Is substitution an abstract notion?

Murdoch J Gabbay

King's College London, UK, 13/3/2006

Thanks to Iman Poernomo (and Maribel Fernández)

Nominal Terms

Nominal Terms model abstraction (in the sense of α-equivalence) without functional abstraction (in the sense of β-equivalence).

Mathematically this is quite interesting!

It brings things into balance: β-equivalence has a well-understood semantics (sets and functions between them). Nominal Terms also have a well-understood semantics (Fraenkel-Mostowski sets).

Substitution

Substitution seems to sit in between α-equivalence and β-equivalence.

It uses abstraction

$$a[a \mapsto t] = b[b \mapsto t]$$

in the sense that atoms 'bound' by substitution.

But it is not as powerful as β-equivalence: it cannot represent functions.

I will talk about the theory and semantics of substitution, approached in 'nominal' style.

Syntax of nominal terms

Fix countably many atoms $a, b, c, \ldots$. Fix some term-formers f. Let π vary over finitely-supported bijections on atoms.

π is finitely supported when $\pi(a) = a$ for all atoms except for finitely many atoms.

Fix countably many unknowns X. These are morally 'unknown terms', but represented in the syntax. In short, X is a variable.

The syntax of nominal terms is given by:

$$t ::= a \quad | \quad \pi X \quad | \quad [a]t \quad | \quad \mathsf{f}(t, \ldots, t).$$

a is an atom. πX is an unknown term with a suspended permutation. $[a]t$ is an abstraction. $\mathsf{f}(t, \ldots, t)$ is a term!

λ-calculus

We can specify a signature of the λ-calculus with just three constants:

$$\lambda \qquad \mathsf{app} \qquad \mathsf{sub}.$$

(Sugar $\mathsf{app}(t, u)$ as tu and $\mathsf{sub}([a]t, u)$ as $t[a \mapsto u]$.)

We use these to write the rule

$$(\lambda[a]t)u \;=\; t[a \mapsto u].$$

Why use abstraction

The standard minimal theory of equality on nominal terms is given by the rule:

$$a\#t,\ b\#t \Rightarrow (a\ b)t = t.$$

Here $a\#t$ means that if a occurs in t then it does so under an abstraction. Think of $a\#t$ as a generalisation of $a \notin fn(t)$.

$fn(t)$ equals 'free names of t'.

$(a\ b)t$ is a and b swapped in t.

Call this theory of equality CORE.

For example:

The canonical example is

$$b\#X \Rightarrow [a]X = [b](b\ a)X.$$

Note that $(b\ a)[a]X = [b](b\ a)X$.

That is, we need permutations to handle renaming in the presence of unknowns X.

CORE is α-equivalence.

β

If we add this equality

$$(\lambda[a]t)u \;=\; t[a{\mapsto}u]$$

we get LAMBDA.

But of course there's a bit missing — the theory of substitution.

Theory of substitution

$$
\begin{array}{lcll}
a\#Z & \Rightarrow & Z[a\mapsto X] & = Z \\
 & & \mathsf{f}(Z_1,\ldots,Z_n)[a\mapsto X] & = \mathsf{f}(Z_1[a\mapsto X],\ldots,Z_n[a\mapsto X]) \\
b\#X & \Rightarrow & ([b]Y)[a\mapsto X] & = [b](Y[a\mapsto X]) \\
b\#X & \Rightarrow & X[a\mapsto b] & = (b\ a)X \\
 & & a[a\mapsto X] = X &
\end{array}
$$

Some notes

I switched from using meta-variables t, u to unknowns X, Y, Z. The axioms of the last slide should be understood as valid also for instantiating unknowns.

(This is no big deal.)

Note also the axiom $b\#X \Rightarrow X[a\mapsto b] = (b\,a)X$. This raises the question

> Q. If you can express swapping using substitution, why bother with all this swapping nonsense?
> A. Because $X[a\mapsto b]$ has one more term-former than X whereas $(b\,a)X$ does not. With swapping we can rename atoms to avoid capture, without increasing the size of a term.

Of course we could base an entire theory on substitution rather than renaming — but that would defeat our purpose.

