Nominal algebra and sets representations of computer science foundations

Murdoch J. Gabbay

May 30, 2013

What?

This talk is based on the following thread of research:

- Nominal Semantics for Predicate Logic: Algebras, Substitution, Quantifiers, and Limits
  (15 pages)
- Stone duality for nominal Boolean algebras with New
  (15 pages)
- Stone duality for first order logic
  (32 pages)
- Semantics out of context: nominal absolute denotations for first-order logic and computation
  (56 pages)
- Representation and duality of the untyped lambda-calculus in nominal lattice and topological semantics, with a proof of topological completeness
  (86 pages)

Why?

That's nearly 200 pages of maths, and it's dense.

I put effort into the exposition but that can't make the maths be something it's not: it's tough stuff.

But just knowing that what I have done, can be done, is a useful message. The details are important but secondary.

(I can drive a car and use Google Maps on my smartphone, though I don't know the details.)

Nominal techniques

Based on nominal sets (advertisment: an excellent recent book by Pitts, and two survey papers by myself).

Broadly speaking, nominal sets are structures built out of numbers $1, 2, 3, \ldots$ and atoms $a, b, c, \ldots$.

Numbers are totally ordered in a canonical way in a line, whereas atoms are totally unordered and symmetric up to permutations.

Numbers are like 'data'; atoms are like 'pointers' or 'abstract symbols'. We can add numbers, multiply numbers, Gödel encode with numbers. We can compare atoms for equality ... and that's it.

(Set theorists: assume we are in Fraenkel–Mostowski sets. Category theorists: assume we are in the Schanuel Topos.)

Logic

What is logic? Logic is many things. But at its most elementary, Logic is an abstraction of powersets.

Given a set X we can form its powerset $pset(X)$. $U \in pset(X)$ if and only if $U \subseteq X$.

This gives Boolean algebra; a bounded distributive complemented lattice.

So we have:

- $\bot$ (emptyset $\varnothing \in pset(X)$),
- $\wedge$ (sets intersection $\cap \in pset(X)^2 \to pset(X)$), and
- $\neg$ (sets complement ${}^c \in pset(X) \to pset(X)$).

Every Boolean algebra can be concretely represented in a powerset (leads on to Stone duality).

Logic

You can do a lot just with Boolean algebra.

Think of propositional logic and SAT solvers.

Boolean algebras are big business and powersets are the canonical example.

Motivation map

Our motivation now goes like this:

Step 1. Nominal sets are a semantic foundation for computer science.

Step 2. First-order logic and the lambda-calculus are logical/computational foundations for computer science.

Step 3. So build first-order logic and the lambda-calculus in nominal sets using nominal powersets—just like we built predicate logic in ordinary sets using ordinary powersets.

Obviously we could replicate the usual constructions familiar from the ZF/HOL universe (based just on numbers $1, 2, 3, \ldots$). But that does not use the atoms $a, b, c, \ldots$.

We should try to find models of first-order logic (FOL) and the λ-calculus living already in nominal powersets.

It's more complicated than that, of course.

A fuller map is this:

- Powersets (sets of points; good for representing stuff).
- Lattices (these are posets with limits and stuff).
- Topologies (sets of points, subject to conditions on which sets are 'good').
- Algebras (sets and operations subject to equalities).
- Algebraic logic (syntax for specifying algebras).

Boolean algebras ($\bot$, $\wedge$, $\neg$) have a logical axiomatisation, a lattice characterisation, and a topological characterisation, and have simple canonical models in powersets.

It should be possible to do the same for $\forall$ and λ, using nominal algebra and lattices and topologies in nominal (power)sets.

Existing work is as follows:

- Nominal powersets.
- Lattices (posets with limits and stuff).
- Topologies ('good' sets of points).
- Nominal algebra (nominal terms axioms specifying equalities over nominal sets).

Nominal algebra syntax, logic, models, and theories for FOL and the λ-calculus were worked out with Maribel Fernandez and Aad Mathijssen and proved sound and complete.

The 200 pages cited above establish what the corresponding concrete sets operations, lattices, and topologies look are for $\forall$ and λ.

I'll now give a brief taste of what they look like.

Start with lattices and limits

Assume a distributive lattice $(\mathcal{X}, \leq, \wedge, \vee)$ in nominal sets. So in particular given $x, y \in \mathcal{X}$ we have a limit

$$x \wedge y \text{ the greatest element in } \{z \mid z \leq x,\ z \leq y\}.$$

Now $\mathcal{X}$ is a nominal set. So, given $x \in \mathcal{X}$ assume we have

$$\forall a.x \text{ the greatest element in } \{z \mid z \leq x,\ a\#z\}.$$

This is a finite limit, but its universal property is clearly nominal.

$\forall a.x$ is the largest element below x for which a is fresh.

We have leveraged the nominal foundation to characterise quantifiers as limits.

Surprisingly easy to do.

Obvious questions:

- How does this relate to quantifiers-as-adjoints?
- What are the properties of the (obvious?) nominal category theory that generalises this notion of limit?

Really good questions, especially for this audience.

