Problem

Do definitions like the following make sense as functions?

- $\mathsf{Var}(y)[x := t] = \mathtt{if}\ x = y\ \mathtt{then}\ t\ \mathtt{else}\ \mathsf{Var}(y)$
- $\mathsf{App}(s_1, s_2)[x{:=}t] = \mathsf{App}(s_1[x{:=}t], s_2[x{:=}t])$
- $\mathsf{Lam}\ ([a].s)[x{:=}t] = \mathsf{Lam}\ [a].(s[x{:=}t])$

  provided $a \# (x, t)$

Substitution as Relation

Let's define the four-place relation $\_[\_:=\_] \mapsto \_$:

$$\frac{}{\mathsf{Var}(x)[x := t] \mapsto t}$$

$$\frac{x \neq y}{\mathsf{Var}(y)[x := t] \mapsto \mathsf{Var}(y)}$$

$$\frac{s_1[x := t] \mapsto s'_1 \quad s_2[x := t] \mapsto s'_2}{\mathsf{App}(s_1, s_2)[x := t] \mapsto \mathsf{App}(s'_1, s'_2)}$$

$$\frac{s[x := t] \mapsto s' \quad a \mathrel{\#} (x, t)}{\mathsf{Lam}\,([a].s)[x := t] \mapsto \mathsf{Lam}\,[a].s'}$$

Substitution is "Total"

This substitution relation is "total":

$$\forall t_1\, x\, t_2.\ \exists t_3.\ t_1[x := t_2] \mapsto t_3$$

Proof by induction on t_1 using the convenient structural induction principle.

Convenient Principle

$$\frac{\begin{array}{l}\forall x\, a.\ \ P\,(\mathsf{Var}(a))\,x \\ \forall x\, t_1\, t_2.\,(\forall z.\,P\,t_1\,z) \Rightarrow (\forall z.\,P\,t_2\,z) \Rightarrow P\,(\mathsf{App}(t_1, t_2))\,x \\ \forall x\, a\, t.\ a \,\#\, x \Rightarrow (\forall z.\,P\,t\,z) \Rightarrow P\,(\mathsf{Lam}\,[a].t)\,x\end{array}}{P\,t\,x}$$

- $P = \lambda t_1(x, t_2).\ \exists t_3.\ t_1[x := t_2] \mapsto t_3$
- $t = t_1$
- $x = (x, t_2)$

Lambda-case: $a \,\#\, (x, t_2)$ and $\exists t_3.t[x := t_2] \mapsto t_3$ implies $\exists t_3.\ (\mathsf{Lam}\,[a].t)[x := t_2] \mapsto t_3$. Trivial.

Substitution is "Unique"

This substitution relation is, so to say, total:

$$\forall s_1\, s_2.\; t_1[x := t_2] \mapsto s_1 \wedge t_1[x := t_2] \mapsto s_2 \Rightarrow s_1 = s_2$$

Proof by induction on t_1 using the convenient structural induction principle. Lambda-Case relies on the fact...

Lambda-Case

This case is a bit hairy:

$$(\mathsf{Lam}\,[a].s)[x := t_2] \mapsto t \wedge a \# (x, t_2)$$
$$\Rightarrow\ \exists t'.\ t = (\mathsf{Lam}\,[a].t')\ \wedge\ s[x := t_2] \mapsto t'$$

Problem: if we know that $(\mathsf{Lam}\,[a].s)[x := t_2] \mapsto t$ and $a \# (x, t_2)$, do we know that t must be of the form $(\mathsf{Lam}\,[a].s')$ with $s[x := t_2] \mapsto s'$?

$$\frac{s[x{:=}t] \mapsto s' \quad a \# (x, t)}{\mathsf{Lam}\,([a].s)[x{:=}t] \mapsto \mathsf{Lam}\,[a].s'}$$

we get:

$t\ =\ \mathsf{Lar}$ ... ll as

$s'[x :=$ ...

we have ...

$\exists t'.\mathsf{Lam}$ ...

there are ...

