The Frankel-Mostowski (FM) model of abstraction applied to name-generation

Murdoch J. Gabbay

Heriot-Watt University, Edinburgh, Scotland, 9/1/2006

Substitution in the λ-calculus

Fix a countably infinite set of variable (symbols) $a, b, c, \ldots$. Let terms be defined by:

$$t ::= a \mid tt \mid \lambda a.t.$$

Henceforth let t, u be metavariables ranging over terms.

Define free variables inductively by:

$$fv(a) = \{a\} \qquad fv(tt') = fv(t) \cup fv(t')$$

$$fv(\lambda a.t) = (fv(t) \setminus \{a\})$$

Substitution in the λ-calculus

Fix a choice of fresh variable for each finite set of variables.

Define substitution by:

$$
\begin{aligned}
a[a{\mapsto}t] &\equiv t \\
a[b{\mapsto}t] &\equiv a \\
(uu')[a{\mapsto}t] &\equiv (u[a{\mapsto}t])(u'[a{\mapsto}t]) \\
(\lambda a.u)[a{\mapsto}t] &\equiv \lambda a.u \\
(\lambda b.u)[a{\mapsto}t] &\equiv \lambda b.(u[a{\mapsto}t]) && b \notin fv(t) \\
(\lambda b.u)[a{\mapsto}t] &\equiv \lambda b'.(u[b{\mapsto}b'][a{\mapsto}t]) && b \in fv(t),\ b' \text{ fresh}
\end{aligned}
$$

Here 'b' fresh' means 'pick b' fresh for the variables in the terms to the left of $\equiv$'.

$$a \notin fv(t) \quad \Longrightarrow \quad a \notin fv(u[a \mapsto t]).$$

Proof by induction on u: *(fails)*

- Suppose $u \equiv a$. Then $a[a \mapsto t] \equiv t$.
- Suppose $u \equiv b$. Then $b[a \mapsto t] \equiv b$.
- Suppose $u \equiv u'u''$. Then $(u'u'')[a \mapsto t] \equiv (u'[a \mapsto t])(u''[a \mapsto t])$. Now $a \notin fv(u')$ so $a \notin fv(u'[a \mapsto t])$ by inductive hypothesis. Similarly for u''. Result follows.
- Suppose $u \equiv \lambda a.u'$. Then $(\lambda a.u')[a \mapsto t] \equiv \lambda a.u'$. Result follows.
- Suppose $u \equiv \lambda b.u'$ and $b \notin fv(t)$. Then $(\lambda b.u')[a \mapsto t] \equiv \lambda b.(u'[a \mapsto t])$. By inductive hypothesis $a \notin fv(u'[a \mapsto t])$ and result follows.

A simple lemma

$$a \notin fv(t) \quad \Longrightarrow \quad a \notin fv(u[a \mapsto t]).$$

Proof by induction on u:

- Suppose $u \equiv \lambda b.u'$ and $b \in fv(t)$. Then $(\lambda b.u')[a \mapsto t] \equiv \lambda b'.(u'[b \mapsto b'][a \mapsto t])$.

Cannot apply the inductive hypothesis since $u'[b \mapsto b']$ is not a subterm of $\lambda b.u'$.

One solution: reason by induction on the size (length) of terms.

Choice of fresh variable symbol

$\text{И}a.\phi(a)$ means...

- There is some finite set S such that $\phi(a)$ need not hold for $a \in S$.
- However $\phi(a)$ holds for all $a \notin S$.

This enables us to pick a fresh element, without precise reference to what exactly it is fresh for.

For example

1. $\text{И}a.(a \neq b)$,
2. $\text{И}a.(a \notin \{b, c, d\})$, and
3. $\neg\text{И}a.(a \notin \text{set of variable symbols})$

all hold.

