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 Σ_1

$$
\begin{array}{ll}
\text{sorts} & T_1, T_2, T_3 \\
\text{constants} & a_1, a_2 : T_1 \\
\text{constants} & b_1, b_2 : T_2 \\
\text{constant} & f : (T_1 \times T_2 \to T_3) \\
\text{variable} & X : T_1 \\
\text{variable} & Y : T_2
\end{array}
$$

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 GI(M), gives surprising results.

Example 1.2 Suppose F is a variable of type $(T_1 \times T_2 \to T_3)$ in signature Σ_1. Then $F(X, Y)$ can be interpreted as the set of all terms of type T_3, i.e.

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

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

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

So there is no difference between $\mathrm{GI}(F(X, b_1))$ and $\mathrm{GI}(F(X, Y))$, 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 $\sigma(T_1)$, ..., $\sigma(T_m)$ actually occur in $\sigma(f(T_1, \ldots, T_m))$. We will denote this set by $\mathrm{RGI}(M)$ for a term M. This results in a simple restriction to the allowed substitutions σ: If $\sigma(X) = \lambda \overline{x_n \colon T_n}. N$ for some variable X, then all the variables in the binder must occur in the matrix N, i.e. $x_i \in FV(N)$ for all i, $1 \leq i \leq n$. So $\mathrm{RGI}(F(X, b_1)) = \{f(a_1, b_1), f(a_2, b_1)\}$, 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

$$
\begin{array}{ll}
\text{sort} & T \\
\text{constants} & a, b : \\
\text{constant} & f : (T \to (T \to T)) \\
\text{variable} & F : (T \to (T \to T)),
\end{array}
$$

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

$$
\begin{array}{c}
\{F(a, b) \stackrel{?}{=} f(a, g(a, b))\} \\
\Big\downarrow \begin{array}{c}\text{Imitation rule using} \\ \rho_1 = \{F/\lambda\, x_1, x_2.\, f(H_1(x_1, x_2), (H_2(x_1, x_2)))\}\end{array} \\
\{f(H_1(a, b), H_2(a, b)) \stackrel{?}{=} f(a, g(a, b))\} \\
\Big\downarrow \text{Decomposition rule}
\end{array}
$$

$$
\begin{tikzcd}[row sep=large]
\{H_1(a,b) \stackrel{?}{=} a, H_2(a,b) \stackrel{?}{=} g(a,b)\} \arrow[d, bend right=60, "\substack{\text{Imitation rule using}\\ \sigma_1=\{H_1/\lambda\, y_1,y_2.\, a\}}"'] \arrow[d, bend left=60, "\substack{\text{Projection rule using}\\ \sigma_2=\{H_1/\lambda\, y_1,y_2.\, y_1\}}"] \\
\{a \stackrel{?}{=} a, H_2(a,b) \stackrel{?}{=} g(a,b)\} \arrow[d, "\text{Trivial removal}"] \\
\{H_2(a,b) \stackrel{?}{=} g(a,b)\} \arrow[d, "\substack{\text{Imitation rule using}\\ \tau_1=\{H_2/\lambda\, z_1,z_2.\, g(K_1(z_1,z_2),K_2(z_1,z_2))\}}"] \\
\{g(K_1(a,b), K_2(a,b)) \stackrel{?}{=} g(a,b)\} \arrow[d, "\text{Decomposition rule}"] \\
\{K_1(a,b) \stackrel{?}{=} a, K_2(a,b) \stackrel{?}{=} b\} \arrow[d, bend right=60, "\substack{\text{Imitation rule using}\\ \phi_1=\{K_1/\lambda\, y_1,y_2.\, a\}}"'] \arrow[d, bend left=60, "\substack{\text{Projection rule using}\\ \phi_2=\{K_1/\lambda\, y_1,y_2.\, y_1\}}"] \\
\{a \stackrel{?}{=} a, K_2(a,b) \stackrel{?}{=} b\} \arrow[d, "\text{Trivial removal}"] \\
\{K_2(a,b) \stackrel{?}{=} b\} \arrow[d, bend right=60, "\substack{\text{Imitation rule using}\\ \psi_1=\{K_2/\lambda\, y_1,y_2.\, b\}}"'] \arrow[d, bend left=60, "\substack{\text{Projection rule using}\\ \psi_2=\{K_2/\lambda\, y_1,y_2.\, y_2\}}"] \\
\{b \stackrel{?}{=} b\} \arrow[d, "\text{Trivial removal}"] \\
\{\}
\end{tikzcd}
$$

The set of all unifiers of U_1 is

$$\{\psi_i \circ \phi_j \circ \tau_1 \circ \sigma_k \circ \rho_1 \mid 1 \leq i \leq 2 \wedge 1 \leq j \leq 2 \wedge 1 \leq k \leq 2\} = \\ \{\{F/\lambda\, x_1, x_2.\, f(x_1, g(x_1, x_2))\}, \{F/\lambda\, x_1, x_2.\, f(x_1, g(x_1, b))\}, \{F/\lambda\, x_1, x_2.\, f(x_1, g(a, x_2))\}, \\ \{F/\lambda\, x_1, x_2.\, f(x_1, g(a, b))\}, \{F/\lambda\, x_1, x_2.\, f(a, g(x_1, x_2))\}, \{F/\lambda\, x_1, x_2.\, f(a, g(x_1, b))\}, \\ \{F/\lambda\, x_1, x_2.\, f(a, g(a, x_2))\}, \{F/\lambda\, x_1, x_2.\, f(a, g(a, b))\}\}$$

The set of those unifiers obeying our restriction is

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

It is easy to see that neither σ_1, σ_2, ϕ_1, ϕ_2, ψ_1 nor ψ_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 $\mathcal{T}_0$ of *base types* we define the set of *types* $\mathcal{T}$ inductively as the smallest set containing $\mathcal{T}_0$ and if S, $T \in \mathcal{T}$ then $(S \to T) \in \mathcal{T}$.