Next: algebra

The axioms are not hard. $\forall a.x$ is characterised by nominal algebra axioms

$$\forall a.x \leq x \qquad a\#y \Rightarrow \forall a.(x \vee y) = (\forall a.x) \vee y.$$

Aad and I wrote these axioms back in 2006.

These are the usual axioms of 'forall', captured in the formal syntax and semantics of nominal terms and nominal algebra.

Aad and I were playing a game of trapping logical specifications in a formal framework which included an off-the-shelf nominal theory for handling names.

ZF sets and the languages arising from them are not able to do this off-the-shelf. Nominal techniques allow the treatment of names to scale.

Nominal algebra for substitution

In [capasn 2006,capasn-jv 2008] we specify a nominal algebra for substitution—σ-algebras:

$$
\begin{array}{lll}
(\sigma\mathbf{id}) & & x[a{\mapsto}a] = x \\
(\sigma\#) & a\#x \Rightarrow & x[a{\mapsto}u] = x \\
(\sigma\alpha) & b\#x \Rightarrow & x[a{\mapsto}u] = ((b\ a){\cdot}x)[b{\mapsto}u] \\
(\sigma\sigma) & a\#v \Rightarrow & x[a{\mapsto}u][b{\mapsto}v] = x[b{\mapsto}v][a{\mapsto}u[b{\mapsto}v]]
\end{array}
$$

These axioms usually appear as lemmas: e.g. $(\sigma\#)$ abstracts 'if $x \notin fv(t)$ then $t[s/x] = t$' is a lemma of syntax, and $(\sigma\sigma)$ abstracts the substitution lemma.

We characterise first-order logic as: a nominal lattice with fresh finite limits ($\wedge$ and $\forall$) and a monotone σ-algebra structure.

Properties like $\forall a.x \leq x[a{\mapsto}u]$ drop out (want a proof?), so $\forall a.x = \bigwedge_u x[a{\mapsto}u]$.

If you take standard models of first-order logic in ZF sets and translate them in a certain standard manner to nominal sets, then you find they have infinite limits—not just fresh finite ones. Valuation models are a strict subset of the nominal ones.

The $a\#$limit characterisation is not equivalent to the infinite limit $\bigwedge_u x[a{\mapsto}u]$, though in the models they coincide where both exist. This is because $\bigwedge_u x[a{\mapsto}u]$ depends on the size of the set of u, whereas an $a\#$limit does not and can be managed by creating a fresh atom a.

Shades here of the ($\forall$**L**) / ($\forall$**R**) rules.

Amgis algebras

Take a σ-algebra $\mathcal{X}$.

Its nominal powerset $pset(\mathcal{X})$ has a useful dual structure of an $\mathfrak{o}$-algebra (amgis-algebra):

$$\begin{array}{lll} (\mathfrak{o}\mathbf{id}) & & p[a{\leftarrow\!\shortmid}a] = p \\ (\mathfrak{o}\alpha) & b\#p \Rightarrow & p[u{\leftarrow\!\shortmid}a] = (b\ a){\cdot}(p[u{\leftarrow\!\shortmid}b]) \\ (\mathfrak{o}\sigma) & a\#v \Rightarrow & p[v{\leftarrow\!\shortmid}b][u{\leftarrow\!\shortmid}a] = p[u[b{\mapsto}v]{\leftarrow\!\shortmid}a][v{\leftarrow\!\shortmid}b] \end{array}$$

This looks like a σ-algebra, but the axioms are dualised.

$$p[u{\leftarrow\!\shortmid}a] = \{x \in \mathcal{X} \mid x[a{\mapsto}u] \in p\} \qquad p \in pset(\mathcal{X})$$

So $(\mathfrak{o}\sigma)$ comes about since $x[a{\mapsto}u][b{\mapsto}v] \in p$ if and only if $x[b{\mapsto}v][a{\mapsto}u[b{\mapsto}v]] \in p$ by $(\sigma\sigma)$.

From $\mathfrak{o}$ to σ

Take powersets again, and you get back to σ-algebras.

Lemma: If $\mathcal{X}$ is a σ-algebra, then $\mathit{pset}(\mathcal{X})$ is an $\mathfrak{o}$-algebra. If $\mathcal{Y}$ is an $\mathfrak{o}$-algebra then $\mathit{pset}(\mathcal{Y})$ is a σ-algebra.

Corollary: If $\mathcal{X}$ is a σ-algebra then so is $\mathit{pset}(\mathit{pset}(\mathcal{X}))$.

From $\mathfrak{o}$ to σ

Making this work is far from obvious. Let me give you just a flavour of why.

Suppose $\mathcal{P}$ is an $\mathfrak{o}$-algebra and $X \in \mathit{pset}(\mathcal{P})$.

Here is the definition of the σ-action from the λ-calculus paper:

$$X[a \mapsto u] = \{p \mid \mathit{I\!N}c.p[u \leftarrow\!\shortmid c] \in (c\ a) \cdot X\}$$

In addition, because of how duality theorems work, we may want $\mathcal{P}$ to be a set with a permutation action instead of a nominal set.

A lot of very careful design goes into these definitions.