$a = a' \wedge s'' = s$ or ... $\Rightarrow t' = s''$

$a \neq a' \wedge s = (a\,a')\bullet s' \wedge a \# s'$

Lambda-Case

This case is a bit hairy:

$$(\mathsf{Lam}\,[a].s)[x := t_2] \mapsto t \wedge a \# (x, t_2)$$
$$\Rightarrow\ \exists t'.\, t = (\mathsf{Lam}\,[a].t')\ \wedge\ s[x := t_2] \mapsto t'$$

we get:

$t = \mathsf{Lam}\,[a'].s''$ and $\mathsf{Lam}\,[a].s = \mathsf{Lam}\,[a'].s'$ as well as $s'[x := t_2] \mapsto s''$ and $a' \# (x, t_2)$

we have to show:

$$\exists t'.\mathsf{Lam}[a'].s'' = (\mathsf{Lam}[a].t') \wedge s[x := t_2] \mapsto t'$$

there are two cases:

$a = a' \wedge s' = s$ or $\Rightarrow t' = s''$

$a \neq a' \wedge s = (a\, a') \bullet s' \wedge a \# s'$

Lambda-Case

In the other case we have:

$a \neq a', t = \mathsf{Lam}\,[a'].s'',$
$s = (a\,a')\bullet s', a \# s'$
$s'[x := t_2] \mapsto s''$ and $a' \# (x, t_2), a \# (x, t_2)$

we have to show:

$\exists t'.\mathsf{Lam}[a'].s'' = \mathsf{Lam}[a].t' \wedge$
$\quad ((a\,a')\bullet s')[x := t_2] \mapsto t'$

we are setting t' to $(a\,a')\bullet s''$ and show:

$\mathsf{Lam}[a'].s'' = \mathsf{Lam}[a].((a\,a')\bullet s'')$ and
$((a\,a')\bullet s')[x := t_2] \mapsto (a\,a')\bullet s''$

Lambda-Case

In the other case we have:

$$a \neq a', t = \mathsf{Lam}\,[a'].s'',$$
$$s = (a\,a')\bullet s', a \# s'$$
$$s'[x := t_2] \mapsto s'' \text{ and } a' \# (x, t_2), a \# (x, t_2)$$

we have to show:

$$\exists t'.\mathsf{Lam}[a'].s'' = \mathsf{Lam}[a].t' \wedge$$
$$((a\,a')\bullet s')[x := t_2] \mapsto t'$$

we are setting t' to $(a\,a')\bullet s''$ and show:

$$s'' = ((a\,a')\bullet(a\,a')\bullet s''), a' \# (a\,a')\bullet s'' \text{ and}$$
$$((a\,a')\bullet s')[x := t_2] \mapsto (a\,a')\bullet s''$$

Lambda-Case

In the other case we have:

$$a \neq a', t = \mathsf{Lam}\,[a'].s'',$$
$$s = (a\,a')\bullet s', a \# s'$$
$$s'[x := t_2] \mapsto s'' \text{ and } a' \# (x, t_2), a \# (x, t_2)$$

we have to show:

$$\exists t'.\mathsf{Lam}[a'].s'' = \mathsf{Lam}[a].t' \wedge$$
$$((a\,a')\bullet s')[x := t_2] \mapsto t'$$

we are setting t' to $(a\,a')\bullet s''$ and show:

$$a' \# (a\,a')\bullet s'' \text{ and}$$
$$((a\,a')\bullet s')[x := t_2] \mapsto (a\,a')\bullet s''$$

Lambda-Case

In the other case we have:

$a \neq a', t = \text{Lam}\,[a'].s''$,
$s = (a\,a')\bullet s', a \# s'$
$s'[x := t_2] \mapsto s''$ and $a' \# (x$

we have to show:

$\exists t'.\text{Lam}[a'].s'' = \text{Lam}[a].t' \wedge$
$((a\,a')\bullet s')[x := t_2] \mapsto t'$

we are setting t' to $(a\,a')\bullet s''$ (

$a \# s''$ and
$((a\,a')\bullet s')[x := t_2] \mapsto (a\,a')\bullet s''$

We have $a \# s'$; now
$s'[x := t_2] \mapsto s''$ implies
$a \# s''$

because $\text{supp}(s'')$ is a
subset of
$(\text{supp}(s') - \{x\}) \cup \text{supp}(t_2)$
and $a \# t_2$

Lambda-Case

In the other case we have:

$$a \neq a', t = \mathsf{Lam}\,[a'].s'',$$
$$s = (a\,a')\bullet s', a \# s'$$
$$s'[x := t_2] \mapsto s'' \text{ and } a' \# (x, t_2), a \# (x, t_2)$$

we have to show:

$$\exists t'.\mathsf{Lam}[a'].s'' = \mathsf{Lam}[a].t' \wedge$$
$$((a\,a')\bullet s')[x := t_2] \mapsto t'$$

we are setting t' to $(a\,a')\bullet s''$ and show:

$$((a\,a')\bullet s')[x := t_2] \mapsto (a\,a')\bullet s''$$

By equivariance we get from $s'[x := t_2] \mapsto s''$ that $((a\,a')\bullet s')[((a\,a')\bullet x) := ((a\,a')\bullet t_2)] \mapsto ((a\,a')\bullet s'')$ holds.

Lambda-Case

In the other case we have:

$$a \neq a', t = \text{Lam}\,[a'].s'',$$
$$s = (a\,a')\bullet s', a \# s'$$
$$s'[x := t_2] \mapsto s'' \text{ and } a' \# (x, t_2), a \# (x, t_2)$$

we have to show:

$$\exists t'.\text{Lam}[a'].s'' = \text{Lam}[a].t' \wedge$$
$$((a\,a')\bullet s')[x := t_2] \mapsto t'$$

we are setting t' to $(a\,a')\bullet s''$ and show:

$$((a\,a')\bullet s')[x := t_2] \mapsto (a\,a')\bullet s''$$

By equivariance we get from $s'[x := t_2] \mapsto s''$ that $((a\,a')\bullet s')[x := t_2)] \mapsto ((a\,a')\bullet s'')$ holds.

Definite Description

After having shown that

$$\forall t_1\, x\, t_2.\ \exists! t_3.\ t_1[x := t_2] \mapsto t_3$$

we can use Isabelle's definite description operator THE to define substitution as function.

if $P\ a$ and
$\forall x.\ P\ x \ \Rightarrow\ x = a$,

then $(THE\ x.\ P\ x) = a$

$(THE\ x.\ P\ x)$ denotes the x such that $P\ x$ is true, provided there exists a unique such x. Otherwise, it returns an "arbitrary" value.

Recursion Combinator

Next we will show that **under certain conditions** there exists a function fun such that:

$$fun_{f_{\mathsf{Var}},f_{\mathsf{App}},f_{\mathsf{Lam}}}\ \mathsf{Var}(a) = f_{\mathsf{Var}}\ a$$

$$fun_{f_{\mathsf{Var}},f_{\mathsf{App}},f_{\mathsf{Lam}}}\ \mathsf{App}(t_1, t_2)$$
$$= f_{\mathsf{App}}\ (fun_{f_{\mathsf{Var}},f_{\mathsf{App}},f_{\mathsf{Lam}}}\ t_1)(fun_{f_{\mathsf{Var}},f_{\mathsf{App}},f_{\mathsf{Lam}}}\ t_2)$$

$$fun_{f_{\mathsf{Var}},f_{\mathsf{App}},f_{\mathsf{Lam}}}\ \mathsf{Lam}\,[a].t$$
$$= f_{\mathsf{Lam}}\ a\ (fun_{f_{\mathsf{Var}},f_{\mathsf{App}},f_{\mathsf{Lam}}}\ t)$$

provided $a \# (f_{\mathsf{Var}}, f_{\mathsf{App}}, f_{\mathsf{Lam}})$

Example: Substitution

The parameters b and t correspond to the substitution $[b := t]$:

$$s_{\mathsf{Var}}\, b\, t \stackrel{\mathsf{def}}{=} \lambda a.\, \mathtt{if}\ a = b\ \mathtt{then}\ t\ \mathtt{else}\ \mathsf{Var}(a)$$

$$s_{\mathsf{App}}\, b\, t \stackrel{\mathsf{def}}{=} \lambda t_1\, t_2.\, \mathsf{App}(t_1, t_2)$$

$$s_{\mathsf{Lam}}\, b\, t \stackrel{\mathsf{def}}{=} \lambda a\, t.\, \mathsf{Lam}\, [a].t$$

The function we define is $subst_{\mathsf{lam}}\, t'$ which corresponds to $t'[b := t]$. It is defined as the function $fun_{s_{\mathsf{Var}}\, b\, t, s_{\mathsf{App}}\, b\, t, s_{\mathsf{Lam}}\, b\, t}$.

Freshness Lemma

- A criterion when a construction that "picks a fresh atom" is independent of which fresh atom is chosen:

Given $pt_{\alpha,\iota}$ and a function $h : \alpha \to \iota$:

- if $\mathsf{finite}(\mathsf{supp}_\alpha\ h)$ and

  $\exists a.\ a \# h \wedge a \# (h\ a)$

  then

  $\exists! fr.\ \forall a.\ a \# h \;\Rightarrow\; (h\ a) = fr$

Proof (Existence)

From the assumption there exists an a_0 with

1. $a_0 \# h$ and 2. $a_0 \# (h\ a_0)$

Proof (Existence)

From the assumption there exists an a_0 with

1. $a_0 \# h$ and 2. $a_0 \# (h\ a_0)$

We take $fr(h)$ to be $h\ a_0$ and have

3. $a \# h$ and claim that $(h\ a) = (h\ a_0)$

Proof (Existence)

From the assumption there exists an a_0 with

1. $a_0 \# h$ and 2. $a_0 \# (h\ a_0)$

We take $fr(h)$ to be $h\ a_0$ and have

3. $a \# h$ and claim that $(h\ a) = (h\ a_0)$

Trivially true for $a = a_0$. For $a \neq a_0$:

4. $a \# (h\ a_0)$

since $a \# h$ (3), $a \# a_0$, $\mathsf{finite}(\mathsf{supp}\ h)$, $\mathsf{finite}(\mathsf{supp}\ a_0)$ and thus

$$\mathsf{supp}(h\ a_0) \subseteq \mathsf{supp}(h) \cup \mathsf{supp}(a_0)$$

Proof (Existence)

From the assumption there exists an a_0 with

1. $a_0 \# h$ and 2. $a_0 \# (h\ a_0)$

We take $fr(h)$ to be $h\ a_0$ and have

3. $a \# h$ and claim that $(h\ a) = (h\ a_0)$

Trivially true for $a = a_0$. For $a \neq a_0$:

4. $a \# (h\ a_0)$

We get from (2,4) and (1,3):

5. $(a\, a_0) \bullet (h\ a_0) = (h\ a_0)$
6. $(a\, a_0) \bullet (h\ a_0) = ((a\, a_0) \bullet h)((a\, a_0) \bullet a_0) = (h\ a)$

Proof (Uniqueness)

From the assumption there exists an a_0 with $a_0 \# h$ and we have

1. $\forall a.\ a \# h \ \Rightarrow\ (h\ a) = fr_1$
2. $\forall a.\ a \# h \ \Rightarrow\ (h\ a) = fr_2$

Clearly, $fr_1 = fr_2$.

