much undetermined as possible and at least as determined as needed. We are using the fact that for any term $M$ there exists a partial binding $P$ and a substitution $\sigma$ such that $\sigma(P) \xrightarrow{\beta} M$. But this wouldn’t be true if $\sigma$ has to be a non-forgetful substitution.

**Example 4.2** Consider the term

$$M_1 = \lambda x_1 : A_1. f(a)$$

where $f$ is a constant of type $(A_1 \rightarrow A_1)$ and $a$ is a constant of type $(A_1 \rightarrow A_1)$. An appropriate partial binding is

$$P_1 = \lambda x_1 : A_1. f(\lambda z_1 : A_1. H_1(x_1, z_1))$$

where $H_1$ is a free variable of type $(A_1, A_1 \rightarrow A_1)$. It is not possible to find a non-forgetful substitution $\sigma$ such that $\sigma(P_1) = M_1$. This substitution has to satisfy $\sigma(\lambda z_1 : A_1. H_1(x_1, z_1)) \equiv a$, where . The reason is that the variable $x_1$ has to occur in $\sigma(\lambda z_1 : A_1. H_1(x_1, z_1))$, but it doesn’t occur in $a$.

But the $H_i$ are only auxiliary variables we are using to construct a instantiation for a free variable $F$ in the original unification problem we consider. So it doesn’t matter that we instantiate $H_i$ with a forgetful term as long as the instantiation for $F$ is non-forgetful.

For this reason we augment the system of equation $D$ we want to unify with the set of free variables in it. Now we can distinguish the variables which must be instantiated with non-forgetful terms from those which can be instantiated with arbitrary terms. In every step in the transformation sequence we will ensure that this restriction is obeyed. A unification problem $(D \mid V)$ such that the restriction of any unifier $\sigma$ of $(D \mid V)$ to $V$ is forgetful is called a forgetful unification problem. Otherwise it is called non-forgetful.

We define a predicate on unification problems which distinguishes non-forgetful unification problems from obviously forgetful ones.

**Definition 4.3 ((Non-)Forgetful unification problems)**
An equation $F \equiv M$ is forgetful with respect to $V$ if $F$ is an element of the set of variables $V$, $F$ does not occur in the free variables of $M$, and $M$ is a forgetful term. Otherwise an equation is called possibly non-forgetful with respect to $V$. A unification problem $(D \mid V)$ is forgetful if some equation $M \equiv N$ in $D$ is forgetful with respect to $V$. Otherwise it is possibly non-forgetful. A solved unification problem that is possibly non-forgetful is called non-forgetful.

**Lemma 4.4** If $U_1$ is non-forgetful, i.e. it is solved and possibly non-forgetful, then $\lbrack U_1 \rbrack^{\text{sub}}$ is a non-forgetful substitution.
Proof: Because $U_1$ is solved it has the form
\[ \langle F_1 \doteq M_1, \ldots, F_m \doteq M_m \mid V \rangle, \]
for some free variables $F_1, \ldots, F_m$ and some terms $M_1, \ldots, M_m$. Because $U_1$ is possibly non-forgetful there is no equation $F_i \doteq M_i$, $1 \leq i \leq m$, such that $M_i$ is a forgetful term. Without restriction of generality we can assume that $V = \{F_1, \ldots, F_n\}$ for some $n$, $0 \leq n \leq m$. Then $[U_n]^{U_1}$ is
\[ \{F_1/M_1, \ldots, F_n/M_n\} \]
and this substitution is non-forgetful.

**Definition 4.5 (Transformation system $\mathcal{HU}$)**
The following rules form the transformation system on unification problems given by Snyder and Gallier (1989).

**Trivial removal**
\[ \langle \{M \doteq M\} \cup D \mid V \rangle \Rightarrow \langle D \mid V \rangle \quad \mathcal{HU}_1 \]

**Decomposition**
\[ \langle \{\lambda \overline{x_k}: T_k. a(M_m) \doteq \lambda \overline{x_k}: T_k. a(N_m)\} \cup D \mid V \rangle \quad \mathcal{HU}_2 \]
\[ \langle \cup_{1 \leq i \leq m} \{\lambda \overline{x_k}: T_k. M_i \doteq \lambda \overline{x_k}: T_k. N_i\} \cup D \mid V \rangle, \]
where $a$ is an arbitrary atom.

**Variable elimination**
\[ \langle \{\lambda \overline{x_k}: T_k. F(\overline{x}) \doteq \lambda \overline{x_k}: T_k. N\} \cup D \mid V \rangle \quad \mathcal{HU}_3 \]
\[ \langle \{\lambda \overline{x_k}: T_k. F(\overline{x}) \doteq \lambda \overline{x_k}: T_k. N\} \cup \theta(D) \downarrow \mid V \rangle, \]
where
- $F$ is a variable, and
- $F \notin \text{FV}(\lambda \overline{x_k}: T_k. N)$ and $F \in \text{FV}(D)m$
- $\theta = \{F/\lambda \overline{x_k}: T_k. N\}$.

