

Substitution in Fraenkel-Mostowski sets

Murdoch J. Gabbay, Heriot-Watt University, Scotland

*Cambridge University, England
Saturday 2 December 2006*

Thanks to Thomas Forster

Names vs. variables

What are the properties of **names** a, b, c, \dots ?

- They are atomic entities: ' a ' has no internal structure.
- At any point in time we may have only finitely many of them: syntax is finite.
- They are **das Ding an sich**: ' $a = b$ ' is **false**.

... oh yes, and they turn up everywhere.

E.g. **pointers** name data in computer memory. **Procedure names** name the procedures they name. **Ports** (like http port 80) name services. **IP addresses** name computers. And so on.

Names vs. variables

What are the properties of **variables** x, y, z, \dots ?

- They may be substituted for: ' x ' implies the presence of $[x \mapsto t]$.
- At any point in time we may have only finitely many of them: syntax is finite.

- They are not **das Ding an sich**: ' $x = y$ ' may be true or false.

This is because x and y may be substituted for. Most formal languages do not provide access to the names of variables, i.e. they do not provide ' $x = y$ '.

... and they turn up in most formal languages of note, from **first-order logic** to **JAVA**.

Names vs. variables

So: A variable is a name with a substitution action.

Problem: what does that mean, mathematically?

Set-theoretic model of names

Got a problem? Model it in set theory!

Is there a set-theoretic model of names? Yes! Fraenkel-Mostowski (FM) sets [newaas].

Is there a set-theoretic model of variables? Yes, as of this talk!

In this talk I will:

- Explore the FM model of names.
- Exhibit a substitution function on FM names.
- Argue that by virtue of this function, **the FM model of names** along with the substitution action, is **a model of variables**.

Set-theoretic model of names

It is standard to use set theory as a foundation for mathematics.

Variables are pervasive in mathematics, and particularly in mathematical computer science. As computer programs and computer systems become more complex, it is necessary to use programs to construct them and logic to reason about them. Thus the reflective nature of the enterprise **forces** us to take variables, names, pointers, etc. — mathematically very seriously.

A **set-theoretic model of variables** is a significant step in foundations. It puts variables, and thus formal methods themselves, on a new kind of (set-theoretic) foundation.

Set-theoretic model of names

Variables are typically given meaning using a valuation, which maps variables to denotational elements. Typical quote: “ $\llbracket x \rrbracket_\sigma = \sigma(x)$ ”.

This means that:

- Variables do not exist in the denotation.
- Variables are purely syntactic; limited to that model (syntax).

In our model, variables **do** exist in the denotation, and since substitution is an operation on the set universe there is no need to assume a term model.

Furthermore, we can program on the names of variables. This is of interest to (meta-)programming.

Set-theoretic model of names

FM is an extension of Zermelo-Fraenkel set theory with atoms (ZFA). So let's talk about ZFA first.

In ZFA atoms $a, b, c, \dots \in \mathbb{A}$ extend the set universe with base elements aside from \emptyset .

a, b, c, \dots are extensionally empty, but not equal to each other or the empty set.

Example ZFA sets:

$$\{\} \quad \{a\} \quad \{\{\}, \{b\}\} \quad \{a, b, c, \dots\} = \mathbb{A}$$

ZFA atoms are atomic: equivariance

Define a **swapping action** $(a\ b)$ on the ZFA set universe $(a\ b) : \mathcal{U} \rightarrow \mathcal{U}$ by ϵ -induction:

- $(a\ b)a = b.$
- $(a\ b)b = a.$
- $(a\ b)c = c.$
- $(a\ b)X = \{(a\ b)x \mid x \in X\}$

For example:

$$(a\ b)\{a\} = \{b\} \quad (a\ b)\{\{a\}, c\} = \{\{b\}, c\} \quad (a\ b)\{a, b\} = \{a, b\}$$
$$(a\ b)(\mathbb{A} \setminus \{a\}) = (\mathbb{A} \setminus \{b\})$$

Equivariance

Note that:

- $x \in y$ if and only if $(ab)x \in (a b)y$.
- $x = y$ if and only if $(ab)x = (a b)y$.
- \mathbb{A} is equal to $(a b)\mathbb{A}$.

