Fraenkel Mostowski for Names and Binding

Murdoch J. Gabbay, December 2002

Cambridge University, UK,
www.cl.cam.ac.uk/~mjg1003

What are names? What is binding? Answer: NOM.

NOM is the category of **Nominal Sets**.

- Originally a set theory FM **Sets** GabbayMJ:newaas-jv.
- Is equivalent to already-existing **Schanuel Topos**.
- Axiomatised in First-Order Logic (FOL) in **Nominal Logic** PittsAM:nomlfo-jv in TACS'01 Sendai (*Japan, Where* FM *News Breaks First!*).
- Presented and applied in GabbayMJ:thempc 2002.

It's a category of sets with a **permutation action** and **finite supporting set**:

NOM: Permutation Action

Choose some countably infinite set of **atoms** $a, b, c, n, m, \ldots \in \mathbb{A}$.

Let $P_{\mathbb{A}}$ be the set $\left\{\pi : \mathbb{A} \to \mathbb{A} \;\middle|\; \pi \text{ bijective}\right\}$.

For example, $(a\ b)$ such that $a \mapsto b$, $b \mapsto a$, and $u \mapsto u$ for $u \neq a, b$. Also **Id** such that $u \mapsto u$. This is a group under $\circ$ functional composition, with identity **Id**.

Then $X \in \mathrm{NOM}$ has an action $P_{\mathbb{A}} \to X \to X$ written $\pi \cdot x$ which satisfies

(1) $\quad \mathbf{Id} \cdot x \quad = x$

(2) $\quad \pi \cdot \pi' \cdot x \quad = (\pi \circ \pi') \cdot x.$

I.e. the standard rules of a permutation action.

NOM: Examples of permutation action

For example, $\mathbb{A}$ has a permutation action given by $\pi \cdot u = \pi(u)$. Also $\mathbb{A} \times \mathbb{A}$ has one given by $\pi \cdot \langle u, v\rangle = \langle \pi \cdot u, \pi \cdot v\rangle$.

$\mathcal{P}(X)$ has a permutation action given pointwise: $\pi \cdot U = \left\{\pi \cdot u \;\middle|\; u \in U\right\}$. Write $\mathcal{P}_{fin}(X)$ for the set of finite sets of X, this inherits that pointwise action.

$\mathbb{N} = \{0, 1, 2, \ldots\}$ and $\mathbb{B} = \{\top, \bot\}$ have trivial permutation actions given by $\pi \cdot x = x$ always.

The set of finite trees with labels has a permutation action given by the permutation action on the labels. If we model syntax of terms as finite trees labelled by tags (from $\mathbb{N}$, say) and atoms $a, b, c, \ldots \in \mathbb{A}$ for variable symbols, then the permutation action acts on a term by acting on the variable symbols in that term. More on that later.

Axiom: An $X \in \mathrm{NOM}$ has **finite supporting sets**:

(3) $\forall x \in X.\, \exists S \in \mathcal{P}_{fin}(\mathbb{A}).\, \forall a, b \in \mathbb{A}.$

$$a, b \notin S \implies (a\ b) \cdot x = x.$$

If we write $Fix(S) \stackrel{\text{def}}{=} \left\{\pi \;\middle|\; \forall s \in S.\, \pi(s) = s\right\}$ and $Stab(x) = \left\{\pi \;\middle|\; \pi \cdot x = x\right\}$ then this means

$$\exists S \ \underline{\text{finite}}.\, Fix(S) \subseteq Stab(x).$$

(Infinite such exists $\mathbb{A}$; **Id** $\cdot\, x = x$) Write $S\ supports\ x$ when $Fix(S) \subseteq Stab(x)$.

$$\exists S \text{ finite}.\ S\ supports\ x.$$

Finite supporting set.

NOM: Examples of objects 1/3

$\mathbb{A} \in \mathrm{NOM}$; a is supported by $\{a\}$.

$X, Y \in \mathrm{NOM} \implies X \times Y \in \mathrm{NOM}$ with $\pi \cdot \langle x, y \rangle = \langle \pi \cdot x, \pi \cdot y \rangle$. If S supports x and T supports y then $S \cup T$ supports $\langle x, y \rangle$.

$\mathbb{N} = \{0, 1, 2, \ldots\}$ and $\mathbb{B} = \{\top, \bot\}$ have trivial permutation actions $\pi \cdot x = x$ so every x has finite support $\emptyset$.

NOM: Examples of objects 2/3

$X \in \mathrm{NOM} \implies \mathcal{P}_{fin}(X) \in \mathrm{NOM}$. Pointwise action as described. If S_i supports x_i for each x_i in some finite $U \subseteq X$ then $\bigcup_i S_i$ is also finite and supports U.

