Typed λ-calculus: Substitution and Equations

P. B. Levy

University of Birmingham

1 Substitution again

1.1 Substitutions and Renamings

Suppose we have a term $\Gamma \vdash M : B$, and we want to turn it into a term in context Δ, by replacing the identifiers. For example, we're given the term

$$\mathtt{x} : \mathtt{int}, \mathtt{y} : \mathtt{bool}, \mathtt{z} : \mathtt{int} \vdash \mathtt{z}+\mathtt{case}\ \mathtt{y}\ \mathtt{of}\ \{\mathtt{true}.\ \mathtt{x}+\mathtt{z}, \mathtt{false}.\ \mathtt{x}+1\} : \mathtt{int}$$

and we want to change it to something in the context $\mathtt{u} : \mathtt{bool}, \mathtt{x} : \mathtt{int}, \mathtt{y} : \mathtt{bool}$.

A *substitution* from Γ to Δ is a function k taking each identifier $\mathtt{x} : A$ in Γ to a term $\Delta \vdash k(\mathtt{x}) : A$.

For example, using the above Γ and Δ, a substitution from Γ to Δ is

$$\begin{aligned} &\mathtt{x} \mapsto 3 + \mathtt{x} \\ &\mathtt{y} \mapsto \mathtt{u} \\ &\mathtt{z} \mapsto \mathtt{case}\ \mathtt{y}\ \mathtt{of}\ \{\mathtt{true}.\ \mathtt{x} + 2, \mathtt{false}.\ \mathtt{x}\} \end{aligned}$$

We write k^*M for the result of replacing all the free identifiers in M according to k (avoiding capture, of course). In the above example, we obtain

$$\begin{aligned} &\mathtt{u} : \mathtt{bool}, \mathtt{x} : \mathtt{int}, \mathtt{y} : \mathtt{bool} \vdash \\ &\mathtt{case}\ \mathtt{y}\ \mathtt{of}\ \{\mathtt{true}.\ \mathtt{x} + 2, \mathtt{false}.\ \mathtt{x}\}+ \\ &\mathtt{case}\ \mathtt{u}\ \mathtt{of}\ \{\mathtt{true}.\ (3 + \mathtt{x}) + \mathtt{case}\ \mathtt{y}\ \mathtt{of}\ \{\mathtt{true}.\ \mathtt{x} + 2, \mathtt{false}.\ \mathtt{x}\}, \\ &\mathtt{false}.\ (3 + \mathtt{x}) + 1\} : \mathtt{int} \end{aligned}$$

Exercise 1. Apply to the term

$$\mathtt{x} : \mathtt{int} \to \mathtt{int}, \mathtt{y} : \mathtt{int} \vdash \mathtt{let}\ 5\ \mathtt{be}\ \mathtt{w}.\ (\mathtt{xy}) + (\mathtt{xw}) : \mathtt{int}$$

the substitution

$$\begin{aligned} \mathtt{x} &\mapsto \mathtt{y} \\ \mathtt{y} &\mapsto \mathtt{w} + 1 \end{aligned}$$

to obtain a term in context

$$\mathtt{w} : \mathtt{int}, \mathtt{y} : \mathtt{int} \to \mathtt{int}, \mathtt{z} : \mathtt{int}$$

An important special kind of substitution is one that maps each identifier to an identifier; this is called a *renaming*. An even more special case is the inclusion from Γ to Γ', where $\Gamma \subseteq \Gamma'$. This is called *weakening*. You will often see it expressed as a proposition.

Proposition 1. *If* $\Gamma \subseteq \Gamma'$ *and* $\Gamma \vdash M : A$ *then* $\Gamma' \vdash M : A$.

This is proved by induction, using the fact that if $\Gamma \subseteq \Gamma'$ then $\Gamma, \mathtt{x} : B \subseteq \Gamma', \mathtt{x} : B$.