**Imitation**
\[ \langle \{\lambda \overline{x_k}: T_k. F(M_m) \doteq \lambda \overline{x_k}: T_k. a(N_n)\} \cup D \mid V \rangle \quad \mathcal{HU}_4a \]
\[ \langle \{F \doteq \lambda \overline{x_k}: T_k. F(M_m) \doteq \lambda \overline{x_k}: T_k. a(N_n)\} \cup \{F/P\}(D) \downarrow \mid V \rangle, \]
where
- $F$ is a free variable and $a$ is either a constant or a free variable not equal to $F$, and
- $P$ is a variant of a imitation binding appropriate to $F$ e.g. $P = \lambda \overline{y_m}. S_m. a(\overline{z_{p_n}}. R_{p_n}. H_n(\overline{y_m}, \overline{z_{p_n}})).$

**Projection**
\[ \langle \{\lambda \overline{x_k}: T_k. F(M_m) \doteq \lambda \overline{x_k}: T_k. a(N_n)\} \cup D \mid V \rangle \quad \mathcal{HU}_4b \]
\[ \langle \{F \doteq \lambda \overline{x_k}: T_k. F(M_m) \doteq \lambda \overline{x_k}: T_k. a(N_n)\} \cup \{F/P\}(D) \downarrow \mid V \rangle, \]
where
- $F$ is a free variable and $a$ a arbitrary atom,
- $P$ is a variant of a $i$th projection binding for $1 \leq i \leq m$, appropriate to the free variable $F$, that is, $P = \lambda \overline{y_m}. S_m. y_i(\overline{z_{p_n}}. R_{p_n}. H_q(\overline{y_m}, \overline{z_{p_n}}))$, and
Lemma 4.8

If $\sigma$ is non-forgetful.

If $\sigma$ is a variable, $F$ a transformation step $U \Rightarrow V$ and $U$ is possibly non-forgetful.

\[ \{\{\lambda x_k: T_k, F(M_m) \triangleq \lambda x_k: T_k, G(N_n)\} \cup D \mid V\} \]

\[ \{\{F \triangleq P, \lambda x_k: T_k, F(M_m) \triangleq \lambda x_k: T_k, G(N_n)\} \cup \{F/P\}(D) \downarrow \mid V\}, \]

where

- $F$ and $G$ are free variables and
- $P = \lambda y_m: S_m, a(\lambda \bar{x}_m: \mu_m, H_n(y_m, \bar{x}_m))$ is a variant of some arbitrary partial binding appropriate to the term $\lambda x_k: T_k, F(M_m)$ such that $a \neq F$ and $a \neq G$.

\[ \triangle \]

Definition 4.6 (Transformation system $\mathcal{PUL}$)

We obtain the transformation system $\mathcal{PUL}$ by adding the restriction that $U_1 \Rightarrow_{\mathcal{PUL}} U_2$ if $U_1 \Rightarrow_{\mathcal{UL}} U_2$ and $U_2$ is possibly non-forgetful.

\[ \triangle \]

4.1 Correctness and Completeness of $\mathcal{PUL}$

Theorem 4.7 (Correctness) If $U = (D \mid FV(D)) \Rightarrow_{\mathcal{PUL}} U'$, with $U'$ in solved form and possibly non-forgetful, then the substitution $[U'\uparrow]_{\mathcal{UL}}$ is a non-forgetful unifier of $U$.

Proof: Snyder and Gallier (1989) have shown that $\mathcal{HU}$ is correct, i.e. if $U' = (D' \mid V)$, where $V = FV(D)$, and $U \Rightarrow_{\mathcal{UL}} U'$ then $[D']_{\mathcal{UL}} \in SU(D)$. Because for any transformation step $U_1 \Rightarrow_{\mathcal{UL}} U_2$ there exists a transformation step $U_1 \Rightarrow_{\mathcal{UL}} U_2$ we can conclude that $[U'\uparrow]_{\mathcal{UL}} \in SU(U)$.

Now if $(D' \mid V)$ is solved and possibly non-forgetful then any equation in $D'$ has the form $F \triangleq M$, where $F$ is a variable, $D'$ does not occur anywhere else in $D'$, and $M$ is a non-forgetful term if $F$ is in $V$. So $[U'\uparrow]_{\mathcal{UL}}$ is non-forgetful.

Lemma 4.8 If $M = \lambda x_n: T_n, N$ is a forgetful term then exists no substitution $\sigma$ such that $\sigma(M) \downarrow$ is a non-forgetful term.

Proof: Because $M$ is forgetful some bound variable, let’s assume it is $x_i$, for some $i$, $1 \leq i \leq n$, does not occur in $N$. The substitution $\sigma$ instantiates free variables with some $\lambda$-terms but applied to $M$ it could never happen that the instantiation results in a new occurrence of the bound variable $x_i$. So $x_i$ does not occur free in the matrix of $\sigma(M)$ or its normal form.

Lemma 4.9 If $U_1$ is a forgetful unification problem then there exists no transformation $U_1 \Rightarrow_{\mathcal{UL}} U_2$ such that $U_2$ is possibly non-forgetful.

Proof: We consider each transformation rule in turn:

Trivial removal Trivial equations are non-forgetful, so the removal of a trivial equation doesn’t change forgetfulness.

Decomposition If either term in the decomposed equation is forgetful, it’s decomposition will be forgetful too.

Variable elimination If the equation $F \triangleq \lambda x_k: T_k, N$ is forgetful, the resulting unification problem is forgetful because this equation is preserved. If some equation in $D$ is forgetful then $\theta(D) \downarrow$ will be forgetful because of lemma 4.8.

Imitation If the equation $\lambda x_k: T_k, F(M_m) \triangleq \lambda x_k: T_k, a(N_n)$ is forgetful then the resulting unification problem is forgetful because this equation is preserved. If some equation in $D$ is forgetful then $\theta(D) \downarrow$ will be forgetful because of lemma 4.8.
Proof: The same argumentation as for the imitation rule holds.

Explosion If the equation \( \lambda x_k : T_k. F(M_m) \upharpoonright \lambda x_k : T_k. G(N_n) \) is forgetful then the resulting unification problem is forgetful because this equation is preserved. If some equation in \( D \) is forgetful then \( \theta(D) \downarrow \) will be forgetful because of lemma 4.8.