$X \in \mathrm{NOM} \implies \{U \subseteq X \mid \exists S \in \mathcal{P}_{fin}(\mathbb{A}).\, S \; supports \; U\} \in \mathrm{NOM}$. This is the NOM powerset. It is not equal to the "external" one: we cut down to subsets of finite support.

In particular we can verify by calculation that $A \in \mathcal{P}(\mathbb{A})$ if and only if A is finite, with finite supporting set A, or A is cofinite ($\mathbb{A} \setminus A$ finite) with finite supporting set $\mathbb{A} \setminus A$. So

$$\mathcal{P}(\mathbb{A}) = \mathcal{P}_{fin}(\mathbb{A}) + \mathcal{P}_{cofin}(\mathbb{A}).$$

The set of finite trees with labels has a permutation action given by the permutation action on the labels. The support of a tree is the union of the supports of its (finitely many) labels. Thus for example:

(4) $$\Lambda \cong \mathbb{A} + \Lambda \times \Lambda + \mathbb{A} \times \Lambda.$$

Permutation action on labels:

(5) $$\pi \cdot a = \pi(a) \qquad \pi \cdot (t_1 t_2) = (\pi \cdot t_1)(\pi \cdot t_2)$$
$$\pi \cdot \lambda a t = \lambda(\pi(a))\pi \cdot t.$$

Lemma 1: Any $x \in X$ in NOM has a unique *smallest* supporting set. Call this the **support of** x.

Support of $t \in \Lambda$ is $n(t)$, the names in t (free or bound).

Proofs for α-equivalence $t =_\alpha t'$ are trees built up using the rules

(6)
$$a =_\alpha a \quad (\mathbf{Var})_a \qquad \frac{t_1 =_\alpha t_1' \quad t_2 =_\alpha t_2'}{t_1 t_2 =_\alpha t_1' t_2'} \quad (\mathbf{App})$$
$$\frac{t\{n/a\} =_\alpha t'\{n/a'\}}{\lambda a t =_\alpha \lambda a' t'} \quad (\mathbf{Lam})_n$$

where in $(\mathbf{Lam})_n$ there is a side-condition that $n \notin \{a, a'\} \cup n(t) \cup n(t')$ (the obvious support of the conclusion). So the valid proofs of $=_\alpha$ are an inductively defined subset of

(7)
$$\begin{array}{rll} T \cong & \mathbb{A} & (\mathbf{Var})_a \\ + & T \times T & (\mathbf{App}) \\ + & \mathbb{A} \times T \times T & (\mathbf{Lam})_n \end{array}$$

of “well-formed” trees schematically described by the rules above.

Equivariance

Theorem 2: If a relation is defined using rules invariant under permuting atoms, then the inductively defined set is itself invariant under permuting atoms.

Proof: Well-formedness of proof-trees is preserved, by assumption, under permuting atoms.

So we verify by simple inspection that all rules defining $=_\alpha$ are invariant under permutation, so $t =_\alpha t'$ if and only if $(a\ b) \cdot t =_\alpha (a\ b) \cdot t'$.

Proof by induction on proof-trees that

$$s =_\alpha s' \implies \forall s''.\,(s' =_\alpha s'' \implies s =_\alpha s'').$$

Consider just the case $s = \lambda a t$, $s' = \lambda a' t'$, $s'' = \lambda a'' t''$. Suppose we have two proofs

$$\frac{\dfrac{\pi_n}{t\{n/a\} =_\alpha t'\{n/a'\}}}{\lambda a t =_\alpha \lambda a' t'}\ (\mathbf{Lam})_n \tag{8}$$

$$\frac{\dfrac{\pi'_{n'}}{t'\{n'/a\} =_\alpha t''\{n'/a''\}}}{\lambda a' t' =_\alpha \lambda a'' t''}\ (\mathbf{Lam})_{n'}.$$

Observe that we can permute n and n' to entirely fresh m to obtain two valid proofs

(9)
$$\dfrac{\dfrac{\pi_m}{t\{m/a\} =_\alpha t'\{m/a'\}}}{\lambda a t =_\alpha \lambda a' t'} \quad (\mathbf{Lam})_m$$
$$\dfrac{\dfrac{\pi'_m}{t'\{m/a\} =_\alpha t''\{m/a''\}}}{\lambda a' t' =_\alpha \lambda a'' t''} \quad (\mathbf{Lam})_m.$$

Transitivity of $=_\alpha$ 3/3

Furthermore proofs of the inductive hypothesis are themselves trees and using Theorem 2 we deduce from the inductive hypothesis for $t\{n/a\}, t'\{n/a'\}$ the same hypothesis for $t\{m/a\}, t'\{m/a'\}$.