Example equality

Suppose $a\#u$. Then

$$v[a\mapsto t][b\mapsto u] = v[b\mapsto u][a\mapsto t[b\mapsto u]]$$

is derivable. Remember that sub is just another term-former!

$$\mathsf{sub}([b](\mathsf{sub}([a]v, t)), u) = \mathsf{sub}([a](\mathsf{sub}([b]v, u)), \mathsf{sub}([b]t, u)).$$

Semantics

So now we have a nominal algebraic theory. What do the semantics look like?

Semantics

A concrete semantics is a Fraenkel-Mostowski set (that's sets with permutation actions built-in) which interprets sub and atoms to validate the axioms.

$$
\begin{array}{lcll}
a\#z & \Rightarrow & z[a\mapsto x] & = z \\
 & & \mathsf{f}(z_1,\ldots,z_n)[a\mapsto x] & = \mathsf{f}(z_1[a\mapsto x],\ldots,z_n[a\mapsto x]) \\
b\#x & \Rightarrow & ([b]y)[a\mapsto x] & = [b](y[a\mapsto x]) \\
b\#x & \Rightarrow & x[a\mapsto b] & = (b\ a)x \\
 & & a[a\mapsto x] & = x
\end{array}
$$

A set model of concrete substitution sets

Pick any 'normal' set $\mathcal{X}$ (e.g. $\mathcal{X} = \{\top, \bot\}$ or $\mathcal{X} = \{0, 1, 2, \ldots\}$). Let

$$\begin{aligned} \mathbb{A} \Rightarrow \mathcal{X} &= \{\kappa \mid И a, b.\, \kappa(a) = \kappa(b)\} \\ (\mathbb{A} \Rightarrow \mathcal{X}) \Rightarrow \mathcal{X} &= \{\tau \mid И a.\, \forall x \in \mathcal{X}.\, \tau\kappa = \tau(\kappa\{a \mapsto x\})\}. \end{aligned}$$

Then $(\mathbb{A} \Rightarrow \mathcal{X}) \Rightarrow \mathcal{X}$ is a concrete substitution set.

A set model of concrete substitution sets

$$\mathbb{A} \Rightarrow \mathcal{X} = \{\kappa \mid И a, b.\ \kappa(a) = \kappa(b)\}$$
$$(\mathbb{A} \Rightarrow \mathcal{X}) \Rightarrow \mathcal{X} = \{\tau \mid И a.\ \forall x \in \mathcal{X}.\ \tau\kappa = \tau(\kappa\{a \mapsto x\})\}.$$

Here

- $\kappa\{a \mapsto x\}(a) = x$ and $\kappa\{a \mapsto x\}(b) = \kappa(b)$.
- $И a.\ \Phi(a)$ holds when
  - $\Phi(a)$ is false of some finite (possibly empty) subset of $\mathbb{A}$, BUT
  - for all b not in that finite set, $\Phi(b)$ is true.

  Read this as 'Φ is true for a new/for most atoms'.

Set model of concrete substitution sets

What this means is:

- κ is a valuation, which must for most atoms be constant ('boring') but may vary (be 'interesting') on some finite set.
- τ is a function from valuations as described to elements of $\mathcal{X}$ which only 'cares' about the values of κ on some finite set.

In particular if τ does not care about the values where κ and κ' are interesting, then $\tau\kappa = \tau\kappa'$.

Set model of concerete substitution sets

So we can set

- $[\![a]\!] = \lambda\kappa.\kappa(a)$ (interpretation of atoms).
- $\mu[a{\mapsto}\tau](\kappa) = \mu(\kappa\{a{\mapsto}\tau\kappa\})$ (interpretation of substitution).

These validate the axioms. We give just one example:

$$[\![a]\!][a{\mapsto}\tau](\kappa) = (\kappa\{a{\mapsto}\tau\kappa\})(a) = \tau(\kappa).$$

Semantics (again)

Call a abstract substitution set a model which validates axioms to do with sub, but does not interpret atoms.