Lemma 4.10 If \( U_1 \) is a unification problem then in any transformation sequence

\[ U_1 \Rightarrow_{\mathcal{HU}} U_2 \Rightarrow_{\mathcal{HU}} \cdots \Rightarrow_{\mathcal{HU}} U_n \]

such that \( U_n \) is in solved form and \( [U_n]^{un} \) is a non-forgetful substitution each \( U_i \) is possibly non-forgetful for \( 1 \leq i \leq n \).

Proof: We will first show that \( U_n \) is possibly non-forgetful: Assume that \( U_n \) is forgetful. Then \( U_n \) has the form \( U_n = \langle \{ F_1 \uparrow = M_1, \ldots, F_m \uparrow = M_m \} \mid V \rangle \) and there is an equation \( F_i \uparrow = M_i \) for some \( i, 1 \leq i \leq m \), such that \( F_i \in V \) and \( M_i \) is a forgetful term. But then \( [U_n]^{un}(F_i) = M_i \) implies that \( [U_n]^{un} \) is a forgetful substitution in contradiction to our assumption. So \( U_n \) must be non-forgetful.

If \( U_i+1 \) is possibly non-forgetful then \( U_i \) must be possibly non-forgetful for all \( i, 1 \leq i \leq n - 1 \). Assume again the contrary, i.e. that \( U_i \) is forgetful. Then following lemma 4.9 there could be no transformation \( U_i \Rightarrow_{\mathcal{HU}} U_{i+1} \) such that \( U_{i+1} \) is non-forgetful.

So we can conclude that each \( U_i \) in the transformation sequence is possibly non-forgetful.

Theorem 4.11 (Completeness of \( \mathcal{HU} \)) Let \( U = \langle D \mid FV(D) \rangle \) be a unification problem. If \( \theta \in SU_{nf}(U) \), then there exists a sequence of transformations

\[ U = U_0 \Rightarrow_{\mathcal{HU}} U_1 \Rightarrow_{\mathcal{HU}} U_2 \Rightarrow_{\mathcal{HU}} \cdots \Rightarrow_{\mathcal{HU}} U_n \]

where \( U_n \) is in solved form and \( [U_n]^{un} \leq_{\beta} \theta[FV(U)] \).

Proof: From the completeness \( \mathcal{HU} \) we know that for \( \theta \in SU_{nf}(U) \), there exists a sequence of transformations

\[ U = U_0 \Rightarrow_{\mathcal{HU}} U_1 \Rightarrow_{\mathcal{HU}} U_2 \Rightarrow_{\mathcal{HU}} \cdots \Rightarrow_{\mathcal{HU}} U_n \]

where \( U_n \) is in solved form and \( [U_n]^{un} \leq_{\beta} \theta[FV(U)] \), i.e. there exists a substitution \( \sigma \) such that \( \sigma \circ [U_n]^{un} =_{\beta} \theta[FV(U)] \). Because \( \theta \) is non-forgetful and following lemma 2.3 \( [U_n]^{un} \) must be non-forgetful too. From lemma 4.10 we know that each \( U_i, 1 \leq i \leq n \), must be possibly non-forgetful. So there exists a transformation sequence

\[ U = U_0 \Rightarrow_{\mathcal{HU}} U_1 \Rightarrow_{\mathcal{HU}} U_2 \Rightarrow_{\mathcal{HU}} \cdots \Rightarrow_{\mathcal{HU}} U_n \].

5 The Lawful Unification Algorithm

If we want to use non-forgetful substitutions only, we have to modify the notion of partial bindings. Consider again the general form of a partial binding, i.e.

\[ P_1 = \lambda x_n : A_n. a(\lambda y_{p_m} : B_{p_m}. H_m(x_m, y_{p_m})). \]

Let’s assume that this partial binding is used in a transformation sequence resulting in a unifier \( \sigma \). Let \( \sigma(\lambda y_{p_j} : B_{p_j}. H_j(x_m, y_{p_j})) = M_j \) and let \( \{ x_{k_1,j}, \ldots, x_{k_{n_j},j} \} \), \( 1 \leq k_{1,j} < j_{2,j} < \cdots < j_{n_j,j} \leq n \), be the set of all \( x_i \) appearing in \( M_j \) for all \( j, 1 \leq j \leq m \). Then a binding of the form

\[ P_2 = \lambda x_n : A_n. a(\lambda y_{p_m} : B_{p_m}. H_m(x_{k_{m-1,m}, m}, y_{p_m})). \]

could be used instead of \( P_1 \) in a transformation sequence resulting in the unifier \( \sigma \). We will prove this conjecture after we defined the notion of selective partial bindings and the presentation of the lawful transformation system.
Definition 5.1 (Selective Partial Bindings)
A selective partial binding of type \( \bar{\alpha} \mapsto \bar{\beta} \) is a term of the form
\[
\lambda x_n : \bar{\alpha}, \theta \left( \lambda y_{p_m} : \bar{\beta}_{p_m}, H_m(x_{q_{m,n} : \bar{\gamma}_{m,n}}, y_{p_m}) \right)
\]
for some atom \( \alpha \) of type \( \bar{\beta} \) and free variables \( H_j \) of type \( \bar{A}_{j} \mapsto \bar{B}_{j} \) where \( 1 \leq k_{1,j} < k_{2,j} < \cdots < k_{q,j} \leq n \) for all \( j, 1 \leq j \leq m \).