Since $t'\{m/a'\} =_\alpha t''\{m/a''\}$ we deduce $\lambda at =_\alpha \lambda a''t''$ as required.

This was a sketch of a type of reasoning which seems in practice to be *the* lemma people often need in practice: if the name is fresh, you can rename it without changing truth values so long as the proposition is invariant under permuting names in its parameter.

They always are: we just parametarise over all atoms. Call this **equivariance reasoning**.

So what is the support of the proof

(10)
$$\dfrac{\dfrac{\pi_m}{t\{m/a\} =_\alpha t'\{m/a'\}}}{\lambda a t =_\alpha \lambda a' t'} \quad (\mathbf{Lam})_m.$$

As a tree, it is the union of the supports of its components. However, there is an equivalence class of proofs for different m so long as m is fresh. This motivates the following definitions:

- Write $S(x)$ for the least set supporting x (Lemma 1).
- Write $[a]x$ for $\big\{\pi \cdot \langle a, x\rangle \;\big|\; \pi \in Fix(S(x) \setminus a)\big\}$.

Abstractions: Examples 1/3

$[a]x = \big\{\pi \cdot \langle a, x\rangle \;\big|\; \pi \in Fix(S(x) \setminus a)\big\}$.

Recall $\pi \cdot \langle a, x\rangle = \langle \pi(a), \pi \cdot x\rangle$.

$$\begin{array}{ll} S(a) = \{a\} & [a]a = \{\langle a, a\rangle, \langle b, b\rangle, \ldots\} \\ S(b) = \{b\} & [a]b = \{\langle a, b\rangle, \langle c, b\rangle, \ldots\} \\ S(t) = n(t) & [a]t = \big\{\langle b, (b\ a) \cdot t\rangle \;\big|\; b \notin n(t) \vee b = a\big\}\,. \end{array}$$

If we call the following proof κ_m:

(11) $$\dfrac{\dfrac{\pi_m}{t\{m/a\} =_\alpha t'\{m/a'\}}}{\lambda a t =_\alpha \lambda a' t'} \quad (\mathbf{Lam})_m$$

then $[m]\kappa_m$ is the equivalence class mentioned two slides ago.

Write $[\mathbb{A}]X$ for $\left\{[a]x \;\middle|\; a \in \mathbb{A},\ x \in X\right\}$. Then a datatype of proofs-up-to-equivalence can be written

(12) $$\begin{array}{rlr} T \cong & \mathbb{A} & (\mathbf{Var})_a \\ + & T \times T & (\mathbf{App}) \\ + & [\mathbb{A}](T \times T) & (\mathbf{Lam})_*. \end{array}$$

These are proofs of $=_\alpha$ up to choices of fresh atoms.

Abstractions: Examples 3/3

We can also simplify the syntax and be rid of $=_\alpha$ entirely:

$$\begin{array}{rlcl} \Lambda_\alpha \cong & \mathbb{A} & t ::= & a \\ + & \Lambda_\alpha \times \Lambda_\alpha & & t_1 t_2 \\ + & [\mathbb{A}]\Lambda_\alpha & & [a]t \end{array} \tag{13}$$

This is an inductive datatype of terms of Λ pre-quotiented by $=_\alpha$: $\Lambda_\alpha \cong (\Lambda / =_\alpha)$. But Λ_α is inductive.

```
bindable_type Name          (* names *)
;
datatype Lambda =           (* Lambda-terms *)
    Var of Name             (* a *)
  | App of Lambda*Lambda    (* t1 t2 *)
  | Lam of <Name>Lambda     (* lam a t *)
;

val rec subst : Name*Lambda*Lambda -> Lambda =
 fn (n,Var x,s) =>
          if n=x then Var x else s
  | (n,App t1 t2,s) =>
          subst(n,t1,s) subst(n,t2,s)
  | (n,Lam <a>t,s) =>
          Lam <a>(subst(n,t,s))
;
```

FM: the *nominal* approach

Standard formula: Have a type of names such as $\mathbb{N}$. Model variables using X or $\mathbb{N} \times X$ (de Bruijn and Name-carrying), or possibly $\mathbb{N} \to X$ (Higher-Order Abstract syntax, HOAS).
Deal with freshness using index sets of “known names”, or relegate them to the meta-level in the case of HOAS.
Deal with binding with difficulty.
FM **formula:** Work in the previous domain (e.g. sets) but with permutation actions and finite support (e.g. *nominal* sets).
Deal with freshness using support $S(-)$.
Deal with binding with $[\mathbb{A}]-$.
Also: И, @, abstractive functions, $\otimes$, . . .