So it is a Fraenkel-Mostowski set (that's sets with permutation actions built-in) which interprets sub and validates the axioms:

$$
\begin{array}{lcll}
a\#z & \Rightarrow & z[a\mapsto x] & = z \\
 & & \mathsf{f}(z_1,\ldots,z_n)[a\mapsto x] & = \mathsf{f}(z_1[a\mapsto x],\ldots,z_n[a\mapsto x]) \\
b\#x & \Rightarrow & ([b]y)[a\mapsto x] & = [b](y[a\mapsto x])
\end{array}
$$

Function spaces

Suppose $\mathbb{X}$ and $\mathbb{Y}$ are two substitution sets. Set

$$\mathbb{X} \Rightarrow \mathbb{Y} \quad = \quad \{f \mid И a, b.\, ((a\ b) f) = f\}$$

Here

- f varies over functions from the underlying set of $\mathbb{X}$ to the underlying set of $\mathbb{Y}$, and
- $((a\ b) f)(X) = (a\ b)(f((a\ b)X))$
  (the conjugation action).

Function spaces

The conjugation action is very literal in its action.

For example if $\mathbb{X} = \mathbb{Y} = \mathbb{A}$ then...

if $f = \lambda x.a$ then $(a\,b)f = \lambda x.b$,

if $f = \lambda x.\begin{cases} b & x = a \\ c & x = b \\ c & x = c \end{cases}$ then $(a\,c)f = \lambda x.\begin{cases} b & x = c \\ a & x = b \\ a & x = a \end{cases}$.

Function spaces

Function spaces $f, g \in \mathbb{X} \Rightarrow \mathbb{Y}$ form an abstract substitution set!

The substitution action is defined by:

$$(f[a \mapsto g])X \;=\; И a'.\, (((a'\ a) f) X)[a' \mapsto gX].$$

An equivalent form is:

$$\text{If} \quad a \# g, X \quad \text{then} \quad f[a \mapsto g](X) = (f\,X)[a \mapsto gX].$$

Here $a \# g$ when $И b.\, (b\ a) g = g$.

Function spaces satisfy...

$$(\#\mapsto) \qquad a\#h \;\vdash\; h[a\mapsto f] = h$$

$$(sub\mapsto) \quad a\#g \;\vdash\; h[a\mapsto f][b\mapsto g] = h[b\mapsto g][a\mapsto f[b\mapsto g]]$$

Cartesian product

If $\mathbb{X}$ and $\mathbb{Y}$ are substitution sets then $\mathbb{X} \times \mathbb{Y}$ is a substitution set, with substitution action

$$(X, Y)[a \mapsto (X', Y')] = (X[a \mapsto X'], Y[a \mapsto Y']).$$

Some interesting example functions

One-step β-reduction mapping s to s' if $s \rightarrow_\beta s'$ is in $\Lambda \Rightarrow \Lambda$.

The function -$[a \mapsto t]$ mapping s to $s[a \mapsto t]$ is in $\Lambda \Rightarrow \Lambda$.

The substitution function -$[a \mapsto$-$]$ mapping s and t to $s[a \mapsto t]$ is in $(\Lambda \times \Lambda) \Rightarrow \Lambda$.

The context function $\lambda a.$- mapping s to $\lambda a.s$ is in $\Lambda \Rightarrow \Lambda$.

So is the function

$$s \text{ maps to } C[s]$$

for general (possibly binding) contexts $C[$-$]$.

This is a nice semantics for contexts.

Some more interesting example functions

The unordered datatype $\{a.t, b.u\}$ is represented by

$$f \;=\; \text{-}[a{\mapsto}t][b{\mapsto}u]$$

mapping x to $x[a{\mapsto}t][b{\mapsto}u]$ is in $\Lambda \Rightarrow \Lambda$.

This behaves like an unordered datatype with t located at a and u located at b. To retrieve the data it suffices to apply to a or b.

In-place update is function composition. For example if $g = \text{-}[a{\mapsto}t']$ then

$$f \circ g \;=\; \text{-}[a{\mapsto}t'][b{\mapsto}u].$$

Conclusions

Substitution is a fundamental operation. Using FM and Nominal techniques we have teased it free of syntax and function application.

We have investigated the functional properties of the collection of all models of substitution. It is an interesting world to explore.

New mathematical semantics possible for logic and programming? Perhaps!