Equivariance

Therefore for any ZFA predicate $\phi(x_1, \dots, x_n)$ with free variables in $\{x_1, \dots, x_n\}$, we have by a simple induction on the syntax of the language of ZFA sets ($=, \in, \mathbb{A}, \perp, \Rightarrow, \forall$):

$$\phi(x_1, \dots, x_n) \quad \text{if and only if} \quad \phi((a \ b)x_1, \dots, (a \ b)x_n).$$

I say that this formally expresses the essential extensional property that **atoms are atomic**.

Atoms may also be compared for equality. $a = b$ is **false**.

The NEW quantifier \forall

If ϕ is a predicate in the language of ZFA (which is also the language of FM) then

$$\forall a.\phi$$

is:

$$\exists S \text{ finite set. } \forall a \in \mathbb{A} \setminus S. \phi.$$

$\forall a.\phi$ means

“ ϕ holds for all but finitely many atoms”

or

“ ϕ holds **mostly**”.

Freshness

Using \forall and swapping $(a\ b)$ it is possible to obtain a notion of ‘not in the free atoms of’.

Write:

$$a \# x \quad \text{when} \quad \forall b. (b\ a)x = x.$$

Read “ a fresh for x ”. That is for **most** atoms we can replace a by b (and b by a) and x will not notice. For example:

$\neg a \# a$	since	$\forall b. (b\ a)a = b$
$\neg a \# \{a, c\}$	since	$\forall b. (b\ a)\{a, c\} = \{b, c\}$
$\neg a \# (\mathbb{A} \setminus \{a\})$	since	$\forall b. (b\ a)(\mathbb{A} \setminus \{a\}) = (\mathbb{A} \setminus \{b\})$
$a \# \mathbb{A}$	since	$\forall b. (b\ a)\mathbb{A} = \mathbb{A}.$

FM sets: Finite support

FM set theory extends ZFA with a **finite support property** (**Fresh**):

$$\forall a. a \# x \quad \text{or in full} \quad \forall b. \forall a. (a \ b)x = x.$$

That is, for all x and **most** a and b , x cannot tell the difference between a and b . Write $\text{supp}(x)$ for $\{a \mid \neg a \# x\}$ and call this the **support of x** . Then ‘ $\text{supp}(x)$ is finite’ (just like syntax!). For example:

$$\begin{aligned} \text{supp}\{c\} &= \{c\} & \text{supp}(\mathbb{A} \setminus \{c\}) &= \{c\} \\ \text{supp}\{\{a\}, c\} &= \{a, c\}. \end{aligned}$$

FM sets: Finite support

Impose some ordering on \mathbb{A} , so

$$\mathbb{A} = \{a_1, a_2, a_3, \dots\}.$$

The **comb** does not have finite support:

$$\text{Comb} = \{a_1, a_3, a_5, \dots\}$$

For any finite set of atoms, there is:

- Some $a \in \text{Comb}$ outside that set.
- Some $b \notin \text{Comb}$ outside that set.

So $(a\ b)\text{Comb} \neq \text{Comb}$.

Intuition of FM sets

FM sets include all the usual sets of ZF set theory, such as

$$0 = \{\} \quad \text{and} \quad i + 1 = \{i, \{i\}\}.$$

Sets mentioning atoms are restricted to be (unions of) sets that **look like**

$$\{a_1, a_2, \dots, a_n\} \quad \text{or} \quad \mathbb{A} \setminus \{a_1, a_2, \dots, a_n\}$$

so for example

$$\{\{a_1\}, \{a_2\}, \dots, \{a_n\}\} \quad \text{or} \quad \{(a_0, a_1), (a_0, a_2), \dots, (a_0, a_n)\} \dots$$

... and **do not** look like **Comb**.