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

$$\mathcal{RL}^{\to} = \mathcal{V} \mid (\Sigma{:}\,\mathcal{T}) \mid (\mathcal{V}{:}\,\mathcal{T}) \mid (\mathcal{RL}^{\to} \cdot \mathcal{RL}^{\to}) \mid \lambda\, \mathcal{V}{:}\,\mathcal{T}.\, \mathcal{RL}^{\to}$$

where $\mathcal{V}$ is a set of variables and Σ a set of constants. We suppose $\mathcal{V}$, Σ, and $\mathcal{T}_0$ to be pairwise disjoint.

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

Bound variable: $\dfrac{x{:}T \in \Gamma}{\Gamma \vdash x{:}T}$, if $x \in \mathcal{V} \wedge T \in \mathcal{T}$

Free variable: $\dfrac{}{\Gamma \vdash F{:}\,T{:}T}$, if $F \in \mathcal{V} \wedge T \in \mathcal{T}$

Constant: $\dfrac{}{\Gamma \vdash (c{:}\,T){:}T}$, if $c{:}\,T \in \Sigma \wedge T \in \mathcal{T}$

Application: $\dfrac{\Gamma \vdash M{:}(S \to T) \quad \Gamma \vdash N{:}S}{\Gamma \vdash (M \cdot N){:}T}$

Abstraction: $\dfrac{\Gamma \oplus x{:}S \vdash M{:}T}{\Gamma \vdash \lambda\, x{:}\, S.\, N{:}(S \to T)}$

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

We simply write $\mathcal{L}^{\to}$ if $\mathcal{V}$ and Σ are obvious from the context.

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

Church (1941) defines a λ-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 $\mathcal{L}^{\rightarrow}_{nf}$. The terms in $\mathcal{L}^{\rightarrow} \setminus \mathcal{L}^{\rightarrow}_{nf}$ 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 λ-terms is closed under α-, β-, and η-conversion. The β-normal form of a term M is denoted $M{\downarrow}$, its η-expanded form is denoted $\eta[M]$. The set of all terms in η-expanded form is $\mathcal{L}^{\rightarrow}_{\eta}$.

Definition 2.2 (Substitution)
A *substitution* σ in $\mathcal{L}^{\rightarrow}$ is a mapping $\sigma : \mathcal{V} \rightarrow \mathcal{L}^{\rightarrow}$ such that the domain of σ, defined $\mathcal{DOM}(\sigma) = \{x \in \mathcal{V} \mid \sigma(x) \neq x\}$, is finite. The identity substitution is denoted ι. The *set of all substitutions in* $\mathcal{L}^{\rightarrow}$ is denoted $\mathrm{SUB}(\mathcal{L}^{\rightarrow\mathcal{V}})$.

The *domain* of a substitution σ is

$$\mathcal{DOM}(\sigma) = \{F \in \mathcal{FV} \mid \sigma_2(F) \neq F\}.$$

The set of variables *introduced by* σ is

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

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

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

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

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

The restriction $\sigma_{|V}$ of a substitution σ and the equality $\sigma = \tau\,[V]$ of two substitutions σ and τ over a set of variables V is defined in the usual way. $\triangle$

Lemma 2.3 *Let σ be a forgetful substitution and τ 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 $\mathcal{L}^{\rightarrow}$)
An *equation in* $\mathcal{L}^{\rightarrow}$ is a multiset of terms M and N in $\mathcal{L}^{\rightarrow}_{\eta}$ of the same type. We will use the notation $M \stackrel{?}{=} N$ for equations. A *system* D *in* $\mathcal{L}^{\rightarrow}$ is a multiset of equations in $\mathcal{L}^{\rightarrow}$. A *unification problem in* $\mathcal{L}^{\rightarrow}$ is an ordered pair $\langle D, V\rangle$, written $\langle D \mid V\rangle$, such that D is a system and V is a set of variables. $\triangle$

For a system

$$D = \{M_1 \stackrel{?}{=} N_1, \dots, M_n \stackrel{?}{=} N_n\}$$

we write $D{\downarrow}$ instead of $\{\eta[M_1{\downarrow}] \stackrel{?}{=} \eta[N_1{\downarrow}], \dots, \eta[M_n{\downarrow}] \stackrel{?}{=} \eta[N_n{\downarrow}]\}$. $D{\downarrow}$ is unique up to renaming of bound variables.

Definition 3.2 (Unifier)
Let SUB be a set of substitutions. A substitution θ in SUB is called *unifier in* SUB of two terms M and N from $\mathcal{L}^{\rightarrow}_{\eta}$ iff $\theta(M) \stackrel{*}{\longleftrightarrow}_{\beta\eta} \theta(N)$ holds.

A substitution θ in SUB is a *unifier of a unification problem* $\langle D \mid V\rangle$ *in* SUB iff $\theta_{|V}$ is a unifier for every equation in D.

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

Definition 3.3 (Complete set of unifiers)
Let *SUB* 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 $\mathrm{CSU}(U)[Z]$ of substitutions in SUB is a *complete set of unifiers for* U *in* SUB *separated on* $FV(U)$ *away from* Z iff

(1) $\mathrm{CSU}(U)[Z] \subseteq \mathrm{SU}(U) \subseteq \mathrm{SUB}$

(2) $\forall \phi \in \mathrm{SU}(\Gamma) \colon \exists \theta \in \mathrm{CSU}(U)[Z] \colon \theta \leq_{\beta} \phi\, [FV(U)]$

(3) $\forall \theta \in \mathrm{CSU}(U)[Z] \colon \mathcal{DOM}(\theta) \subseteq FV(U)$ and $\mathcal{I}(\theta) \cap (Z \cup \mathcal{DOM}(\theta)) = \emptyset$.