If \( \alpha \) is a constant or a free variable, the selective partial binding is called an selective imitation binding.

For a variable \( F \), a selective partial binding \( M \) is appropriate to \( F \) if \( \text{type}(F) = \text{type}(M) \).
\( \triangle \)

Definition 5.2 (Transformation system \( \mathcal{LU} \))
The following rules form the lawful unification algorithm for unification in the non-forgetful lambda-calculus.

Trivial removal
\[
\langle \{ M \mapsto M \} \cup D, V \rangle \Rightarrow \langle D, V \rangle \quad \mathcal{LU}_1
\]

Decomposition
\[
\langle \{ \lambda \bar{x}_k : \bar{T}_k, a(M_m) \mapsto \lambda \bar{x}_k : \bar{T}_k, a(N_m) \} \cup D, V \rangle \downarrow
\]
\[
\langle D \downarrow \rangle \quad \mathcal{LU}_2
\]

Variable elimination
\[
\langle \{ \lambda \bar{x}_k : \bar{T}_k, F(\bar{x}_k) \mapsto \lambda \bar{x}_k : \bar{T}_k, N \} \cup D, V \rangle \downarrow
\]
\[
\langle \{ \lambda \bar{x}_k : \bar{T}_k, F(\bar{x}_k) \mapsto \lambda \bar{x}_k : \bar{T}_k, N \} \cup \theta(D) \downarrow, V \rangle, \quad \mathcal{LU}_3
\]

where
- \( F \) is a variable,
- \( F \notin \text{FV}(\lambda \bar{x}_k : \bar{T}_k, N) \) and \( F \in \text{FV}(D) \), and
- \( \theta = \{ F/\lambda \bar{x}_k : \bar{T}_k, N \} \).

Imitation
\[
\langle \{ \lambda \bar{x}_k : \bar{T}_k, F(M_m) \mapsto \lambda \bar{x}_k : \bar{T}_k, a(N_m) \} \cup D, V \rangle \downarrow
\]
\[
\langle F \mapsto P, \lambda \bar{x}_k : \bar{T}_k, F(M_m) \mapsto \lambda \bar{x}_k : \bar{T}_k, a(N_m) \} \cup \{ F/P \} \downarrow, V \rangle, \quad \mathcal{LU}_{ga}
\]

where
- \( F \) is a free variable and \( a \) is either a constant or a free variable not equal to \( F \), and
- \( P \) is a variant of a selective imitation binding appropriate to \( F \), e.g.
\[
P = \lambda y_{m'} : \bar{\gamma}_{m'}, a(\lambda \bar{z}_{p_n} : \bar{R}_{p_n}, H_n(y_{q_{m,n} : \bar{\gamma}_{m,n}}, \bar{\gamma}_{p_n})).
\]

Projection
\[
\langle \{ \lambda \bar{x}_k : \bar{T}_k, F(M_m) \mapsto \lambda \bar{x}_k : \bar{T}_k, a(N_m) \} \cup D, V \rangle \downarrow
\]
\[
\langle F \mapsto P, \lambda \bar{x}_k : \bar{T}_k, F(M_m) \mapsto \lambda \bar{x}_k : \bar{T}_k, a(N_m) \} \cup \{ F/P \} \downarrow, V \rangle, \quad \mathcal{LU}_{qb}
\]

falls
- \( F \) is a free variable and \( a \) an arbitrary atom,
- \( P \) is a variant of a selective \( i \)th projection binding for \( 1 \leq i \leq m \), appropriate to \( F \), that is, \( P = \lambda y_{m'} : \bar{\gamma}_{m'}, y_i(\lambda \bar{z}_{p_n} : \bar{R}_{p_n}, H_q(y_{q_{m,n} : \bar{\gamma}_{m,n}}, \bar{\gamma}_{p_n})) \), and
• head($M_i$) = $a$, if head($M_i$) is a constant.

**Explosion**

\[
\begin{align*}
\langle \{ \lambda x_k : T_k. F(M_m) \rangle & \vdash \lambda x_k : T_k. G(N_n) \} \cup D, V \rangle \\
\downarrow
\langle \{ F \triangleright P, \lambda x_k : T_k. F(M_m) \rangle & \vdash \lambda x_k : T_k. G(N_n) \} \cup \{ F/P \}(D) \downarrow, V \rangle,
\end{align*}
\]

where

- $F$ and $G$ are free variables and
- $P = \lambda y_m : S_m.a(\lambda z_p : R_p. H_n(y_{k_{xm}}, z_{p}))$ is a variant of some arbitrary selective partial binding appropriate to $F$ such that $a \neq F$ and $a \neq G$.

\[\square\]

### 5.1 Correctness and Completeness of $\mathcal{LU}$

**Lemma 5.3** If $U \rightarrow U'$ using transformation rules $\mathcal{LU}_1$ or $\mathcal{LU}_3$ then $SU_{nf}(U) = SU_{nf}(U')$.

**Proof:** Snyder and Gallier (1989) show in lemma 4.12 that if $D \rightarrow \mathcal{LU} D'$ using transformations $\mathcal{HU}_t$ and $\mathcal{HU}_g$, then $SU(D) = SU(D')$. Now $\mathcal{LU}_1$ and $\mathcal{LU}_3$ are exactly the same rules as $\mathcal{HU}_t$ and $\mathcal{HU}_g$, so we conclude $SU((D \mid V)) = SU((D' \mid V))$. This implies $SU_{nf}((D \mid V)) = SU_{nf}((D' \mid V))$.