1.2 Substitution by Induction

Let us think how to define substitution on terms (rather than on binding diagrams) by induction. Some of the inductive clauses are easy:

$$\begin{aligned} k^*3 &= 3 \\ k^*(M + N) &= k^*M + k^*N \\ k^*\mathtt{x} &= k(\mathtt{x}) \end{aligned}$$

But what about substituting into a `let` expression? Let's first remember the typing rule for `let` :

$$\frac{\Gamma \vdash M : A \quad \Gamma, \mathtt{x} : A \vdash N : B}{\Gamma \vdash \mathtt{let}\ M\ \mathtt{be}\ \mathtt{x}.\ N : B}$$

We define

$$k^*(\mathtt{let}\ M\ \mathtt{be}\ \mathtt{x}.\ N) = \mathtt{let}\ k^*M\ \mathtt{be}\ \mathtt{w}.\ (k, \mathtt{x} \mapsto \mathtt{w})^*N$$

where $\mathtt{w}$ is some identifier that doesn't appear in Δ, and the substitution $\Gamma, \mathtt{x} : A \xrightarrow{k,\mathtt{x}\mapsto\mathtt{w}} \Delta, \mathtt{x} : A$ is defined to map $(\mathtt{y} : B) \in \Gamma$ (provided $\mathtt{y} \neq \mathtt{x}$) to $k(\mathtt{y})$, and $\mathtt{x}$ to $\mathtt{w}$. Note the use of Proposition 1 in this definition: $\Delta, \mathtt{w} : A \vdash k(\mathtt{y}) : B$ follows from $\Delta \vdash k(\mathtt{y}) : B$ since $\mathtt{w} \notin \Delta$.

A consequence of this is that if you want to prove a theorem about substitution, you'll first have to prove it for renaming, or at least for weakening.

Next we define

- the identity subsitution on Γ to send each $(\mathtt{x} : A) \in \Gamma$ to $\mathtt{x}$
- the composite of substitutions $\Gamma \xrightarrow{k} \Gamma' \xrightarrow{l} \Gamma''$ to send $(\mathtt{x} : A) \in \Gamma$ to $l^*(k(\mathtt{x}))$.

While we have defined subsitution for terms, this involves an arbitrary choice of fresh identifier. Because of this, it is only on binding diagrams that we obtain a *canonical* operation. Furthermore, provided we work with binding diagrams (or up to α-equivalence), we have equations:

$$(k; l)^* M = l^* k^* M$$
$$\mathsf{id}_\Gamma^* M = M$$

It follows that contexts and substitutions form a category, i.e. composition satisfies the associativity, left unital and right unital laws.

2 Evaluation Through $\boldsymbol{\beta}$-reduction

Intuitively, a β-reduction means simplification. I'll write $M \rightsquigarrow N$ to mean that M can be simplified to N. We begin with some arithmetic simplifications, sometimes called δ-*reductions*:

$$\underline{m} + \underline{n} \rightsquigarrow \underline{m+n}$$
$$\underline{m} \times \underline{n} \rightsquigarrow \underline{m \times n}$$
$$\underline{m} > \underline{n} \rightsquigarrow \mathtt{true} \text{ if } m > n$$
$$\underline{m} > \underline{n} \rightsquigarrow \mathtt{false} \text{ if } m \leqslant n$$

There is a β-reduction rule for local definitions:

$$\texttt{let}\ M\ \texttt{be}\ \texttt{x}.\ N \rightsquigarrow N[M/\texttt{x}]$$

But the most interesting are the β-reductions for all the types. The rough idea is: if you use an introduction rule and then, immediately, use an elimination rule, then they can be simplified.

For the boolean type, the β-reduction rule is

$$\begin{array}{c} \texttt{case true of}\ \{\texttt{true}.N, \texttt{false}.N'\} \rightsquigarrow N \\ \texttt{case false of}\ \{\texttt{true}.N, \texttt{false}.N'\} \rightsquigarrow N' \end{array}$$

For the type $A \times B$, if we use projections the β-reduction rule is

$$\begin{array}{c} \pi\langle M, M'\rangle \rightsquigarrow M \\ \pi'\langle M, M'\rangle \rightsquigarrow M' \end{array}$$

If we use pattern-matching, the β-reduction rule is

$$\texttt{case}\ \langle M, M'\rangle\ \texttt{of}\ \langle \texttt{x}, \texttt{y}\rangle.\ N \rightsquigarrow N[M/\texttt{x}, M'/\texttt{y}]$$

For the type $A + B$, the β-reduction rule is

$$\begin{array}{c} \texttt{case inl}\ M\ \texttt{of}\ \{\texttt{inl x}.\ N, \texttt{inr y}.\ N'\} \rightsquigarrow N[M/\texttt{x}] \\ \texttt{case inr}\ M\ \texttt{of}\ \{\texttt{inl x}.\ N, \texttt{inr y}.\ N'\} \rightsquigarrow N'[M/\texttt{y}] \end{array}$$

For the type $A \rightarrow B$, the β-reduction rule is

$$(\lambda\texttt{x}.M)N \rightsquigarrow M[N/\texttt{x}]$$

A term which is the left-hand-side of a β-reduction is called a *β-redex.*

You can simplify any term M by picking a subterm that's a β-redex, and reduce it. Do this again and again until you get a *β-normal* term, i.e. one that doesn't contain any β-redex. It can be shown that this process has to terminate (the *strong normalization theorem*).