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

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

Definition 3.4 (Solved Form)
An equation is in *solved form* in a unification problem U if it is in the form $\eta[F] \stackrel{?}{=} N$, 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 $S = \{F_1 \stackrel{?}{=} N_1, \dots, F_n \stackrel{?}{=} N_n\}$ in solved form we associate a substitution

$$\lceil S \rceil^{\mathrm{SUB}} = \{F_1/N_1, \dots, F_n/N_n\}.$$

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

$$\langle D \mid V\rangle|_{\mathrm{VAR}} = \{F \stackrel{?}{=} N \mid F \stackrel{?}{=} N \in D \wedge F \in V\}.$$

Then we can associate to a unification problem U in solved form the substitution $\lceil U \rceil^{\mathrm{SUB}} = \lceil U|_{\mathrm{VAR}} \rceil^{\mathrm{SUB}}$. $\triangle$

Lemma 3.5 *If*

$$U = \langle \{F_1 \stackrel{?}{=} N_1, \dots, F_n \stackrel{?}{=} N_n\} \mid V\rangle$$

is a unification problem in solved form and $\{F_1, \dots, F_n\} \subseteq V$*, then* $\{\lceil U \rceil^{\mathrm{SUB}}\}$ *is a* $\mathrm{CSU}_{nf}(U)[W]$ *for any* W *such that* $W \cap FV(F_1 \stackrel{?}{=} N_1, \dots, F_n \stackrel{?}{=} N_n) = \emptyset$*.*

Proof: $\lceil U \rceil^{\mathrm{SUB}}$ is obviously a unifier of U, so $\{\lceil U \rceil^{\mathrm{SUB}}\} \subseteq \mathrm{SU}(U)$. If $\theta \in \mathrm{SU}(U)$ then $\theta =_{\beta} \theta \circ \lceil U \rceil^{\mathrm{SUB}}$, since $\theta(F_i) \stackrel{*}{\longleftrightarrow}_{\beta} \theta(N_i) = \theta(\lceil U \rceil^{\mathrm{SUB}}(F_i))$ for $1 \leq i \leq n$, and $\theta(x) = \theta(\lceil U \rceil^{\mathrm{SUB}}(x))$ otherwise. So $\lceil U \rceil^{\mathrm{SUB}} \leq_{\beta} \theta$ and $\lceil U \rceil^{\mathrm{SUB}} \leq_{\beta} \theta\, [FV(U)]$. Because $\mathcal{DOM}(\lceil U \rceil^{\mathrm{SUB}}) = \{F_1, \dots, F_n\} \subseteq FV(U)$ and $\lceil U \rceil^{\mathrm{SUB}}$ is idempotent the third condition is also fulfilled.

4 The Permissive Unification Algorithm

Snyder and Gallier (1989) give a correct and complete transformation system $\mathcal{HU}$ for unification in $\mathcal{L}^{\rightarrow}$. Our first approach to the problem of unification in $\mathcal{L}^{\rightarrow}_{nf}$ 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

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

for some atom a of type $(\overline{B_m} \rightarrow A_0)$ and free variables H_i of type $(\overline{A_n}, \overline{B_{p_i}} \rightarrow B_i)$ 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 $\mathrm{type}(F) = \mathrm{type}(M)$. $\triangle$

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 σ such that $\sigma(P) \xrightarrow{*}_{\beta} M$. But this wouldn't be true if σ has to be a non-forgetful substitution.

Example 4.2 Consider the term

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

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

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

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 σ such that $\sigma(P_1) = M_1$. This substitution has to satisfy $\sigma(\lambda\, z_1\colon A_1.\, H_1(x_1, z_1)) =_{\beta\eta} a$, where . The reason is that the variable x_1 has to occur in $\sigma(\lambda\, z_1\colon A_1.\, H_1(x_1, z_1))$, 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 $\langle D \mid V\rangle$ such that the restriction of any unifier σ of $\langle D \mid V\rangle$ 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 $F \stackrel{?}{=} M$ 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 $\langle D \mid V\rangle$ is *forgetful* if some equation $M \stackrel{?}{=} N$ 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.* $\triangle$

Lemma 4.4 *If* U_1 *is non-forgetful, i.e. it is solved and possibly non-forgetful, then* $\lceil U_1 \rceil^{\mathrm{SUB}}$ *is a non-forgetful substitution.*

Proof: Because U_1 is solved it has the form

$$\langle F_1 \stackrel{?}{=} M_1, \ldots, F_m \stackrel{?}{=} M_m \mid V\rangle,$$

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 $F_i \stackrel{?}{=} M_i$, $1 \leq i \leq m$, 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 $\lceil U_n \rceil^{\text{SUB}}$ 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

$$\langle \{M \stackrel{?}{=} M\} \cup D \mid V\rangle \Rightarrow \langle D \mid V\rangle \qquad \mathcal{HU}_1$$

Decomposition

$$\begin{array}{c}\langle\{\lambda \overline{x_k\colon T_k}.\, a(\overline{M_m}) \stackrel{?}{=} \lambda \overline{x_k\colon T_k}.\, a(\overline{N_m})\} \cup D \mid V\rangle \\ \Downarrow \\ \langle \cup_{1\leq i\leq m}\{\lambda \overline{x_k\colon T_k}.\, M_i \stackrel{?}{=} \lambda \overline{x_k\colon T_k}.\, N_i\} \cup D \mid V\rangle,\end{array} \qquad \mathcal{HU}_2$$

where a is an arbitrary atom.
Variable elimination

$$\begin{array}{c}\langle\{\lambda \overline{x_k\colon T_k}.\, F(\overline{x_k}) \stackrel{?}{=} \lambda \overline{x_k\colon T_k}.\, N\} \cup D \mid V\rangle \\ \Downarrow \\ \langle\{\lambda \overline{x_k\colon T_k}.\, F(\overline{x_k}) \stackrel{?}{=} \lambda \overline{x_k\colon T_k}.\, N\} \cup \theta(D)\!\downarrow \mid V\rangle,\end{array} \qquad \mathcal{HU}_3$$

where

- F is a variable, and
- $F \notin FV(\lambda \overline{x_k\colon T_k}.\, N)$ and $F \in FV(D)$m
- $\theta = \{F/\lambda \overline{x_k\colon T_k}.\, N\}$.

