Consistency of Quine's NF using nominal techniques

Murdoch J. Gabbay

October 22, 2014

Introduction

Thanks to the Leeds Logic Group for giving me the opportunity to be here today. Thank you all for coming.

Naive set theory has one rule; naive sets comprehension:

- If ϕ is a predicate then $\{a \mid \phi(a)\}$ is a set (the set of a such that ϕ).
  "Everything is a set, and a sets comprehension is a set."

This is inconsistent. Russell's famous 1901 paradox:

$$\{a \mid a \notin a\} \in \{a \mid a \notin a\} \quad \Leftrightarrow \quad \{a \mid a \notin a\} \notin \{a \mid a \notin a\}$$

Solutions

Various solutions proposed:

- Zermelo-Fraenkel set theory.
  Familiar to me as a PhD student as e.g. the category of sets, or Isabelle/ZF, and so on. ‘Proved’ consistent by the von Neumann cumulative hierarchy model; $\varnothing$, $\mathit{powerset}(\varnothing)$, ...
- Type Theory.
  Familiar as Higher-Order Logic, ML, and so on. ‘Proved’ consistent by taking sets and function-sets; ι, ι^ι, $(\iota^\iota)^\iota$, ι^{ι^ι}, ...
- Quine’s New Foundations (NF).

Why NF is quite nice

NF is a pretty theory. It admits a universal set: $\{a \mid \top\}$, the set of all sets, is a set.

- My favourite NF trick: model 2 as 'the set of all two-element sets'.
- Nicer than the standard, brutal, efficient, ZF model: $2 = \{\varnothing, \{\varnothing\}\}$.
- Nicer even than the Church numeral at type α: $2_\alpha = \lambda f{:}\alpha{\to}\alpha.\lambda x{:}\alpha.f(f(x))$.

The specification of NF is almost as concise as that of naive sets:

- If ϕ is a stratifiable predicate than $\{a \mid \phi(a)\}$ is a set. "Everything is a set, and stratifiable sets comprehension is a set."

Stratifiability

ϕ is stratifiable when there exists an assignment of a level to its variables such that:

- If $a{=}b$ appears in ϕ then $\mathit{level}(a) = \mathit{level}(b)$.
- If $a{\in}b$ appears in ϕ then $\mathit{level}(a){+}1 = \mathit{level}(b)$.

$a \notin a$ is unstratifiable; this blocks the comprehension of Russell's paradox.

How to semantically interpret stratifiability is part of the mystery of proving NF's consistency.

(cf. Thomas Forster's notion of stratimorphism. I will propose my own interpretation later in this talk.)

NF-the-theory versus NF-the-universe

I am not myself an NF-iste. I use Zermelo-Fraenkel set theory and Fraenkel-Mostowksi set theory for my proofs.

To me, NF is a logical theory to prove things about, not a universe to prove things in.

Thomas Forster and Randall Holmes are examples of NF-istes; they live and work in NF. I was exposed to this via Thomas, who taught me logic in university.

NF is important as a foundation with a universal set, and for its mysterious and fascinating stratifiability condition. Inconsistency of NF would tell us interesting things about the viability of having a universal set.

So consistency is an important question.

The case for NF

There are engineering arguments to take NF seriously.

ZF (and type theory) suffer from size issues. We get classes, hierarchies of universes, cardinality restrictions, and so forth. NF doesn't do that to you.

Type theory needs polymorphism (e.g. to define 'one' number 2). But polymorphism is like cocaine: you feel strong, but you always want more; then life gets really complicated and you easily end up worse than when you began. Thus when taken to its logical conclusion (such as COQ) hierarchies of universes reappear—this time as a proof-engineering issue.

NF has its own peculiarities, but it's a sweet spot.

Note: NFU (NF with urelemente) can be proved consistent relatively easily. But then we lose extensionaliity (not everything is 'a set').

About the proof

My consistency proof is in two halves:

- Sections 1 to 4 are standard computer science: build a datatype of syntax and do inductive proofs till we're sick.
  It's nominal abstract syntax and we use nominal algebra, and the definitions and proofs are subtle. But it's still just a bunch of inductions.
- Sections 5 to 8 contain the wild and crazy sets stuff: notably amgis-algebras and points.
  Amgis-algebras are modelled after previous topological duality results (cf earlier talk). Points are modelled after prime filters.

An interesting observation

I suspect the NF-istes find the first half of my paper harder to get to grips with than the second.

They are used to wild and crazy sets constructions. That is what they do; it does not frighten them.

But inductive arguments by subtle measures on complex datatypes are a relatively less common type of proof, to them. There seems to be some cultural difference.

High-level tour of the paper

Whatever your background, I do not propose to transmit a full technical account of the paper, today.