Proposition 2. *A closed term M that is β-normal must have an introduction rule at the root. (Remember that we consider $\underline{n}$ to be an introduction rule, but not $+ \times >$.) Hence, if M has type* `int`*, then it must be $\underline{n}$ for some $n \in \mathbb{Z}$.*

We prove the first part by induction on M.

Exercise 2. All the sums that we did can be turned into expressions and evaluated using β-reduction. Try:

1. $\texttt{let } \langle 5, \langle 2, \texttt{true} \rangle \rangle \texttt{ be x. } \pi\texttt{x} + \pi(\texttt{case x of } \langle \texttt{y}, \texttt{z} \rangle.\ \texttt{z})$
2. $\texttt{case } (\texttt{case } (3 < 7) \texttt{ of } \{\texttt{true. inr } 8 + 1, \texttt{false. inl } 2\}) \texttt{ of}$
   $\{\texttt{inl u. u} + 8, \texttt{inr u. u} + 3\}$
3. $((\lambda \texttt{f}_{\texttt{int} \to \texttt{int}}.\lambda \texttt{x}_{\texttt{int}}.(\texttt{f}(\texttt{fx})))\lambda \texttt{x}_{\texttt{int}}.(\texttt{x} + 3))2$

3 η-expansion

The η-expansion laws express the idea that

- everything of type `bool` is `true` or `false`
- everything of type $A \times B$ is a pair $\langle x, y \rangle$
- everything of type $A + B$ is a pair `inl` x or a pair `inr` x
- everything of type $A \to B$ is a function.

They are given by first applying an elimination, then an introduction (the opposite of β-reduction).

Let's begin with the type `bool`. Suppose we have a term $\Gamma \vdash M : \texttt{bool}$. Then for any term $\Gamma, \texttt{z} : \texttt{bool} \vdash N : B$, we can expand $N[M/\texttt{z}]$ to

$$\texttt{case } M \texttt{ of } \{\texttt{true}.\ N[\texttt{true}/\texttt{z}], \texttt{false}.\ N[\texttt{false}/\texttt{z}]\}$$

The reason this ought to be true is that, whatever we define the identifiers in Γ to be, M will be either `true` or `false`. Either way, both sides should be the same.

What about $A \times B$? If we're using projections, then any $\Gamma \vdash M : A \times B$ can be η-expanded to $\langle \pi M, \pi' M \rangle$.

And if we're using pattern-match, for terms $\Gamma \vdash M : A \times B$ and $\Gamma, \texttt{z} : A \times B \vdash N : C$, we can expand $N[M/\texttt{z}]$ into

$$\texttt{case } M \texttt{ of } \langle \texttt{x}, \texttt{y} \rangle N[\langle \texttt{x}, \texttt{y} \rangle / \texttt{z}]$$

(I'm supposing the $\mathtt{x}$ and $\mathtt{y}$ we use here don't appear in $\Gamma, \mathtt{z} : A \times B$.)

For $A+B$, it's similar. Suppose $\Gamma \vdash M : A+B$ and $\Gamma, \mathtt{z} : A+B \vdash N : C$. Then $N[M/\mathtt{z}]$ can be expanded into

$$\mathtt{case}\ M\ \mathtt{of}\ \{\mathtt{inl}\ \mathtt{x}.N[\mathtt{inl}\ \mathtt{x}/\mathtt{z}], \mathtt{inr}\ \mathtt{y}.N[\mathtt{inr}\ \mathtt{y}/\mathtt{z}]\}$$

(Again, I'm supposing the $\mathtt{x}$ and $\mathtt{y}$ don't appear in $\Gamma, \mathtt{z} : A + B$.)

And finally, $A \to B$. Any term $\Gamma \vdash M : A \to B$ can be expanded as $\lambda \mathtt{x}_A. (M\mathtt{x})$.