Imitation

$$\begin{array}{c}\langle\{\lambda \overline{x_k\colon T_k}.\, F(\overline{M_m}) \stackrel{?}{=} \lambda \overline{x_k\colon T_k}.\, a(\overline{N_n})\} \cup D \mid V\rangle \\ \Downarrow \\ \langle\{F \stackrel{?}{=} P, \lambda \overline{x_k\colon T_k}.\, F(\overline{M_m}) \stackrel{?}{=} \lambda \overline{x_k\colon T_k}.\, a(\overline{N_n})\} \cup \{F/P\}(D)\!\downarrow \mid V\rangle,\end{array} \qquad \mathcal{HU}_{4a}$$

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. $P = \lambda \overline{y_m\colon S_m}.\, a(\overline{\lambda \overline{z_{p_n}\colon R_{p_n}}.\, H_n(\overline{y_m}, \overline{z_{p_n}})})$.

Projection

$$\begin{array}{c}\langle\{\lambda \overline{x_k\colon T_k}.\, F(\overline{M_m}) \stackrel{?}{=} \lambda \overline{x_k\colon T_k}.\, a(\overline{N_n})\} \cup D \mid V\rangle \\ \Downarrow \\ \langle\{F \stackrel{?}{=} P, \lambda \overline{x_k\colon T_k}.\, F(\overline{M_m}) \stackrel{?}{=} \lambda \overline{x_k\colon T_k}.\, a(\overline{N_n})\} \cup \{F/P\}(D)\!\downarrow \mid V\rangle,\end{array} \qquad \mathcal{HU}_{4b}$$

where

- F is a free variable and a a arbitrary atom,
- P is a variant of a ith projection binding for $1 \leq i \leq m$, appropriate to the free variable F, that is, $P = \lambda \overline{y_m\colon S_m}.\, y_i(\overline{\lambda \overline{z_{p_q}\colon R_{p_q}}.\, H_q(\overline{y_m}, \overline{z_{p_q}})})$, and

- $\text{head}(M_i) = a$, if $\text{head}(M_i)$ is a constant.

Explosion

$$\begin{array}{c}\langle\{\lambda\,\overline{x_k\colon T_k}.\,F(\overline{M_m}) \stackrel{?}{=} \lambda\,\overline{x_k\colon T_k}.\,G(\overline{N_n})\} \cup D \mid V\rangle \\ \Downarrow \\ \langle\{F \stackrel{?}{=} P, \lambda\,\overline{x_k\colon T_k}.\,F(\overline{M_m}) \stackrel{?}{=} \lambda\,\overline{x_k\colon T_k}.\,G(\overline{N_n})\} \cup \{F/P\}(D)\!\downarrow \mid V\rangle,\end{array} \qquad \mathcal{HU}_{4c}$$

where

- F and G are free variables and
- $P = \lambda\,\overline{y_m\colon S_m}.\,a(\overline{\lambda\,\overline{z_{p_n}\colon R_{p_n}}.\,H_n(\overline{y_m}, \overline{z_{p_n}})})$ is a variant of some arbitrary partial binding appropriate to the term $\lambda\,\overline{x_k\colon T_k}.\,F(\overline{M_m})$ such that $a \neq F$ and $a \neq G$.

$\triangle$

Definition 4.6 (Transformation system $\mathcal{PU}$)
We obtain the transformation system $\mathcal{PU}$ by adding the restriction that $U_1 \Longrightarrow_{\mathcal{PU}} U_2$ iff $U_1 \Longrightarrow_{\mathcal{HU}} U_2$ and U_2 is possibly non-forgetful. $\triangle$

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

Theorem 4.7 (Correctness) *If $U = \langle D \mid FV(D)\rangle \stackrel{*}{\Longrightarrow}_{\mathcal{PU}} U'$, with U' in solved form and possibly non-forgetful, then the substitution $\lceil U' \rceil^{\text{SUB}}$ is a non-forgetful unifier of U.*

Proof: Snyder and Gallier (1989) have shown that $\mathcal{HU}$ is correct, i.e. if $U' = \langle D' \mid V\rangle$, where $V = FV(D)$, and $U \stackrel{*}{\Longrightarrow}_{\mathcal{HU}} U'$ then $\lceil D \rceil^{\text{SUB}} \in \text{SU}(D)$. Because for any transformation step $U_1 \Longrightarrow_{\mathcal{PU}} U_2$ there exists a transformation step $U_1 \Longrightarrow_{\mathcal{HU}} U_2$ we can conclude that $\lceil U' \rceil^{\text{SUB}} \in \text{SU}(U)$.

Now if $\langle D' \mid V\rangle$ is solved and possibly non-forgetful then any equation in D' has the form $F \stackrel{?}{=} M$, where F is a variable, F does not occur anywhere else in D', and M is a non-forgetful term if F is in V. So $\lceil U' \rceil^{\text{SUB}}$ is non-forgetful.

Lemma 4.8 *If $M = \lambda\,\overline{x_n\colon T_n}.\,N$ is a forgetful term then exists no substitution σ 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 σ instantiates free variables with some λ-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 $U_1 \Longrightarrow_{\mathcal{HU}} U_2$ 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 elemination If the equation $F \stackrel{?}{=} \lambda\,\overline{x_k\colon T_k}.\,N$ 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 $\lambda\,\overline{x_k\colon T_k}.\,F(\overline{M_m}) \stackrel{?}{=} \lambda\,\overline{x_k\colon T_k}.\,a(\overline{N_n})$ 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.

Projection The same argumentation as for the imitation rule holds.