**Theorem 5.4 (Correctness)** If $U$ is a unification problem in $\mathcal{L}^+$ and $U \rightarrow \mathcal{LU} U'$, with $U'$ in solved form and non-forgetful, then the substitution $[U']^{\tau}_{\mathcal{LU}}$ is a non-forgetful unifier of $U$.

**Proof:** The proof is the exact same as for the correctness of $\mathcal{HU}$.

**Lemma 5.5** If $M = \lambda x_n : T_n.a(M_m)$ is a non-forgetful term then there exists a variant of a selective partial binding $P$ and a non-forgetful substitution $\tau$ such that $\tau(P) \rightarrow_{\beta} M$.

**Proof:** We distinguish the following cases:

$m = 0$: Then $M$ itself is a selective partial binding. We let $P = M$ and $\tau = \iota$.

$m > 0$: Suppose $X_i = BV(M_i) \cap \{ x_1, \ldots, x_n \} = \{ x_{k_{1,i}}, \ldots, x_{k_{m,i}} \}$ with $1 \leq k_1 < k_2 < \cdots < k_{m,i} \leq n$. Let $P_i = \eta[H_i(x_{k_{1,i}}, \ldots, x_{k_{m,i}})]$, where $type(P_i) = type(M_i)$ and $H_i$ is a free variable of appropriate type, for all $i$, $1 \leq i \leq m$. Because $M$ is a non-forgetful term, we must have $\bigcup_{1 \leq i \leq m} X_i \cup \{ a \} = \{ x_1, \ldots, x_n \}$. Therefore $P = \lambda x_n : T_n.a(P_m)$ is a selective partial binding. Let $\tau = \{ H_1/\lambda x_{k_{1,1}}, M_1, \ldots, H_m/\lambda x_{k_{m,m}}, M_m \}$. $\tau$ is obviously non-forgetful and $\tau(P) \rightarrow_{\beta} M_i$, for each $i$, $1 \leq i \leq m$. Thus $\tau(P) \rightarrow_{\beta} M$.

**Lemma 5.6** If $\theta = \{ F/M \} \cup \theta'$ is a non-forgetful substitution then there exists a variant of a selective partial binding $P$ appropriate to $F$ and a non-forgetful substitution $\tau$ such that

\[
\begin{align*}
\theta &= \{ F/M \} \cup \tau \cup \theta'[DOM(\theta)] \\
&= \beta \tau \circ \{ F/P \} \cup \theta'[DOM(\theta)]
\end{align*}
\]

If $\theta$ is idempotent then $\theta'' = \{ F/M \} \cup \tau \cup \theta'$ is an idempotent, non-forgetful unifier of the equation $F \equiv P$.

**Proof:** Because $\theta$ is non-forgetful, the term $M$ must be non-forgetful and we can define $P$ and $\tau$ as in lemma 5.5. Because $P$ is a variant, we have $DOM(\tau) \cap DOM(\theta) = \emptyset$. Therefore the first equation holds. We have already shown $\tau(P) \rightarrow_{\beta} M$, so

\[
\begin{align*}
\{ F/M \} &= \{ F/M \} \cup \tau[DOM(\theta)] \\
&= \beta \tau \circ \{ F/P \}[DOM(\theta)]
\end{align*}
\]

If $\theta$ is idempotent and $DOM(\tau) \cap I(\theta) = \emptyset$ then $DOM(\theta'') \cap I(\theta'') = \emptyset$. Finally $\theta''(P) = \tau(P) \rightarrow_{\beta} \theta''(F)$ shows that $\theta''$ is a unifier of $F \equiv P$. 

11
distinguish the following three cases:

1. If $\theta$ is solved in $W$, then there exists some transformation $\langle u, v, \theta \rangle \Rightarrow \langle u', v', \theta' \rangle$

Definition 5.7 (Transformation system $\mathcal{LV}$)
We define the transformation system $\mathcal{LV}$ on pairs of unification problems and substitutions in the following way:

**Trivial removal**
\[
\langle\langle\{M \vdash M\} \cup D \mid V\rangle, \theta\rangle \Rightarrow \langle\langle D \mid V\rangle, \theta\rangle
\]

**Decomposition**
\[
\langle\langle\{\lambda x_k : T_k . a(M_m) \vdash \lambda x_k : T_k . a(N_m)\} \cup D \mid V\rangle, \theta\rangle
\]
\[
\langle\langle \cup_{1 \leq i \leq m} \{\lambda x_k : T_k . M_i \vdash \lambda x_k : T_k . N_i\} \cup D \mid V\rangle, \theta\rangle
\]

where $a$ is not a free variable in $\mathcal{DOM}(\theta)$.

**Variable elimination**
\[
\langle\langle\{\lambda x_k : T_k . F(x_k) \vdash \lambda x_k : T_k . N\} \cup D \mid V\rangle, \theta\rangle
\]
\[
\langle\langle\{\lambda x_k : T_k . F(x_k) \vdash \lambda x_k : T_k . N\} \cup \sigma(D) \mid V\rangle, \theta\rangle
\]

where
- $F$ is a variable,
- $F \notin FV(\lambda x_k : T_k . N)$ and $F \in FV(D)$, and
- $\sigma = \{F/\lambda x_k : T_k . N\}$.