Example definition of α-equivalence

$$a \approx_\alpha a \qquad \frac{s \approx_\alpha s' \quad t \approx_\alpha t'}{st \approx_\alpha s't'}$$

$$\frac{\text{И}c.\ s[a\mapsto c] \approx_\alpha t[b\mapsto c]}{\lambda a.s \approx_\alpha \lambda b.t}$$

Ͷ only good for predicates

This notion of 'is fresh' does not help us directly to choose b' as e.g. in the 'b' fresh' in the definition of explicit substitution.

This is because explicit substitution was a function — $(\lambda b.u)$ maps under $[a{\mapsto}t]$ to $\lambda b'.(u[b{\mapsto}b'][a{\mapsto}t])$ where b' is fresh, and this term depends on which b' we choose (because we did not quotient by α-equivalence, because we wanted inductive principles!). All is not lost.

A word on sets

Standard models of set theory are well-founded trees where the parent-daughter relation is exactly set-membership $\in$.

Set theory is more than that because we cleverly choose axioms to define such a model.

This is a separate issue from what to do if somebody you trust gives you a set.

It's a well-founded tree.

Fix some collection $\mathbb{A}$ of atoms a, b, c. Let an FM set universe be well-founded trees with leaves labelled by atoms or $\emptyset$.

- This is the tree which is a node labelled by a: ‘a’.
- This is the tree with a node labelled by $\emptyset$: ‘$\emptyset$’.
- This is a tree with two daughters: ‘$\{\emptyset, a\}$’.

Let x, y vary over FM sets.

Define the FM set permutation action on sets by:

$$(b\,a)\cdot a = b \qquad (b\,a)\cdot b = a \qquad (b\,a)\cdot c = c$$

$$(b\,a)\cdot x = \{(b\,a)\cdot y \mid y \in x\}$$

I.e. ‘swap b and a in x’.

Note that $(b\,a)\cdot(x \setminus y) = (b\,a)\cdot x \setminus (b\,a)\cdot y$.

FM model of fresh

Write

$$a\#x \qquad \text{when} \qquad \text{И}b.(b\ a)\cdot x = x.$$

1. $a\#b$ is true. $(c\ a)\cdot b = b$ for $c \notin S = \{b\}$.
2. $a\#a$ is false. For no finite S is it the case that for $b \notin S$, $(b\ a)\cdot a = a$.
3. $a\#\mathbb{A}$ is true. $(c\ a)\cdot \mathbb{A} = \mathbb{A}$ always.
4. $a\#\mathbb{A}\setminus\{a\}$ is false. $(b\ a)\cdot(\mathbb{A}\setminus\{a\}) = \mathbb{A}\setminus\{b\}$.
5. $a\#\mathbb{A}\setminus\{b,c\}$ is true. Take $S = \{b,c\}$.

FM model of abstraction

Write

$$[a]x = \{(b, (b\ a) \cdot x) \mid b\#x \vee b = a\}.$$

This is the graph of a partial function which we can write as

$$[a]x = \lambda b. \begin{cases} (b\ a) \cdot x & b\#x \vee b = a \\ \bot & \text{otherwise.} \end{cases}$$

The ‘traditional’ model of abstraction is ‘a function-set’. A function-set is a set satisfying a predicate which expresses ‘this set is a set of pairs such that the first projections of two different pairs, are different’.

$[a]x$ is another model of abstraction which models α-equivalence.

The proof is in some facts:

$$\text{И}c.(c\ a)\cdot x = (c\ b)\cdot y \quad \text{if and only if} \quad [a]x = [b]y.$$

E.g. this gives us a rule:

$$\frac{\text{И}c.(c\ a)\cdot x = (c\ b)\cdot y}{[a]x = [b]y}$$

Also, $a\#[a]x$ and $a\#[b]x$ if and only if $a\#x$.

Substitution in the λ-calculus (FM)

Fix atoms. Let terms be defined by:

$$t ::= a \mid tt \mid \lambda[a]t.$$

Henceforth let t, u be a metavariable ranging over terms (terms are now abstract syntax trees in FM sets).