Explosion If the equation $\lambda\,\overline{x_k\colon T_k}.\,F(\overline{M_m}) \stackrel{?}{=} \lambda\,\overline{x_k\colon T_k}.\,G(\overline{N_n})$ 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*

$$U_1 \Longrightarrow_{\mathcal{HU}} U_2 \Longrightarrow_{\mathcal{HU}} \cdots \Longrightarrow_{\mathcal{HU}} U_n$$

such that U_n is in solved form and $\lceil U_n \rceil^{\text{SUB}}$ 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 $U_n = \langle \{F_1 \stackrel{?}{=} M_1, \ldots, F_m \stackrel{?}{=} M_m\} \mid V\rangle$ and there is an equation $F_i \stackrel{?}{=} M_i$ for some i, $1 \leq i \leq m$, such that $F_i \in V$ and M_i is a forgetful term. But then $\lceil U_n \rceil^{\text{SUB}}(F_i) = M_i$ implies that $\lceil U_n \rceil^{\text{SUB}}$ 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 $U_i \Longrightarrow_{\mathcal{HU}} U_{i+1}$ 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 $\mathcal{PU}$) *Let $U = \langle D \mid FV(D)\rangle$ be a unification problem. If $\theta \in \mathrm{SU}_{nf}(U)$, then there exists a sequence of transformations*

$$U = U_0 \Longrightarrow_{\mathcal{PU}} U_1 \Longrightarrow_{\mathcal{PU}} U_2 \Longrightarrow_{\mathcal{PU}} \cdots \Longrightarrow_{\mathcal{PU}} U_n,$$

where U_n is in solved form and $\lceil U_n \rceil^{\text{SUB}} \leq_\beta \theta\,[FV(U)]$.

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

$$U = U_0 \Longrightarrow_{\mathcal{HU}} U_1 \Longrightarrow_{\mathcal{HU}} U_2 \Longrightarrow_{\mathcal{HU}} \cdots \Longrightarrow_{\mathcal{HU}} U_n,$$

where U_n is in solved form and $\lceil U_n \rceil^{\text{SUB}} \leq_\beta \theta\,[FV(U)]$,i.e. there exists a substitution σ such that $\sigma \circ \lceil U_n \rceil^{\text{SUB}} =_\beta \theta\,[FV(U)]$. Because θ is non-forgetful and following lemma 2.3 $\lceil U_n \rceil^{\text{SUB}}$ 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

$$U = U_0 \Longrightarrow_{\mathcal{PU}} U_1 \Longrightarrow_{\mathcal{PU}} U_2 \Longrightarrow_{\mathcal{PU}} \cdots \Longrightarrow_{\mathcal{PU}} U_n.$$

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.

$$P_1 = \lambda\,\overline{x_n\colon A_n}.\,a(\overline{\lambda\,\overline{y_{p_m}\colon B_{p_m}}.\,H_m(\overline{x_n}, \overline{y_{p_m}})}).$$

Let's assume that this partial binding is used in a transformation sequence resulting in a unifier σ. Let $\sigma(\lambda\,\overline{y_{p_j}\colon B_{p_j}}.\,H_j(\overline{x_n}, \overline{y_{p_j}})) = M_j$ and let $\{x_{k_{1,j}}, \ldots, x_{k_{q_j,j}}\}$, $1 \leq k_{1,j} < j_{2,j} < \cdots < j_{q_j,j} \leq n$, 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

$$P_2 = \lambda\,\overline{x_n\colon A_n}.\,a(\overline{\lambda\,\overline{y_{p_m}\colon B_{p_m}}.\,H_m(\overline{x_{k_{q_m,m}}}, \overline{y_{p_m}})}).$$

could be used instead of P_1 in a transformation sequence resulting in the unifier σ. 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 $(\overline{A_n} \to A_0)$ is a term of the form

$$\lambda\, \overline{x_n\colon A_n}.\, a(\overline{\lambda\, \overline{y_{p_m}\colon B_{p_m}}.\, H_m(\overline{x_{k_{q_m,m}}}, \overline{y_{p_m}})})$$

for some atom a of type $(\overline{B_m} \to A_0)$ and free variables H_j of type $(\overline{A_{j_{q_j}}}, \overline{B_{p_j}} \to B_j)$ where $1 \le k_{1,j} < k_{2,j} < \cdots < k_{q_j,j} \le n$ for all j, $1 \le j \le m$.

If a is a constant or a free variable, the selective partial binding is called an *selective imitation binding*; if a is a bound variable x_i for some i, $1 \le i \le n$, then it is called a *selective ith projection binding.*

For a variable F, a selective partial binding M is *appropriate to* F if type$(F) = $ type(M). △

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

$$\langle \{M \stackrel{?}{=} M\} \cup D, V\rangle \Rightarrow \langle D, V\rangle \qquad \mathcal{LU}_1$$

Decomposition

$$\begin{array}{c}\langle\{\lambda\, \overline{x_k\colon T_k}.\, a(\overline{M_m}) \stackrel{?}{=} \lambda\, \overline{x_k\colon T_k}.\, a(\overline{N_m})\} \cup D, V\rangle \\ \Downarrow \\ \langle \cup_{1\le i\le m}\{\lambda\, \overline{x_k\colon T_k}.\, M_i \stackrel{?}{=} \lambda\, \overline{x_k\colon T_k}.\, N_i\} \cup D, V\rangle,\end{array} \qquad \mathcal{LU}_2$$

where a is an arbitrary atom.
Variable elimination

$$\begin{array}{c}\langle\{\lambda\, \overline{x_k\colon T_k}.\, F(\overline{x_k}) \stackrel{?}{=} \lambda\, \overline{x_k\colon T_k}.\, N\} \cup D, V\rangle \\ \Downarrow \\ \langle\{\lambda\, \overline{x_k\colon T_k}.\, F(\overline{x_k}) \stackrel{?}{=} \lambda\, \overline{x_k\colon T_k}.\, N\} \cup \theta(D)\!\downarrow, V\rangle,\end{array} \qquad \mathcal{LU}_3$$

where

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

Imitation

$$\begin{array}{c}\langle\{\lambda\, \overline{x_k\colon T_k}.\, F(\overline{M_m}) \stackrel{?}{=} \lambda\, \overline{x_k\colon T_k}.\, a(\overline{N_n})\} \cup D, V\rangle \\ \Downarrow \\ \langle\{F \stackrel{?}{=} P, \lambda\, \overline{x_k\colon T_k}.\, F(\overline{M_m}) \stackrel{?}{=} \lambda\, \overline{x_k\colon T_k}.\, a(\overline{N_n})\} \cup \{F/P\}(D)\!\downarrow, V\rangle,\end{array} \qquad \mathcal{LU}_{4a}$$