**Selective partial binding**
\[
\langle\langle\{\lambda x_k : T_k . F(M_m) \vdash \lambda x_k : T_k . b(N_n)\} \cup D \mid V\rangle, \{F/Q\} \cup \theta\rangle
\]
\[
\langle\langle\{F \vdash P, \lambda x_k : T_k . F(M_m) \vdash \lambda x_k : T_k . b(N_n)\} \cup \{F/P\} \cup \theta\rangle \\downarrow \langle\langle\{F \vdash P, \lambda x_k : T_k . F(M_m) \vdash \lambda x_k : T_k . b(N_n)\} \cup \{F/P\} \cup \theta\rangle
\]

where
- $F$ is a free variable and the equation $\lambda x_k : T_k . F(M_m) \vdash \lambda x_k : T_k . a(N_n)$ is not solved,
- $Q = \lambda y_m . S_m . a(Q_n)$,
- $P$ is a variant of a selective partial binding appropriate to $F$, e.g.
- $P = \lambda y_m . S_m . a(\lambda z_m : R_m . H_m (y_m, z_m))$,
- and
- $\tau = \{H_1/\lambda y_m : S_m . Q_1, \ldots, H_n/\lambda y_m : S_m . Q_n\}$.

\[\triangle\]

Lemma 5.8 If $\theta \in SU_n(U)$ for some unification problem $U$ not in solved form and $W$ is a set of variables, then there exists some transformation $(U, \theta) \Rightarrow_{\mathcal{LV}} (U', \theta')$ such that

1. $\theta' = \theta' [W]$;
2. If $\theta$ is idempotent then $\theta'$ is an idempotent non-forgetful unifier of $U'$ and
3. $U \Rightarrow_{U' U'}$.

Proof: Because $U$ is not in solved form there exists an equation $M \vdash N$, that is not solved in $U$. We distinguish the following three cases:

1. If $M = N$ then we can use the transformation rule $\mathcal{LV}_1$. If head$(M)$ is not a free variable in $\mathcal{DOM}(\theta)$, we can use $\mathcal{LV}_2$ too.
2. If head$(M) = \text{head}(N) \notin \text{DOM}(\theta)$ then we can use $\mathcal{LV}_2$.

3. Otherwise $M \neq N$ and we have one of the following two cases:
   
   (a) head$(M) \neq \text{head}(N)$: Because there exists a unifier for $U$ one of $M$ or $N$ must be flexible. We assume that $M$ is a flexible term.
   
   (b) head$(M) = \text{head}(N) \in \text{DOM}(\theta)$: In this case $M$ is flexible too.

Let $M = \lambda x_k : T_k. F(M_m)$ and $N = \lambda x_k : T_k. N'$. Then $\mathcal{LV}_4$ is applicable or if $M \rightarrow_{\varphi} F$ and $F \notin \text{FV}(N)$ the rule $\mathcal{LV}_3$ is applicable too.

Therefore there exists a transformation

$$\langle U, \theta \rangle \Longrightarrow_{\mathcal{LV}} \langle U', \theta' \rangle$$

If one of the rules $\mathcal{LV}_1$, $\mathcal{LV}_2$ or $\mathcal{LV}_3$ is applied, the requirements for $\theta'$ are fulfilled

1. Because $\theta = \theta'$;
2. Because of the correctness of $\mathcal{LU}$;
3. Because of the definition of $\mathcal{LV}$.

If the transformation rule $\mathcal{LV}_4$ is applied, we can assume, that $\theta = \{F/Q\} \cup \phi$. Following lemma 5.6 there exists a selective partial binding $P$ and a non-forgetful substitution $\tau$ such that

$$\text{DOM}(\tau) \cap W = \emptyset,$$

$$\theta' = \{F/Q\} \cup \tau \cup \phi =_{\beta} \tau \circ \{F/P\} \cup \phi,$$

$$U' = \{F/P\}(U) \cup \{F \psi = P\}.$$ 

The requirements for $\theta'$ are fulfilled

1. Because of the construction of $\theta'$;
2. We suppose $\text{DOM}(\theta) \cap I(\theta) = \emptyset$, so following lemma 5.6

$$\text{DOM}(\theta') \cap I(\theta') = \emptyset,$$

and

$$\theta'(P) = \tau(P) \rightarrow_{\varphi} Q = \theta'(F).$$

3. Because $U$ is unifiable, we can deduce from the applicability of $\mathcal{LV}_4$ the applicability of at least one of $\mathcal{LU}_{4a}$, $\mathcal{LU}_{4b}$, or $\mathcal{LU}_{4c}$:

- If head$(N)$ is not a variable in $\text{DOM}(\theta)$ and
  - if head$(Q) = \text{head}(N)$ then $U \rightarrow_{\mathcal{LU}_{4a}} U'$ or
  - if head$(Q) \neq \text{head}(N)$ then $U \rightarrow_{\mathcal{LU}_{4b}} U'$.

- If head$(N)$ is a variable then $U \rightarrow_{\mathcal{LU}_{4c}} U'$.

**Lemma 5.9** If $\theta \in \text{SU}_n(U)$ and no transformation applies to $\langle U, \theta \rangle$ then $U$ is in solved form.

**Theorem 5.10 (Completeness of $\mathcal{LU}$)** For any unification problem $U = \langle D \mid \text{FV}(D) \rangle$, if $\theta \in \text{SU}_n(U)$ then there exists a transformation sequence

$$U = U_0 \Longrightarrow_{\mathcal{LU}} U_1 \Longrightarrow_{\mathcal{LU}} \cdots \Longrightarrow_{\mathcal{LU}} U_n$$

such that $U_n$ is in solved form and $[U_n]_{\text{uni}} \leq_{\beta} \theta[\text{FV}(D)]$. 