The freshness function $fr(h)$ can be used to define the lambda-case for recursive functions:

$$g(\mathsf{Lam}\,[a].t) = fr(\lambda a'.\ f_{\mathsf{Lam}}\, a'\, (a\, a') \bullet t)$$

and so for all "fresh" atoms a'' we have

$$g(\mathsf{Lam}\,[a].t) = f_{\mathsf{Lam}}\, a''\, (a\, a'') \bullet t$$

Recursion Comb. as. Rel

$$\frac{}{(\mathsf{Var}(a), \lambda\pi.f_{\mathsf{Var}}(\pi \bullet a)) \in rec_{f_{\mathsf{Var}}, f_{\mathsf{App}}, f_{\mathsf{Lam}}}}$$

$$\frac{(t_1, y_1) \in rec_{f_{\mathsf{Var}}, f_{\mathsf{App}}, f_{\mathsf{Lam}}} \quad (t_2, y_2) \in rec_{f_{\mathsf{Var}}, f_{\mathsf{App}}, f_{\mathsf{Lam}}}}{(\mathsf{App}(t_1, t_2), \lambda\pi.f_{\mathsf{App}}\,(y_1\,\pi)\,(y_2\,\pi)) \in rec_{f_{\mathsf{Var}}, f_{\mathsf{App}}, f_{\mathsf{Lam}}}}$$

$$\frac{(t, y) \in rec_{f_{\mathsf{Var}}, f_{\mathsf{App}}, f_{\mathsf{Lam}}}}{\begin{array}{l}(\mathsf{Lam}\,[a].t, \\ \lambda\pi.fr(\lambda a'.f_{\mathsf{Lam}}\,a'\,(y\,(\pi @ (a\,a'))))) \in rec_{f_{\mathsf{Var}}, f_{\mathsf{App}}, f_{\mathsf{Lam}}}\end{array}}$$

Recursion Comb. as. Rel

$$\frac{}{(\mathsf{Var}(a), \lambda\pi.f_{\mathsf{Var}}(\pi \bullet a)) \in rec_{f_{\mathsf{Var}}, f_{\mathsf{App}}, f_{\mathsf{Lam}}}}$$

$$\frac{(t_1, y_1) \in rec_{f_{\mathsf{Var}}, f_{\mathsf{App}}, f_{\mathsf{Lam}}} \quad (t_2, y_2) \in rec_{f_{\mathsf{Var}}, f_{\mathsf{App}}, f_{\mathsf{Lam}}}}{(\mathsf{App}(t_1, t_2), \lambda\pi.f_{\mathsf{App}}\,(y_1\,\pi)\,(y_2\,\pi)) \in rec_{f_{\mathsf{Var}}, f_{\mathsf{App}}, f_{\mathsf{Lam}}}}$$

$$\frac{(t, y) \in rec_{f_{\mathsf{Var}}, f_{\mathsf{App}}, f_{\mathsf{Lam}}}}{\begin{array}{l}(\mathsf{Lam}\,[a].t, \\ \lambda\pi.fr(\lambda a'.f_{\mathsf{Lam}}\,a'\,(y\,(\pi @ (a\,a'))))) \in rec_{f_{\mathsf{Var}}, f_{\mathsf{App}}, f_{\mathsf{Lam}}}\end{array}}$$

In the last case if a is "sufficiently fresh" then we have

$$(\mathsf{Lam}\,[a].t, \lambda\pi.f_{\mathsf{Lam}}\,a\,(y\,(\pi @ (a\,a)))) \in rec_{f_{\mathsf{Var}}, f_{\mathsf{App}}, f_{\mathsf{Lam}}}$$

Recursion Comb. as. Rel

$$\frac{}{(\mathsf{Var}(a), \lambda\pi.f_{\mathsf{Var}}(\pi \bullet a)) \in rec_{f_{\mathsf{Var}}, f_{\mathsf{App}}, f_{\mathsf{Lam}}}}$$

$$\frac{(t_1, y_1) \in rec_{f_{\mathsf{Var}}, f_{\mathsf{App}}, f_{\mathsf{Lam}}} \quad (t_2, y_2) \in rec_{f_{\mathsf{Var}}, f_{\mathsf{App}}, f_{\mathsf{Lam}}}}{(\mathsf{App}(t_1, t_2), \lambda\pi.f_{\mathsf{App}}\ (y_1\ \pi)\ (y_2\ \pi)) \in rec_{f_{\mathsf{Var}}, f_{\mathsf{App}}, f_{\mathsf{Lam}}}}$$