Intuition of FM sets

Note that

- $n\#\{a_1, \dots, a_k\}$ iff $n \in \{a_1, \dots, a_k\}$.
- $n\#(\mathbb{A} \setminus \{a_1, \dots, a_k\})$ iff $n \in \{a_1, \dots, a_k\}$.

So we might expect that $n\#X$ can be deduced by looking individually at the components of X .

a -orbits

Let us make that formal.

Suppose $\neg a \# u$. Write

$$|u|_a = \{(n \ a)u \mid n = a \vee n \# u\}.$$

Call $|u|_a$ the a -orbit of u . It is obtained by renaming a in u to be all possible fresh atoms.

For example

$$|a|_a = \{a, b, c, \dots\} \quad |(a, c)|_a = \{(a, c), (b, c), (d, c), (e, c), \dots\}.$$

Note $(c, a) = (c \ a)(a, c) \notin |(a, c)|_a$.

a -orbits

Theorem: Suppose $\neg a \# u$ and $\neg a \# u'$. If $|u|_a \cap |u'|_a$ has more than one element then $u = u'$.

So any Z can be expressed as a(n almost) disjoint union

$$Z = \bigcup_u \{Z \cap |u|_a \mid \neg a \# u\}.$$

$Z \cap |u|_a$ is one of the ‘components’ mentioned previously.

Note that $Z \cap |u|_a \neq \emptyset$ does not imply that $u \in Z$. For example

$$(\mathbb{A} \setminus \{a\}) \cap |a|_a = (\mathbb{A} \setminus \{a\}) \cap \mathbb{A} = (\mathbb{A} \setminus \{a\}) \neq \emptyset.$$

Crucial elements

Say that u is a -crucial in Z when

$$\neg a \# u \quad \text{and} \quad \neg a \# (Z \cap |u|_a).$$

For example:

- a is not a -crucial in \mathbb{A} , $\{b, c\}$, or $\mathbb{A} \setminus \{b, c\}$.
- a is a -crucial in $\{a\}$, $\{a, b, c\}$, and $\mathbb{A} \setminus \{a, b, c\}$.

Crucial elements

- (a, c) is c -crucial in $\{(a, c), (b, c), (d, c), \dots\}$.

$$|(a, c)|_c = \{(a, b), (a, c), (a, d), \dots\} \text{ and} \\ \{\dots\} \cap \{\dots\} = \{(a, c)\}.$$

So $\neg a \# \{(a, c)\}$.

It is not a -crucial.

$$|(a, c)|_a = \{(a, c), (b, c), (d, c), \dots\}. \\ a \# \{(a, c), (b, c), (d, c), \dots\}.$$

Write $C_a Z$ for the set of a -crucial elements in Z .

Lemma: $Z = \bigcup_{u \in C_a Z} (Z \cap |u|_a)$.

Lemma: $a \# Z$ if and only if $C_a Z = \emptyset$.

Symmetric difference

It is **not** the case that $C_a(X \cup Y) = C_a X \cup C_a Y$.

For example $C_a(\mathbb{A}) = \emptyset$ and $C_a(\{a\}) = C_a(\mathbb{A} \setminus \{a\}) = \{a\}$.

Write $X \Delta Y$ for the **symmetric difference** of X and Y :

$$X \Delta Y = \{u \mid (u \in X \wedge u \notin Y) \vee (u \notin X \wedge u \in Y)\}.$$

Lemma: $C_a(X \Delta Y) = C_a X \Delta C_a Y$.

So the natural notion of combination of sets, at least as regards crucial elements, is symmetric difference.

For example

$$C_a \mathbb{A} = \emptyset = \{a\} \Delta \{a\} = C_a(\mathbb{A} \setminus \{a\}) \Delta C_a(\{a\}).$$

Support

Theorem: $a \# X \Delta C_a X$ always.

So $C_a X$ is a measure of **reason** for $\neg a \# X$.