Proof: It is easy to show that any sequence of \( \mathcal{L} \)-transformations terminates. Therefore we have for any unification problem \( U \) and unifier \( \theta \) of \( U \) a finite sequence

\[
(U, \theta) = (U_0, \theta_0) \xrightarrow{\mathcal{L}} (U_1, \theta_1)
\]

such that no further transformation is applicable. By induction over \( l \) using lemma 5.8 with \( W = \text{FV}(U) \) we have \( \theta = \theta_l[W] \) and \( \theta_l \in \text{SU}_{nf}(U_l) \). Furthermore there exists a corresponding transformation sequence

\[
U = U_0 \xrightarrow{\mathcal{U}} U_l.
\]

Using the previous lemma, \( U_l \) is in solved form. So we have \([U_l]_{\text{UN}} \leq_\beta \theta_l = \theta[W]\).

6 Matching

To avoid the decision which definition of ‘matching’ we want to deal with, we consider the problem of restricted unification instead. A \( V \)-restricted unification problem is just a unification problem in the sense of definition 3.1. But the definition of a unifier changes in the following way:

**Definition 6.1 (Unifier of a \( V \)-restricted unification problem)**

Let \( \text{SUB} \) be a set of substitutions. A substitution \( \theta \) in \( \text{SUB} \) is a **unifier of a \( V \)-restricted unification problem** \( \langle D \mid V \rangle \) in \( \text{SUB} \) iff \( \text{DOM}(\theta) \cap (\text{FV}(D) \setminus V) = \emptyset \) and \( \theta|_D \) is a unifier for every equation in \( D \).

If \( \text{SUB} \) is the set of all normalized substitutions in \( \text{SUB}(\mathcal{L}^\to) \) then the set of all unifiers of a \( V \)-restricted unification problem \( U \) is denoted \( \text{SU}^V(U) \). If \( \text{SUB} \) is the set of all normalized substitutions in \( \text{SUB}(\mathcal{L}_{nf}^\to) \) then a unifier is called **non-forgetful** and the set of all non-forgetful unifier of a \( U \) is written \( \text{SU}_{nf}^V(U) \).

Obviously we have yet considered the problem of \( \text{FV}(D) \)-restricted unification or **unrestricted unification**. For \( D = \{M_1 \doteq N_1, \ldots, M_n \doteq N_n\} \) the problem of \( \text{FV}(M_1, \ldots, M_n) \setminus \text{FV}(N_1, \ldots, N_n) \)-restricted unification will be called the problem of **matching** \( M_1, \ldots, M_n \) to \( N_1, \ldots, N_n \).

Restricted unification can be reduced to unrestricted unification by an enrichment of the set of constants \( \Sigma \) and a reduction of the set of variables \( V \) generating the set of \( \lambda \)-terms. For a \( V \)-restricted unification problem \( \langle D \mid V \rangle \) let \( V' = V \setminus \text{symbols}((\text{FV}(\mathcal{L}^\to) \setminus V) \cup (V \setminus \text{FV}(\mathcal{L}^\to))) \) and \( \Sigma' = \Sigma \cup \text{symbols}(\text{FV}(\mathcal{L}^\to) \setminus V) \). The free variables in \( \text{FV}(\mathcal{L}^\to) \setminus V \) occur in \( D \) but are not allowed to be instantiated by a \( V \)-restricted unifier. So they are added to the set of constants. Beside that the symbols of variables in \( V \setminus \text{FV}(\mathcal{L}^\to) \) are neither included in \( V' \) nor in \( \Sigma' \). This is necessary if we want to use \( \mathcal{U} \) as a complete transformation system for \( V \)-restricted unification as one can see in the proof of theorem 6.3.

This induces a bijection \( f \) between \( \mathcal{L}^\to(V, \Sigma) \) and \( \mathcal{L}^\to(V', \Sigma') \): \( f \) renames bound variables \( x:T \) such that \( x \in \text{symbols}(\text{FV}(\mathcal{L}^\to) \setminus V) \cup (V \setminus \text{FV}(\mathcal{L}^\to)) \) to some \( y:T \) with \( y \in V' \). So \( f(M) \) is only an \( \alpha \)-variant of \( M \).

Because we have considered \( \alpha \)-equivalence classes of terms all the time we will never use \( f \) explicitly in the following.

Of course, \( f \) can be lifted to a bijection from substitutions in \( \text{SUB}(\mathcal{L}^\to(V, \Sigma)\setminus(\text{FV}(\mathcal{L}^\to(D)\setminus V))) \) into substitutions in \( \text{SUB}(\mathcal{L}^\to(V', \Sigma')\setminus(\text{FV}(\mathcal{L}^\to(D)\setminus V'))) \) and between systems in \( \mathcal{L}^\to(V, \Sigma) \) and systems in \( \mathcal{L}^\to(V', \Sigma') \).

**Theorem 6.2 (Correctness)** If \( U = \langle D \mid V \rangle \) is a \( V \)-restricted unification problem in \( \mathcal{L}^\to(V, \Sigma) \) and there exists a transformation sequence

\[
\langle D \mid \text{FV}(V') \rangle = U_0 \xrightarrow{\mathcal{U}} U_1 \xrightarrow{\mathcal{U}} \cdots \xrightarrow{\mathcal{U}} U_n
\]

in \( \mathcal{L}^\to(V', \Sigma') \) such that \( U_n \) is in solved form and possibly non-forgetful, then \([U_n]_{\text{UN}} \) is a \( V \)-restricted unifier of \( U \).