Define free variables 'magically' by:

$$fv(t) = \{a \mid \neg(a\#t)\}.$$

Substitution in the λ-calculus (FM)

Don't fix a choice of fresh variable for each finite set of variables.

Let substitution take an abstraction of a term and a term $sub([a]u, t)$. Sugar this to

$$u[a \mapsto t].$$

Define it by:

$$a[a \mapsto t] \equiv t$$
$$a[b \mapsto t] \equiv a$$
$$(uu')[a \mapsto t] \equiv (u[a \mapsto t])(u'[a \mapsto t])$$
$$(\lambda[b]u)[a \mapsto t] \equiv \lambda[b](u[a \mapsto t]) \qquad b\#t$$

Last clause can also be written as:

$$\forall a, t, \hat{u}.\ \text{И}b.\ (\lambda \hat{u})[a \mapsto t] \equiv \lambda[b]((\hat{u}b)[a \mapsto t]).$$

Suppose that:

- $\phi(a)$.
- If $\phi(t)$ and $\phi(t')$ then $\phi(tt')$.
- $И a.\forall u.\ \phi(u) \Rightarrow \phi(\lambda[a]u)$.

Then $\forall t.\ \phi(t)$.

We need only check $\phi(u) \Rightarrow \phi(\lambda[a]u)$ for fresh a and all u, since any abstraction $\hat{u}$ may be written as $[a]u$ for fresh a (and some u).

One question remains. Must we check

$$\forall u.\ \phi(u) \Rightarrow \phi(\lambda[a]u)$$

for cofinitely many a? Seems a bit boring really, all those indistinguishable fresh choices for a . . .

The principle of FM equivariance

For any predicate ϕ

$$\phi(x_1, \ldots, x_n) \Leftrightarrow \phi((b\ a) \cdot x_1, \ldots, (b\ a) \cdot x_n).$$

Proof:

1. $x \in y$ if and only if $(b\ a) \cdot x \in (b\ a) \cdot y$.
2. $x = y$ if and only if $(b\ a) \cdot x = (b\ a) \cdot y$.
3. $(b\ a) \cdot \mathbb{A} = \mathbb{A}$.

Now given

$$\phi(t)$$

we conclude

$$\phi((b\ a) \cdot t).$$

The principle of FM equivariance

For example in inductive reasoning we may write

> The property 't has the inductive hypothesis' has one free variable, t.
>
> So from our assumption that t has the inductive hypothesis, we also know that $(b\ a) \cdot t$ has the inductive hypothesis where b is chosen fresh.

We have formal license to rename variable names in inductive hypotheses.

We do not need to switch to induction on a coarser measure invariant under renaming, such as length.

Also, $a\#z \wedge \psi(a, z)$ if and only if $\text{И}a.\psi(a, z)$, so verifying ψ for one fresh atom is the same as verifying it for all fresh atoms.

A simple lemma

$$\forall a, t. \quad a\#t \quad \Longrightarrow \quad a\#(u[a \mapsto t]).$$

Immediate from general FM nonsense since $a\#[a]u$ and $a\#t$ so $a\#sub([a]u, t)$.

A simple lemma

Proof by induction on u:

- Suppose $u \equiv a$. Then $a[a \mapsto t] \equiv t$.
- Suppose $u \equiv b$. Then $b[a \mapsto t] \equiv b$.
- Suppose $u \equiv u'u''$. Then $(u'u'')[a \mapsto t] \equiv (u'[a \mapsto t])(u''[a \mapsto t])$. Now $a\#u'$ so $a\#(u'[a \mapsto t])$ by inductive hypothesis. Similarly for u''. Result follows.

Proof by induction on u:

Suppose $u \equiv \lambda[a']u'$ and suppose the inductive hypothesis of u':

$$\forall a, t.\ a\#t \quad \Longrightarrow \quad a\#(u'[a\mapsto t]).$$

So suppose $a\#t$. We want to show

$$a\#(\lambda[a']u')[a\mapsto t].$$

We would like to say ‘well,

$$(\lambda[a']u')[a\mapsto t] \equiv \lambda[a'](u'[a\mapsto t])$$

so we use the inductive hypothesis’.