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.

$$P = \lambda\, \overline{y_m\colon S_m}.\, a(\overline{\lambda\, \overline{z_{p_n}\colon R_{p_n}}.\, H_n(\overline{y_{k_{q_m,m}}}, \overline{z_{p_n}})}).$$

Projection

$$\begin{array}{c}\langle\{\lambda\, \overline{x_k\colon T_k}.\, F(\overline{M_m}) \stackrel{?}{=} \lambda\, \overline{x_k\colon T_k}.\, a(\overline{N_n})\} \cup D, V\rangle \\ \Downarrow \\ \langle\{F \stackrel{?}{=} P, \lambda\, \overline{x_k\colon T_k}.\, F(\overline{M_m}) \stackrel{?}{=} \lambda\, \overline{x_k\colon T_k}.\, a(\overline{N_n})\} \cup \{F/P\}(D)\!\downarrow, V\rangle,\end{array} \qquad \mathcal{LU}_{4b}$$

falls

- F is a free variable and a an arbitrary atom,
- P is a variant of a selective ith projection binding for $1 \le i \le m$, appropriate to F, that is, $P = \lambda\, \overline{y_m\colon S_m}.\, y_i(\overline{\lambda\, \overline{z_{p_q}\colon R_{p_q}}.\, H_q(\overline{y_{k_{q_m,m}}}, \overline{z_{p_q}})})$, and

- head$(M_i) = a$, if head(M_i) is a constant.

Explosion

$$\begin{array}{c} \langle\{\lambda\,\overline{x_k\colon T_k}.\, F(\overline{M_m}) \stackrel{?}{=} \lambda\,\overline{x_k\colon T_k}.\, G(\overline{N_n})\} \cup D, V\rangle \\ \Downarrow \\ \langle\{F \stackrel{?}{=} P, \lambda\,\overline{x_k\colon T_k}.\, F(\overline{M_m}) \stackrel{?}{=} \lambda\,\overline{x_k\colon T_k}.\, G(\overline{N_n})\} \cup \{F/P\}(D)\!\downarrow, V\rangle, \end{array} \qquad \mathcal{LU}_{4c}$$

where

- F and G are free variables and
- $P = \lambda\,\overline{y_m\colon S_m}.\, a(\overline{\lambda\,\overline{z_{p_n}\colon R_{p_n}}.\, H_n(\overline{y_{k_{q_m,m}}}, \overline{z_{p_n}})})$ is a variant of some arbitrary selective partial binding appropriate to F such that $a \neq F$ and $a \neq G$

$\triangle$

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

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

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

Theorem 5.4 (Correctness) *If $U = \langle D \mid FV(D)\rangle$ is a unification problem in $\mathcal{L}^{\rightarrow}$ and $U \stackrel{*}{\Longrightarrow}_{\mathcal{LU}} U'$, with U' in solved form and non-forgetful, then the substitution $\lceil U' \rceil^{\mathrm{SUB}}$ is a non-forgetful unifier of U.*

Proof: The proof is exactly the same as for the correctness of $\mathcal{PU}$.

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

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 $X_i = BV(M_i) \cap \{x_1, \ldots, x_n\} = \{x_{k_{1,i}}, \ldots, x_{k_{q_i,i}}\}$ with $1 \leq k_{1,i} < k_{2,i} < \cdots < k_{q_i,i} \leq n$. Let $P_i = \eta[H_i(x_{k_{1,i}}, \ldots, x_{k_{q_i,i}})]$, 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 $P = \lambda\,\overline{x_n\colon T_n}.\, a(\overline{P_m})$ is a selective partial binding. Let $\tau = \{H_1/\lambda\,\overline{x_{k_{q_1,1}}}.\, M_1, \ldots, H_m/\lambda\,\overline{x_{k_{q_m,m}}}.\, M_m\}$. τ is obviously non-forgetful and $\tau(P_i) \stackrel{*}{\longrightarrow}_\beta M_i$, for each i, $1 \leq i \leq m$. Thus $\tau(P) \stackrel{*}{\longrightarrow}_\beta M$.

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 τ such that*

$$\begin{array}{rcl} \theta & = & \{F/M\} \cup \tau \cup \theta' \, [\mathcal{DOM}(\theta)] \\ & =_\beta & \tau \circ \{F/P\} \cup \theta' \, [\mathcal{DOM}(\theta)] \end{array}$$

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

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

$$\begin{array}{rcl} \{F/M\} & = & \{F/M\} \cup \tau \, [\mathcal{DOM}(\theta)] \\ & =_\beta & \tau \circ \{F/P\} \, [\mathcal{DOM}(\theta)] \end{array}$$

If θ is idempotent and $\mathcal{DOM}(\tau) \cap \mathcal{I}(\theta) = \emptyset$ then $\mathcal{DOM}(\theta'') \cap \mathcal{I}(\theta'') = \emptyset$. Finally $\theta''(P) = \tau(P) \stackrel{*}{\longrightarrow}_\beta \theta''(F)$ shows that θ'' is a unifier of $F \stackrel{?}{=} P$.