Instead, I will talk at a very high level about how the proof works.

Induction/coinduction

Consider some sets comprehension $\{a \mid \phi\}$.

- This has inductive structure, since ϕ is syntax.
- It has coinductive structure where $\{a \mid \phi\} \rightarrow \{b \mid \psi\}$ when $\{b \mid \psi\} \in \{a \mid \phi\}$.

So a model of NF sets can be viewed as finding a solution, write it *NF*, to this inductive/coinductive equation:

$$SyntaxFormers(NF) \longrightarrow NF \longrightarrow Powerset(NF)$$

NF is difficult because the right-hand side loops back to the left-hand side: everything is a set—including the set's behaviour!

This raises size issues which we deal with shortly. But there is another problem, with variables: we may need to dereference a variable, and we have no control over what gets put into it.

See (σ**elta**) of Figure 1 of my paper.

A peek into the paper: the sigma-action on syntax

$$
\begin{array}{lrl}
(\sigma\mathtt{and}) & & \mathtt{and}(\mathcal{X})[a{\mapsto}x] = \mathtt{and}(\{X[a{\mapsto}x] \mid X{\in}\mathcal{X}\}) \\
(\sigma\mathtt{neg}) & & \mathtt{neg}(X)[a{\mapsto}x] = \mathtt{neg}(X[a{\mapsto}x]) \\
(\sigma\mathtt{all}) & b\#x \Rightarrow & (\mathtt{all}[b]X)[a{\mapsto}x] = \mathtt{all}[b](X[a{\mapsto}x]) \\
(\sigma\mathtt{eltatm}) & a\#y, x \Rightarrow & \mathtt{elt}(y,a)[a{\mapsto}\mathtt{atm}(n)] = \mathtt{elt}(y[a{\mapsto}\mathtt{atm}(n)], n) \\
(\sigma\mathtt{elt}\mathbf{a}) & & \mathtt{elt}(y,a)[a{\mapsto}[a']X] = X[a'{\mapsto}y[a{\mapsto}[a']X]] \\
(\sigma\mathtt{elt}\mathbf{b}) & & \mathtt{elt}(y,b)[a{\mapsto}x] = \mathtt{elt}(y[a{\mapsto}x], b) \\
(\sigma[]) & c\#x \Rightarrow & ([c]X)[a{\mapsto}x] = [c](X[a{\mapsto}x]) \\
(\sigma\mathbf{a}) & & \mathtt{atm}(a)[a{\mapsto}x] = x \\
(\sigma\mathbf{b}) & & \mathtt{atm}(b)[a{\mapsto}x] = \mathtt{atm}(b)
\end{array}
$$

X ranges over predicate syntax, $\mathcal{X}$ over finite sets of predicates, a, b, c over distinct atoms, and x, y over sets syntax.

$\mathtt{and}$ is conjunction, $\mathtt{neg}$ negation, $\mathtt{all}$ quantification, $\mathtt{elt}(y, a)$ means '$y \in a$', $[a]X$ is sets comprehension and means '$\{a \mid \phi\}$', and $\mathtt{atm}(a)$ is syntax for 'the variable a'.

Intuition of stratifiability

Imagine that each time we dereference a variable, it costs us one stratifiability dollar.

My intuition for stratifiability is that ϕ comes with a number n of stratifiability dollars (equal to the difference between the highest- and lowest-level variables in a stratification of ϕ).

When we dereference a variable, we spend one dollar and n is decremented.

Eventually we run out of dollars and then we are trapped on the left-hand side, in syntax. We can no longer afford to look up variables, and we can reason inductively with variables as a base case.

See this happen in Lemma 4.9 and Proposition 5.12.

The fundamental equation: Sets = $[\mathbb{A}]$Predicates

NF has a universal set, so we might try to construct a model such that $X = \mathit{powerset}(X)$ (so that $X \in X$).

This raises size issues. We exploit nominal atoms-abstraction instead.

Sets comprehension is just a binder:

$$\{a \mid \phi\} \quad \forall a.\phi \quad \lambda a.t \quad \int f(a)\,da$$

So our semantics solves the equation:

$$\begin{aligned} \mathrm{NF} &= [\mathbb{A}]\mathrm{SemanticsOfPredicates} \\ [\![\{a \mid \phi\}]\!] &= [a][\![\phi]\!] \end{aligned}$$

$[a]$- is atoms-abstraction (cf my PhD). No size issues: if $\mathcal{X}$ is an infinite nominal set then $[\mathbb{A}]\mathcal{X}{=}\{[a]x \mid a{\in}\mathbb{A},\ x{\in}\mathcal{X}\}$ has same size as $\mathcal{X}$.