(Again, I'm supposing the $\mathtt{x}$ doesn't appear in Γ.)

Exercise 3. Take the term

$$\mathtt{f} : (\mathtt{int} + \mathtt{bool}) \to (\mathtt{int} + \mathtt{bool}) \vdash \mathtt{f} : (\mathtt{int} + \mathtt{bool}) \to (\mathtt{int} + \mathtt{bool})$$

Apply an η-expansion for $\to$, then for $+$, then for $\mathtt{bool}$.

4 Equality

λ-calculus isn't just a set of terms; it comes with an equational theory. If $\Gamma \vdash M : B$ and $\Gamma \vdash N : B$, we write $\Gamma \vdash M = N : B$ to express the intuitive idea that, no matter what we define the identifiers in Γ to be, M and N have the same "meaning" (even though they're different expressions).

First of all we need rules to say that this is an equivalence relation:

$$\frac{\Gamma \vdash M : B}{\Gamma \vdash M = M : B} \qquad \frac{\Gamma \vdash M = N : B}{\Gamma \vdash N = M : B}$$

$$\frac{\Gamma \vdash M = N : B \quad \Gamma \vdash N = P : B}{\Gamma \vdash M = P : B}$$

Secondly, we need rules to say that this is *compatible*—preserved by every construct:

$$\frac{\Gamma \vdash M = M' : A \quad \Gamma, \mathtt{x} : A \vdash N = N' : B}{\Gamma \vdash \mathtt{let}\ M\ \mathtt{be}\ \mathtt{x}.\ N = \mathtt{let}\ M'\ \mathtt{be}\ \mathtt{x}.\ N' : B}$$

and so forth. A compatible equivalence relation is often called a *congruence*.

Thirdly, each of the β-reductions that we've seen is an axiom of this theory.

$$\frac{\Gamma \vdash N : B \quad \Gamma \vdash N' : B}{\Gamma \vdash \texttt{case true of } \{\texttt{true}.\ N, \texttt{false}.\ N'\} = N : B}$$

$$\frac{\Gamma, \texttt{x} : A \vdash M : B \quad \Gamma \vdash N : A}{\Gamma \vdash (\lambda \texttt{x}_A.M)N = M[N/\texttt{x}] : B}$$

Fourthly, each of the η-expansions is an axiom of the theory, e.g.

$$\frac{\Gamma \vdash M : A \to B}{\Gamma \vdash M = \lambda \texttt{x}_A.\,(M\texttt{x}) : A \to B}$$

Proposition 3. *If* $\Gamma \vdash M = N : B$ *and* $\Gamma \xrightarrow{k} \Delta$ *is a substitution, then* $\Delta \vdash k^*M = k^*N : B$

As usual we prove this first for renaming, or at least for substitution.

5 Exercises

1. Suppose that $\Gamma \vdash M : \texttt{bool}$ and $\Gamma \vdash N_0, N_1, N_2, N_3 : C$. Show that

$$\begin{array}{l} \Gamma \vdash \texttt{case } M \texttt{ of } \{ \\ \quad \texttt{true. case } M \texttt{ of } \{\texttt{true}.N_0, \texttt{false}.N_1\}, \\ \quad \texttt{false. case } M \texttt{ of } \{\texttt{true}.N_2, \texttt{false}.N_3\} \\ \} \\ = \texttt{case } M \texttt{ of } \{\texttt{true}.N_0, \texttt{false}.N_3\} : C \end{array}$$

2. Show that $\texttt{inl} -$ is injective, i.e. if $\Gamma \vdash M, M' : A$ and $\Gamma \vdash \texttt{inl } M = \texttt{inl } M' : A + B$ then $\Gamma \vdash M = M' : A$.
3. Write down the η-law for the 0 type.
4. Given a term $\Gamma, \texttt{x} : A \vdash M : 0$, show that it is an "isomorphism" in the sense that there is a term $\Gamma, \texttt{y} : 0 \vdash N : A$ satisfying

$$\Gamma, \texttt{y} : 0 \vdash M[N/\texttt{x}] = \texttt{y} : 0$$
$$\Gamma, \texttt{x} : A \vdash N[M/\texttt{y} = \texttt{x} : A$$

5. Give β and η laws for $\alpha(A, B, C, D, E)$ and for $\beta(A, B, C, D, E, F, G)$. (See yesterday's exercises for a description of these types.)