Equality

Suppose $\mathcal{X}$ is a σ-algebra. We can interpret $u = v$ in $pset(pset(X))$ as

$$\{p \in pset(\mathcal{X}) \mid И c.p[u \leftarrow\!\shortmid c] = p[v \leftarrow\!\shortmid c]\}.$$

$И$ is the new-quantifier meaning 'for some/any fresh c'.

Let's drill down into how this works: Think of p as a maximal choice of elements of $\mathcal{X}$ which we declare to be 'valid at p'. Then

> $x \in p[u \leftarrow\!\shortmid c]$ *if and only if* $x[c \shortmid\!\rightarrow u] \in p$ *and* $x \in p[v \leftarrow\!\shortmid c]$ *if and only if* $x[c \shortmid\!\rightarrow v] \in p$.

So $(u = v)$ is the set of points p according to which $x[c \shortmid\!\rightarrow u]$ is valid if and only if $x[c \shortmid\!\rightarrow v]$ is valid, for any x.

$(u = v)$ is the set of points at which u and v are indistinguishable in any context x.

Equality

In nominal powersets, we can interpret not just $\forall$, but also $=$.

First-order logic with equality.

Foundation for mathematics and computer science.

I still think that's amazing.

I have great respect for people studying duality theorems and λ-calculus semantics.

It's the negative part that does it: in both λ-calculus and duality theory, if you strengthen this bit here then you have to weaken that bit there.

It's particularly fiddly to get the definitions right so all bits line up.

Some conclusions

There now exists an account of $\forall$ and λ as:

- axioms,
- nominal algebras,
- nominal lattices, and
- topological spaces in nominal sets.

We get soundness, completeness, and Stone-type duality results.

We also translate the traditional Tarski and the Herbrand models to the nominal framework.

Some conclusions

The resulting classes of models precisely capture the structure of $\forall$ and λ. For instance, the nominal lattices have fresh finite limits and that's exactly what we need, and no more.

The full proofs can be hard and finely-textured—as all such proofs are: completeness, duality, and models (especially of λ) are usually thus.

But, it's worth emphasising that the basic ideas and constructions are easily visible in $\mathit{pset}(\mathit{pset}(\mathcal{X}))$ for a σ-algebra $\mathcal{X}$.

I hope you will read these papers. If you do, just hold on (as I did) to what all this stuff actually does on $\mathit{pset}(\mathit{pset}(\mathcal{X}))$.

More on the λ-calculus, if there's time

Here's how to make λ work.

Assume that $\mathcal{X}$ has a combination action

$$\circ : \mathit{pset}(\mathcal{X}) \times \mathit{pset}(\mathcal{X}) \to \mathit{pset}(\mathit{pset}(\mathcal{X})).$$

Compare: ternary relations on Kripke models, and models of relevance logic, BI, and linear logic. Contrast: graph models of the λ-calculus.

Assume that atoms are a subset of $\mathcal{X}$.

The combination action acts pointwise to give a binary application function on sets of sets $\mathit{pset}(\mathit{pset}(\mathcal{X}))^2 \to \mathit{pset}(\mathit{pset}(\mathcal{X}))$. This has a right adjoint $\multimap$.

λ-calculus

So given X and Y we can form $X \bullet Y$ and $Y \multimap X$, and $X \bullet Y \subseteq Z$ if and only if $X \subseteq Y \multimap Z$.

Then $\lambda a.X$ can be identified with $\forall a.(\partial a \multimap X)$, where $\partial a = \{p \mid a \in p\}$.

β-reduction and η-expansion emerge from the adjoint properties of $\bullet$ and $\multimap$:

- $(\partial a \multimap X) \bullet \partial a \subseteq X$ leads to β-reduction, and
- $X \subseteq \partial a \multimap (X \bullet \partial a)$ leads to η-expansion.

More on filters, if there's time

Here's how to make filters work.

A filter in a nominal lattice $\mathcal{D}$ is a nonempty subset $p \subseteq |\mathcal{D}|$ (which need not have finite support) such that:

1. $\bot \notin p$.
2. If $x \in p$ and $x \leq x'$ then $x' \in p$.
3. If $x \in p$ and $x' \in p$ then $x \wedge x' \in p$.
4. If $И b.(b\ a) \cdot x \in p$ then $\forall a.x \in p$.

Maximal filters are used to build points, and points are used to build the topological representation of $\mathcal{D}$.

It's just $pset(pset(\mathcal{D}))$ but done in such a way that the result is isomorphic to $\mathcal{D}$.

Why the nominal models are not just ordinary models, if there's time

Tarski-style models of FOL can be converted into corresponding nominal structures. That is, valuation-based models have natural nominal and σ-algebra structure.

These models are complete; they have all limits, because the standard poset of Booleans $\{\bot, \top\}$ is complete and the Tarski denotation of a predicate ϕ is a function from valuations to $\{\bot, \top\}$.

The nominal models give ϕ a semantic in a nominal Stone space. Open and open compact sets are not closed under arbitrary intersections (i.e. do not have all limits).

They only have fresh-finite limits.

Precise characterisation of FOL.