SemanticsOfPredicates

So we avoid size issues using atoms-abstraction and guarantee induction (in spite of the presence of some coinduction) with stratifiability.

Now what does SemanticsOfPredicates look like?

Consider first-order logic (FOL) and recall that a filter is a set of predicates that is consistent and deductively closed. Recall the standard completeness proof by giving FOL predicates semantics as sets of filters.

We do the same here: $[\![\phi]\!] \in$ SemanticsOfPredicates is a set of filters, for a suitable notion of filter (this notion is rather 'nominal', and requires subtle design).

So:

- $[\![\phi]\!] \in$ SemanticsOfPredicates is a set of filters.
- $[\![\{a \mid \phi\}]\!] = [a][\![\phi]\!] \in$ SemanticsOfSets.

Logic in nominal powersets

How to interpret logic on the powerset of all filters?

Conjunction and negation correspond to sets intersection and complement, as usual. Sets membership becomes substitution:

$$[\![\{b \mid \psi\} \in \{a \mid \phi\}]\!] = [\![\phi[a{:=}\{b \mid \psi\}]]\!].$$

What about quantification $[\![\forall a.\phi]\!]$?

It is known from previous work that if $\mathcal{X}$ is a nominal set with a substitution action, then so is *powerset*(*powerset*($\mathcal{X}$)) (cf. topological duality results for FOL and λ-calculus).

Brief sketch of sigma and amgis

Suppose $\mathcal{X}$ has a σ-action $x[a\mapsto u]$ and suppose $p \in \mathit{powerset}(\mathcal{X})$ and $X \in \mathit{powerset}(\mathit{powerset}(\mathcal{X}))$. Then we write:

$$\begin{aligned} x \in p[u\leftarrow a] &\Leftrightarrow x[a\mapsto u] \in p \\ p \in X[a\mapsto u] &\Leftrightarrow \text{И}a'.p[u\leftarrow a'] \in (a'\ a)\cdot X \end{aligned}$$

И is the NEW quantifier, meaning 'for all but finitely many'.

$[\![\phi]\!]$ is a set of filters, and filters are (almost) sets of predicates, so $[\![\phi]\!] \in \mathit{powerset}(\mathit{powerset}(\text{Predicates}))$. Since the predicate ϕ has a substitution action, so does $[\![\phi]\!]$.

Logic in nominal powersets

Thus general nominal abstract nonsense allows us to write the following:

$$[\![\forall a.\phi]\!] = \bigcap_u [\![\phi]\!][a \mapsto u]$$

In fact, an important lemma is that

$$[\![\phi]\!][a \mapsto u] = [\![\phi[a{:=}u]]\!].$$

See Lemma 5.30.

Logic in nominal sigma-powersets

The technical jargon:

Predicates are a nominal sigma-algebra (i.e. 'set with substitution'). The powerset of predicates forms an amgis-algebra, which is a dual notion. The powerset of the powerset of predicates restores us to the original sigma-powerset action.

It is a fact that sigma-powersets of amgis-algebras interpret first-order logic with equality in a natural way.

Furthermore, using nominal lattices and fresh-finite limits we can identify a subset of the full powerset that is necessary for modelling logic—avoids size issues.

(One caveat: we need a notion of quantifier depth to retain inductive reasoning principles.)

In summary:

- Sets comprehension is modelled in both syntax and semantics by nominal atoms-abstraction.
- Semantics is built in the sigma-powerset *powerset*(*powerset*(*Syntax*)).
- This set is large in general, but thanks to nominal lattice theory we can identify a small 'logical' subclass of it.
- Topological duality results in nominal sets give the semantics logical structure: conjunction, negation, substitution, and quantification.
- Stratifiability guarantees we can only 'flip' between syntax and semantics finitely often.
- Quantifier depth guarantees inductive reasoning in the presence of universal quantification.

Applications:

A hard but elementary proof of the consistency of NF. The proof is subtle, but systematic: its three components—a pile of inductions, a duality/filter construction, and some novel but off-the-shelf material on sigma-powersets—are nevertheless ‘elementary’.

A non-evident application of nominal techniques. Don’t read this as just being about NF: it’s about how nominal techniques can help us build models of logics.

A concrete interpretation of stratifiability. I’d love to see more made of this.

So . . . what else could this basket of techniques be applied to?

References

- Consistency of Quine's NF:
  Consistency of Quine's NF using nominal techniques.
  Get it at `gabbay.org.uk/papers.html#conqnf`.
- Logic in nominal sigma-powersets:
  Semantics out of context: nominal absolute denotations for first-order logic and computation.
  Get it at `gabbay.org.uk/papers.html#semooc`.