**Proof:** The correctness of \( \mathcal{U} \) implies that \([U_n]_{\text{UN}} \) is a unifier of \( U \). We have to show that it is \( V \)-restricted. Because all variables in \( \text{FV}(V') \setminus V \) are considered as constants in \( D \) the free variables of \( D \) are a subset of \( V \). Because \( V' \) and \( \text{FV}(V') \setminus V \) are disjoint no free variable in \( U_n \) will be in \( \text{FV}(V') \setminus V \). So the domain of \([U_n]_{\text{UN}} \) will be a subset of \( V \). So \([U_n]_{\text{UN}} \) is a \( V \)-restricted unifier of \( U \).
**Theorem 6.3 (Completeness)** If $U = \langle D | V \rangle$ is a $V$-restricted unification problem in $L^\rightarrow (V, \Sigma)$ and $\theta$ is a non-forgetful unifier of $U$ then there exists a transformation sequence

$$\langle D | V \rangle = U_0 \Rightarrow_\mathcal{LU} U_1 \Rightarrow_\mathcal{LU} \cdots \Rightarrow_\mathcal{LU} U_n$$

in $L^\rightarrow (V', \Sigma')$ such that $U_n$ is in solved form and possibly non-forgetful and $[U_n]^{\text{un}} \leq \theta [V]$.

**Proof:** Because $\theta$ is a unifier of the $V$-restricted unification problem $\langle D | V \rangle$ it is also a unifier of the unification problem $\langle D | FV_{V'}(D) \rangle$. Because of the completeness of $\mathcal{LU}$ there exists a transformation sequence

$$\langle D | FV_{V'}(D) \rangle = U_0' \Rightarrow_\mathcal{LU} \langle D_n | FV_{V'}(D) \rangle = U_n'$$

in $L^\rightarrow (V', \Sigma')$ such that $U_n'$ is in solved form and possibly non-forgetful. Because $\mathcal{LU}$ ignores the set of variables associated with the unification problem there is also a transformation sequence

$$\langle D | V \rangle = U_0' \Rightarrow_\mathcal{LU} \langle D_n | V \rangle = U_n$$

We have $[U_n']^{\text{un}} \leq \theta [FV_{V'}(D)]$.

Now we have $FV_{V'}(D) \subseteq V$. So we have to explain why we expect $[U_n]^{\text{un}} \leq \theta [V]$ to hold. Of course, this is true if $(\text{DOM}([U_n]^{\text{un}}) \cup \text{COD}([U_n]^{\text{un}})) \cap (V \setminus FV_{V'}(D)) = \emptyset$.

Let us carefully reconsider the completeness proof for the transformation system $\mathcal{LU}$ using the system $\mathcal{LV}$. We can see that in neither step

$$\langle \langle D_i | V \rangle, \theta_i \rangle \Rightarrow_\mathcal{LV} \langle \langle D_{i+1} | V \rangle, \theta_{i+1} \rangle$$

a variable not in $\text{DOM}(\theta_i)$ is instantiated. Furthermore $\text{DOM}(\theta_{i+1}) \setminus \text{DOM}(\theta_i) \subseteq V'$. Because variables in $V \setminus FV_{V'}(D)$ are not in $FV'$ and $V \setminus FV_{V'}(D) = V \setminus FV_{V'}(D)$, no variable in $V \setminus FV_{V'}(D)$ will be instantiated by $[U_n]^{\text{un}}$. So we have at least $[U_n]^{\text{un}} = [U_n']^{\text{un}} [V]$.

For the same reason, no variable $x \in (V \setminus FV_{V'}(D))$ is an element of $\text{COD}([U_n]^{\text{un}})$. Because they are neither in $V'$ nor in $\Sigma'$, they will not be introduced in some transformation step

$$\langle \langle D_i | V \rangle, \theta_i \rangle \Rightarrow_\mathcal{LV} \langle \langle D_{i+1} | V \rangle, \theta_{i+1} \rangle.$$  

So they will not occur in $U_n$ and we have $[U_n]^{\text{un}} \leq \theta [V]$.

### 7 Future Work

The question wether or not higher-order matching is decidable in the simply typed lambda calculus is open. It is obvious that neither the matching algorithm based on $\mathcal{PU}$ nor the algorithm based on $\mathcal{LU}$ provides a decision procedure for matching in the non-forgetful lambda calculus.

**Example 7.1** Given the signature

<table>
<thead>
<tr>
<th>sort</th>
<th>$T_1$</th>
</tr>
</thead>
<tbody>
<tr>
<td>constants</td>
<td>$a : T_1$</td>
</tr>
<tr>
<td>variable</td>
<td>$F : ((T_1 \rightarrow T_1) \rightarrow T_1)$</td>
</tr>
</tbody>
</table>

we consider the second-order matching problem

$$M_0 = \langle \{ F : T_1 (\lambda x : T_1 . x) \doteq a : T_1 \} \ | \ F \rangle$$

In $\mathcal{LU}$ as well as in $\mathcal{PU}$ it is possible to use the project rule. We apply the substitution

$$\sigma = \{ F : T_1 / \lambda y : (T_1 \rightarrow T_1) . y(H_1 : ((T_1 \rightarrow T_1) \rightarrow T_1)(y)) \}$$
to $M_0$ resulting in the matching problem

$$M_1 = \langle \{ H_1 : T_1(\lambda x : T_1.x) \equiv a : T_1 \} \mid F \rangle.$$

$M_1$ is identical to $M_0$ up to renaming of the free variable. So the projection rule can be applied again and again resulting in an infinite branch in the search space.

Nevertheless, the use of the non-forgetful lambda calculus puts a strong restriction on the unifiers of a $V$-restricted unification problem. In the above example it is easy to see that there exists no matching substitution for $M_0$. So it could be possible that it is easier to provide a decision procedure for matching in the non-forgetful lambda calculus than to provide such a procedure for the simply typed lambda calculus (if either of them exists).

References