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

$$\langle\langle\{M \stackrel{?}{=} M\} \cup D \mid V\rangle, \theta\rangle \Rightarrow \langle\langle D \mid V\rangle, \theta\rangle \qquad \mathcal{LV}_1$$

Decomposition

$$\begin{array}{c}\langle\langle\{\lambda\,\overline{x_k\colon T_k}.\, a(\overline{M_m}) \stackrel{?}{=} \lambda\,\overline{x_k\colon T_k}.\, a(\overline{N_m})\} \cup D \mid V\rangle, \theta\rangle \\ \Downarrow \\ \langle\langle\cup_{1\le i\le m}\{\lambda\,\overline{x_k\colon T_k}.\, M_i \stackrel{?}{=} \lambda\,\overline{x_k\colon T_k}.\, N_i\} \cup D \mid V\rangle, \theta\rangle,\end{array} \qquad \mathcal{LV}_2$$

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

Variable elimination

$$\begin{array}{c}\langle\langle\{\lambda\,\overline{x_k\colon T_k}.\, F(\overline{x_k}) \stackrel{?}{=} \lambda\,\overline{x_k\colon T_k}.\, N\} \cup D \mid V\rangle, \theta\rangle \\ \Downarrow \\ \langle\langle\{\lambda\,\overline{x_k\colon T_k}.\, F(\overline{x_k}) \stackrel{?}{=} \lambda\,\overline{x_k\colon T_k}.\, N\} \cup \sigma(D)\!\downarrow \mid V\rangle, \theta\rangle,\end{array} \qquad \mathcal{LV}_3$$

where

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

Selective partial binding

$$\begin{array}{c}\langle\langle\{\lambda\,\overline{x_k\colon T_k}.\, F(\overline{M_m}) \stackrel{?}{=} \lambda\,\overline{x_k\colon T_k}.\, b(\overline{N_n})\} \cup D \mid V\rangle, \{F/Q\} \cup \theta\rangle \\ \Downarrow \\ \langle\langle\{F \stackrel{?}{=} P, \lambda\,\overline{x_k\colon T_k}.\, F(\overline{M_m}) \stackrel{?}{=} \lambda\,\overline{x_k\colon T_k}.\, b(\overline{N_n})\} \cup \{F/P\}(D)\!\downarrow \mid V\rangle, \{F/Q\} \cup \tau \cup \theta\rangle,\end{array} \qquad \mathcal{LV}_4$$

where

- F is a free variable and the equation $\lambda\,\overline{x_k\colon T_k}.\, F(\overline{M_m}) \stackrel{?}{=} \lambda\,\overline{x_k\colon T_k}.\, a(\overline{N_n})$ is not solved,
- $Q = \lambda\,\overline{y_m\colon S_m}.\, a(\overline{Q_n})$,
- P is a variant of a selective partial binding appropriate to F, e.g.

$$P = \lambda\,\overline{y_m\colon S_m}.\, a(\overline{\lambda\,\overline{z_{p_n}\colon R_{p_n}}.\, H_n(\overline{y_{n_{q_n}}}, \overline{z_{p_n}})}),$$

  and
- $\tau = \{H_1/\lambda\,\overline{y_m\colon S_m}.\, Q_1, \ldots, H_n/\lambda\,\overline{y_m\colon S_m}.\, Q_n\}$.

$\triangle$

Lemma 5.8 *If $\theta \in \mathrm{SU}_{nf}(U)$ for some unification problem U not in solved form and W is a set of variables, then there exists some transformation $\langle U, \theta\rangle \Longrightarrow_{\mathcal{LV}} \langle U', \theta'\rangle$ such that*

1. *$\theta = \theta'\,[W]$;*
2. *If θ is idempotent then θ' is an idempotent non-forgetful unifier of U' and*
3. *$U \Longrightarrow_{\mathcal{LU}} U'$.*

Proof: Because U is not in solved form there exists an equation $M \stackrel{?}{=} N$, 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 $\mathrm{head}(M)$ is not a free variable in $\mathcal{DOM}(\theta)$, we can use $\mathcal{LV}_2$ too.

2. If $\text{head}(M) = \text{head}(N) \notin \mathcal{DOM}(\theta)$ then we can use $\mathcal{LV}_2$.
3. Otherwise $M \neq N$ and we have one of the following two cases:
   (a) $\text{head}(M) \neq \text{head}(N)$: Because there exists a unifier for U one of M or N must be flexible. We assume that M is a flexible term.
   (b) $\text{head}(M) = \text{head}(N) \in \mathcal{DOM}(\theta)$: In this case M is flexible too.

   Let $M = \lambda\,\overline{x_k\!:T_k}.\,F(M_m)$ and $N = \lambda\,\overline{x_k\!:T_k}.\,N'$. Then $\mathcal{LV}_4$ is applicable or if $M \xrightarrow{*}_\eta F$ and $F \notin FV(N)$ 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 θ' 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 τ such that

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

$$\theta' = \{F/Q\} \cup \tau \cup \phi =_\beta \tau \circ \{F/P\} \cup \phi, \text{ and}$$
$$U' = \{F/P\}(U) \cup \{F \stackrel{?}{=} P\}.$$

The requirements for θ' are fulfilled

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

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

   and
   $$\theta'(P) = \tau(P) \xrightarrow{*}_\beta Q = \theta'(F).$$
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 $\text{head}(N)$ is not a variable in $\mathcal{DOM}(\theta)$ and
     - if $\text{head}(Q) = \text{head}(N)$ then $U \Longrightarrow_{\mathcal{LU}_{4a}} U'$ or
     - if $\text{head}(Q) \neq \text{head}(N)$ then $U \Longrightarrow_{\mathcal{LU}_{4b}} U'$.
   - If $\text{head}(N)$ is a variable then $U \Longrightarrow_{\mathcal{LU}_{4c}} U'$.

Lemma 5.9 *If $\theta \in \mathrm{SU}_{nf}(U)$ 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 $U = \langle D \mid FV(D)\rangle$, if $\theta \in \mathrm{SU}_{nf}(U)$ 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 $\lceil U_n \rceil^{\text{SUB}} \leq_\beta \theta\,[FV(D)]$.

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

$$\langle U, \theta \rangle = \langle U_0, \theta_0 \rangle \overset{*}{\Longrightarrow}_{\mathcal{LV}} \langle U_l, \theta_l \rangle$$

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

$$U = U_0 \overset{*}{\Longrightarrow}_{\mathcal{LU}} U_l.$$

Using the previous lemma, U_l is in solved form. So we have $\lceil U_l \rceil^{\mathrm{SUB}} \leq_\beta \theta_l = \theta \, [W]$.