But we can’t, because we do not know $a'\#t$.

A simple lemma

Choose a'' fresh (so $a''\#a', a, u, u', t$). Then $u \equiv \lambda[a''](a''\ a') \cdot u'$ and we have the inductive hypothesis of $u'' \equiv (a''\ a') \cdot u'$:

$$\forall a, t.\ a\#t \quad \Longrightarrow \quad a\#u''[a{\mapsto}t].$$

Then $(\lambda[a'']u'')[a{\mapsto}t] \equiv \lambda[a''](u''[a{\mapsto}t])$.

By inductive hypothesis $a\#(u''[a{\mapsto}t])$ and result follows.

π-calculus syntax may be specified as

$$P ::= 0 \mid \overline{a}b.P \mid a[b]P \mid \nu[a]P \mid (P|P) \mid \tau.P.$$

Of course, we can reason on this inductively! However, we can do more.

Let an action be:

$$\alpha ::= \tau \mid ab \mid \overline{a}b.$$

Let a (half-)transition be:

$$tr ::= [a](\alpha, P')$$

Write $P \stackrel{[a]\alpha}{\longrightarrow} P'$ for a pair (P, tr). Call this a transition.

Application to transition systems

Thus we represent name-generation by an FM abstraction, and coalgebraically $\longrightarrow$ may be represented as a function

$$\Pi \Longrightarrow \mathcal{P}[\mathbb{A}](Act \times \Pi)$$

In fact by magic this is equivalently

$$\Pi \Longrightarrow [\mathbb{A}]\mathcal{P}(Act \times \Pi).$$

Definition of bisimulation $\sim$

The normal definition of bisimulation is the greatest symmetric relation such that:

$$P{\sim}Q \;\Rightarrow\; \forall \alpha, P'.\, P \xrightarrow{\alpha} P' \;\Rightarrow\; \exists Q'.\, Q \xrightarrow{\alpha} Q' \wedge P'{\sim}Q'$$

Problem with traditional type for transitions; the so-called 'bound output'

$$\nu[b]\overline{a}b.P \xrightarrow{\overline{a}(b)} P$$

is not compatible with the definition above, since this transition is not simulated by

$$(\nu[b]\overline{a}bP.) \mid \nu[a]\overline{a}b$$

because it has 'junk' which mentions b. However our semantic intuitions suggest the two terms should be equivalent, please.

$$P \sim Q \;\Rightarrow\; \text{И} b.\forall \alpha, P'.\ P \xrightarrow{[b]\alpha} P' \;\Rightarrow\; \exists Q'.\ Q \xrightarrow{[b]\alpha} Q' \wedge P' \sim Q'$$

Model bound output by outputting a bound name.

$$\nu[b]\overline{a}b.P \xrightarrow{[b]\overline{a}b} P$$

is compatible with

$$(\nu[b]\overline{a}b.P) \mid \nu[a]\overline{a}b \xrightarrow{[b']\overline{a}b'} \overline{a}b'.(b'\ b) \cdot P \mid \nu[a]\overline{a}b$$

because

$$(\nu[b]\overline{a}b.P,\ [b]\Big(\overline{a}b,\ P\Big)) = (\nu[b]\overline{a}b.P,\ [b']\Big(\overline{a}b',\ (b'\ b) \cdot P\Big)).$$

Furthermore, we know as much about $(b'\ b) \cdot P$ as we do about P, by FM equivariance.

Conclusions

So, model name-generation by an FM abstraction, and unpack that abstraction with a И. This is compatible with the coalgebraic method. For more details, see “The π-calculus in Fraenkel-Mostowski”.

We also exhibit name-generation (a dynamic phenomenon) with name-abstraction in syntax. The only difference is that the abstraction lives on a type which we interpret as a transition and treat coinductively in the first case, and lives on a type which we interpret as syntax and treat inductively, in the second case.