Note again: $u \in C_a X$ does **not** imply $u \in X$.

For example

$$a \in C_a(\mathbb{A} \setminus \{a\}) = \{a\}$$

but

$$a \notin \mathbb{A} \setminus \{a\}.$$

Substitution

But now it is fairly obvious what a definition of substitution should be.

It is $C_a Z$ that is the reason that a is not fresh for Z .

Therefore, to substitute a for x in Z is to substitute a for x in $C_a Z$.

Define:

$$Z[a \mapsto x] = (Z \Delta C_a Z) \Delta \{z[a \mapsto x] \mid z \in C_a Z\}.$$

Substitution

$$Z[a \mapsto x] = (Z \Delta C_a Z) \Delta \{z[a \mapsto x] \mid z \in C_a Z\}.$$

For example:

$$\{a\}[a \mapsto \emptyset] = \{\emptyset\} \quad \{a, b\}[a \mapsto \emptyset] = \{\emptyset, b\}$$

$$(\mathbb{A} \setminus \{a\})[a \mapsto \emptyset] = \mathbb{A} \cup \{\emptyset\}$$

$$(\mathbb{A} \setminus \{a\})[a \mapsto b] = \mathbb{A} \setminus \{b\} \quad \{a, b\}[a \mapsto b] = \emptyset.$$

Substitution

$$Z[a \mapsto x] = (Z \Delta C_a Z) \Delta \{z[a \mapsto x] \mid z \in C_a Z\}.$$

Actually, this is a lie. The construction is more subtle than that.

This is the last talk of the day. You want subtle, you shudn't'a put me in so late.

The papers:

- Substitution as a property of FM sets (substitution on sets).
- A nominal semantics of simple types (related but technically very different: substitution in simple types).

More to follow.

Worked examples

$$C_a\{a\} = \{a\}.$$

$$\{a\}[a \mapsto \emptyset] = \{a\} \Delta \{a\} \Delta \{\emptyset\} = \{\emptyset\}.$$

$$C_a(\mathbb{A} \setminus \{a\}) = \{a\}.$$

$$(\mathbb{A} \setminus \{a\})[a \mapsto \emptyset] = (\mathbb{A} \setminus \{a\}) \Delta \{a\} \Delta \{\emptyset\} = \mathbb{A} \cup \{\emptyset\}.$$

$$C_a\{\{a\}, \{a, b\}\} = \{\{a\}, \{a, b\}\}.$$

$$\{\{a\}, \{a, b\}\}[a \mapsto b] = \{\{b\}\}.$$

Substitution action?

What **is** a substitution action? Mathijssen and I have shown that the following axioms are a sound **and complete** axiomatisation of the properties of substitution in semantics [capasn], using nominal algebra (algebra of names [noma]):

$$(\alpha) \quad b\#x \Rightarrow x[a \mapsto y] = x[a \mapsto b][b \mapsto y]$$

$$(\# \mapsto) \quad a\#x \Rightarrow x[a \mapsto y] = x$$

$$(\mathbf{var} \mapsto) \quad a[a \mapsto y] = y$$

$$(\mathbf{id} \mapsto) \quad x[a \mapsto a] = x$$

$$(\mathbf{fin} \mapsto) \quad \text{supp}(x) \text{ is finite}$$

$$(\mathbf{ren} \mapsto) \quad b\#x \Rightarrow \text{supp}(x[a \mapsto b]) = \text{supp}(x)[a \mapsto b]$$

Semantics of variables

It is common to see variables as a property of syntax, with the interaction between variables given by a valuation.

In this presentation, variables are denotational elements with both axiomatic properties (above) and a set-theoretic realisation.

Obviously this gives a new formal handle on certain old questions, such as the relationship between variables and names.

Semantics of variables

Furthermore if variables \in denotation we can recruit semantic methods and constructions (graphs of functions, etc.) to get new logics and programming languages. This has much topical interest.

Thank you very much for your attention.