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

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

Obviously we have yet considered the problem of $FV(D)$-restricted unification or *unrestricted unification*. For $D = \{M_1 \overset{?}{=} N_1, \ldots, M_n \overset{?}{=} N_n\}$ the problem of $FV(M_1, \ldots, M_n) \setminus FV(N_1, \ldots, N_n)$-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 Σ and a reduction of the set of variables $\mathcal{V}$ generating the set of λ-terms. For a *V*-restricted unification problem $\langle D \mid V \rangle$ let $\mathcal{V}' = \mathcal{V} \setminus \mathit{symbols}(((FV_{\mathcal{V}}(D) \setminus V) \cup (V \setminus FV_{\mathcal{V}}(D))))$ and $\Sigma' = \Sigma \cup \mathit{symbols}((FV(D) \setminus V))$. The free variables in $FV_{\mathcal{V}}(D) \setminus V$ 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 $V \setminus FV_{\mathcal{V}}(D)$ are neither included in $\mathcal{V}'$ nor in Σ'. This is necessary if we want to use $\mathcal{LU}$ 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 $\mathcal{L}^{\rightarrow}(\mathcal{V},\Sigma)$ and $\mathcal{L}^{\rightarrow}(\mathcal{V}',\Sigma')$: f renames bound variables $x{:}T$ such that $x \in \mathit{symbols}((FV(D) \setminus V) \cup (V \setminus FV(D)))$ to some $y{:}T$ with $y \in \mathcal{V}'$. So $f(M)$ is only a α-variant of M. Because we have considered α-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 $\mathrm{SUB}(\mathcal{L}^{\rightarrow}(\mathcal{V},\Sigma)^{\mathcal{V}})_{|\mathcal{V} \setminus (FV(D) \setminus V)}$ into substitutions in $\mathrm{SUB}(\mathcal{L}^{\rightarrow}(\mathcal{V}',\Sigma')^{\mathcal{V}'})$ and between systems in $\mathcal{L}^{\rightarrow}(\mathcal{V},\Sigma)$ and systems in $\mathcal{L}^{\rightarrow}(\mathcal{V}',\Sigma')$.

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

$$\langle D \mid FV_{\mathcal{V}'}(D) \rangle = U_0 \Longrightarrow_{\mathcal{LU}} U_1 \Longrightarrow_{\mathcal{LU}} \cdots \Longrightarrow_{\mathcal{LU}} U_n$$

in $\mathcal{L}^{\rightarrow}(\mathcal{V}',\Sigma')$ such that U_n is in solved form and possibly non-forgetful, then $\lceil U_n \rceil^{\mathrm{SUB}}$ is a V-restricted unifier of U.

Proof: The correctness of $\mathcal{LU}$ implies that $\lceil U_n \rceil^{\mathrm{SUB}}$ is a unifier of U. We have to show that it is *V*-restricted. Because all variables in $FV_{\mathcal{V}}(D) \setminus V$ are considered as constants in D the free variables of D are a subset of V. Because $\mathcal{V}'$ and $FV_{\mathcal{V}'}(D) \setminus V$ are disjoint no free variable in U_n will be in $FV_{\mathcal{V}'}(D) \setminus V$. So the domain of $\lceil U_n \rceil^{\mathrm{SUB}}$ will be a subset of V. So $\lceil U_n \rceil^{\mathrm{SUB}}$ is a *V*-restricted unifier of U.

Theorem 6.3 (Completeness) *If $U = \langle D \mid V\rangle$ is a V-restricted unification problem in $\mathcal{L}^{\rightarrow}(\mathcal{V}, \Sigma)$ and θ is a non-forgetful unifier of U then there exists a transformation sequence*

$$\langle D \mid V\rangle = U_0 \Longrightarrow_{\mathcal{LU}} U_1 \Longrightarrow_{\mathcal{LU}} \cdots \Longrightarrow_{\mathcal{LU}} U_n$$

in $\mathcal{L}^{\rightarrow}(\mathcal{V}', \Sigma')$ such that U_n is in solved form and possibly non-forgetful and

$$\lceil U_n \rceil^{\text{SUB}} \leq \theta\,[V].$$

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

$$\langle D \mid FV_{\mathcal{V}'}(D)\rangle = U_0' \overset{*}{\Longrightarrow}_{\mathcal{LU}} \langle D_n \mid FV_{\mathcal{V}'}(D)\rangle = U_n'$$

in $\mathcal{L}^{\rightarrow}(\mathcal{V}', \Sigma')$ 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

$$\langle D \mid V\rangle = U_0 \overset{*}{\Longrightarrow}_{\mathcal{LU}} \langle D_n \mid V\rangle = U_n$$

We have $\lceil U_n' \rceil^{\text{SUB}} \leq \theta\,[FV_{\mathcal{V}'}(D)]$.

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

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

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

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

For the same reason, no variable $x \in (V \setminus FV_{\mathcal{V}'}(D))$ is an element of $\mathcal{COD}(\lceil U_n \rceil^{\text{SUB}})$: Because they are neither in $\mathcal{V}'$ nor in Σ', they will not be introduced in some transformation step

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

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

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

sort	T_1
constants	$a : T_1$
variable	$F : ((T_1 \rightarrow T_1) \rightarrow T_1)$

we consider the second-order matching problem

$$M_0 = \langle\{F{:}T_1(\lambda\, x{:}T_1.\, x) \stackrel{?}{=} a{:}T_1\} \mid F\rangle$$

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

$$M_1 = \langle \{ H_1 \colon T_1 (\lambda\, x \colon T_1 .\, x) \stackrel{?}{=} a \colon T_1 \} \mid F \rangle.$$

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 substition 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

Alonzo Church, 1941. *The Calculi of Lambda-Conversion.* Number 6 in Annals of Mathematics Studies. Princeton University Press, Princeton, USA.

W. Snyder and J. Gallier, 1989. Higher-Order Unification Revisited: Complete Sets of Transformations. *Journal of Symbolic Computation*, Vol. 8, pp. 101–140.