$$\frac{(t, y) \in rec_{f_{\mathsf{Var}}, f_{\mathsf{App}}, f_{\mathsf{Lam}}}}{\begin{array}{l}(\mathsf{Lam}\ [a].t,\\ \lambda\pi.fr(\lambda a'.f_{\mathsf{Lam}}\ a'\ (y\ (\pi @ (a\ a'))))) \in rec_{f_{\mathsf{Var}}, f_{\mathsf{App}}, f_{\mathsf{Lam}}}\end{array}}$$

In the last case if a is "sufficiently fresh" then we have

$$(\mathsf{Lam}\ [a].t, \lambda\pi.f_{\mathsf{Lam}}\ a\ (y\ \pi)) \in rec_{f_{\mathsf{Var}}, f_{\mathsf{App}}, f_{\mathsf{Lam}}}$$

Showing Uniqueness

One now has to show that this relation **under certain conditions** is a function - namely $fun_{f_{\mathsf{Var}},f_{\mathsf{App}},f_{\mathsf{Lam}}}$.

Certain conditions:

- finite(supp($f_{\mathsf{Var}}, f_{\mathsf{App}}, f_{\mathsf{Lam}}$))
- $\exists a.\ a \# f_{\mathsf{Lam}} \wedge \forall y.\ a \# f_{\mathsf{Lam}}\, a\, y$

Showing Uniqueness

One now has to show that this relation **under certain conditions** is a function - namely $fun_{f_{\mathsf{Var}}, f_{\mathsf{App}}, f_{\mathsf{Lam}}}$.

Certain conditions:

- finite(supp($f_{\mathsf{Var}}, f_{\mathsf{App}}, f_{\mathsf{Lam}}$))
- $\exists a.\ a \# f_{\mathsf{Lam}} \ \wedge\ \forall y.\ a \# f_{\mathsf{Lam}}\, a\, y$

In the case of substitution:

$$\exists a.\ a \# (s_{\mathsf{Lam}}\, b\, t) \ \wedge\ \forall y.\ a \# (s_{\mathsf{Lam}}\, b\, t)\, a\, y$$

Showing Uniqueness

One now has to show that this relation **under certain conditions** is a function - namely $fun_{f_{\mathsf{Var}}, f_{\mathsf{App}}, f_{\mathsf{Lam}}}$.

Certain conditions:

- $\mathsf{finite}(\mathsf{supp}(f_{\mathsf{Var}}, f_{\mathsf{App}}, f_{\mathsf{Lam}}))$
- $\exists a.\ a \# f_{\mathsf{Lam}} \ \wedge\ \forall y.\ a \# f_{\mathsf{Lam}}\, a\, y$

In the case of substitution:

$$\exists a.\ a \# (s_{\mathsf{Lam}}\, b\, t) \ \wedge\ \forall y.\ a \# \mathsf{Lam}\, [a].y$$

Showing Uniqueness

One now has to show that this relation **under certain conditions** is a function - namely $fun_{f_{\mathsf{Var}}, f_{\mathsf{App}}, f_{\mathsf{Lam}}}$.

Certain conditions:

- $\mathsf{finite}(\mathsf{supp}(f_{\mathsf{Var}}, f_{\mathsf{App}}, f_{\mathsf{Lam}}))$
- $\exists a.\ a \# f_{\mathsf{Lam}} \ \wedge\ \forall y.\ a \# f_{\mathsf{Lam}}\, a\, y$

In the case of substitution:

$$\exists a.\ a \# (s_{\mathsf{Lam}}\, b\, t) \ \wedge\ \forall y.\ a \# \mathsf{Lam}\, [a].y$$

$\mathsf{supp}(s_{\mathsf{Lam}}\, b\, t) \subseteq \mathsf{supp}(b, t)$