Stone duality for first-order logic (a nominal approach)

Murdoch J. Gabbay

Thanks to École Polytechnique and Dale Miller

December 14, 2011

Stone duality

Boolean algebras are algebras with conjunction and negation action satisfying certain axioms. The clopens of a topological space are naturally Boolean algebras, interpreting conjunction as intersection and negation as complement.

Stone duality expresses that Boolean algebras are dual to Stone spaces (totally separated compact topological spaces). This connects logic and topology.

It also gives concrete sets representations for Boolean algebras; from any Boolean algebra $\mathcal{B}$ we form a topological space by taking:

- points to be maximal filters $p \subseteq \mathcal{B}$ and
- open sets generated for each $x \in \mathcal{B}$ by $\{p \mid x \in p\}$.

FOL algebra

I developed a notion of 'FOL algebra'.

These nominal structures are to first-order logic as Boolean algebra is to propositional logic.

A FOL algebra is a nominal set equipped with functions on it, satisfying certain equalities.

A nominal set is a set equipped with names.

If you do category theory, we work in the Schanuel Topos (pullback-preserving presheaves on the category of finite sets and injections).

If you do set theory, we work in Fraenkel-Mostowski set theory.

If you don't know what that means, remember that every element x comes equipped with a finite supporting set $\mathrm{supp}(x)$ of atoms/names. This is structure that we just assume elements have.

What do we have to do to axiomatise first-order logic?

First-order logic has the following structure:

- A term-language with substitutional structure over itself $r[a{:=}t]$.
- Predicates with propositional structure $\wedge$ and $\neg$ and a substitution over terms $\phi[a{:=}t]$.
- A universal quantifier $\forall x.\phi$.

So now let's translate this to nominal algebraic axioms.

Term-language $\Rightarrow$ termlike σ-algebra

A termlike σ-algebra is a tuple $\mathcal{U} = (|\mathcal{U}|, \cdot, \mathsf{sub}, \mathsf{atm})$ where:

- $(|\mathcal{U}|, \cdot)$ is a nominal set; we may write this just $\mathcal{U}$;
- an equivariant substitution action $\mathsf{sub} : \mathcal{U} \times \mathbb{A} \times \mathcal{U} \to \mathcal{U}$, written infix $v[a{\mapsto}u]$; and
- an equivariant injection $\mathsf{atm} : \mathbb{A} \to \mathcal{U}$, usually written invisibly (so we write atm(a) just as a),

such that the following equalities hold:

Term-language $\Rightarrow$ termlike σ-algebra

$$
\begin{array}{lrl}
(\textbf{Subid}) & & x[a\mapsto a] = x \\
(\textbf{Sub}\#) & a\#x \Rightarrow & x[a\mapsto u] = x \\
(\textbf{Sub}\alpha) & b\#x \Rightarrow & x[a\mapsto u] = ((b\ a)\cdot x)[b\mapsto u] \\
(\textbf{Sub}\sigma) & a\#v \Rightarrow & x[a\mapsto u][b\mapsto v] = x[b\mapsto v][a\mapsto u[b\mapsto v]]
\end{array}
$$

Examples

1. The set of atoms $\mathbb{A}$ is a termlike σ-algebra where $\mathrm{atm(a)} = \mathrm{a}$ and $a[a{\mapsto}x] = x$ and $b[a{\mapsto}x] = b$.
2. The set $\mathbb{A} \cup \{*\}$ is a termlike σ-algebra where $\mathrm{atm(a)} = \mathrm{a}$, $a[a{\mapsto}x] = x$, $b[a{\mapsto}x] = b$, and $*[a{\mapsto}x] = *$.
3. First-order syntax generated by the grammar $r ::= a \mid \mathrm{f(r, \ldots, r)}$ for f drawn from some set of function symbols, is a termlike σ-algebra with $\mathrm{atm(a)} = \mathrm{a}$ and $r[a{\mapsto}s]$ is equal to r with a replaced by s.
4. Predicates of System F form a termlike σ-algebra, since they have predicate variables and substitution for predicates.
5. FOL predicates are not a termlike σ-algebra. This is because there are no predicate variables or substitution for predicates.

FOL algebra

Suppose $\mathcal{U} = (|\mathcal{U}|, \cdot, \mathsf{sub}, \mathsf{atm})$ is a termlike σ-algebra.

A FOL-algebra over $\mathcal{U}$ is a tuple $\mathcal{B} = (|\mathcal{B}|, \cdot, \wedge, \neg, \mathcal{U}, \mathsf{sub}, \forall)$ where $(|\mathcal{B}|, \cdot)$ is a nonempty nominal set which we may write just $\mathcal{B}$, and equivariant functions

- conjunction $\wedge : \mathcal{B} \times \mathcal{B} \to \mathcal{B}$ written $x \wedge y$ (for $\wedge(x, y)$),
- negation $\neg : \mathcal{B} \to \mathcal{B}$ written $\neg x$,
- substitution $\mathsf{sub} : \mathcal{B} \times \mathbb{A} \times \mathcal{U} \to \mathcal{B}$, and
- forall $\forall : \mathbb{A} \times \mathcal{B} \to \mathcal{B}$ written $\forall a.x$ (for $\forall(a, x)$),

such that the following equalities hold:

FOL algebra axioms

(**Subid**) to (**Sub**σ), and:

(**Sub**$\wedge$)		$(x \wedge y)[a{\mapsto}u] = (x[a{\mapsto}u]) \wedge (y[a{\mapsto}u])$
(**Sub**$\neg$)		$(\neg x)[a{\mapsto}u] = \neg(x[a{\mapsto}u])$
(**Sub**$\forall$)	$b\#u \Rightarrow$	$(\forall b.x)[a{\mapsto}u] = \forall b.(x[a{\mapsto}u])$
(**Commute**)		$x \wedge y = y \wedge x$
(**Assoc**)		$(x \wedge y) \wedge z = x \wedge (y \wedge z)$
(**Huntington**)		$x = \neg(\neg x \wedge \neg y) \wedge \neg(\neg x \wedge y)$
($\forall\alpha$)	$b\#x \Rightarrow$	$\forall a.x = \forall b.(b\ a)\cdot x$
($\forall$**E**)		$\forall a.x \leq x[a{\mapsto}u]$
($\forall\forall$)		$\forall a.\forall b.x = \forall b.\forall a.x$
($\forall\wedge$)		$\forall a.(x \wedge y) = (\forall a.x) \wedge (\forall a.y)$
($\forall\vee$)	$a\#y \Rightarrow$	$\forall a.(x \vee y) = (\forall a.x) \vee y$

Algebra

This is a pure algebraic structure. Just a nominal set with operations on it.

There are no valuations (similar to cylindric algebras). However, there is an axiomatised substitution action.

The axiomatisation is finite. Each axiom is a single expression in nominal algebra—not an infinite axiom-scheme.

The representation theorem

The next step is the duality result. To do this we must map $\mathcal{B}$ to some topological space $\mathcal{B}^\bullet$.

Here is a schematic outline of its kernel:

- We need a notion of filter; intuitively this is 'an up-closed set that is closed under intersections and does not contain $\perp$'. But now we need to account also for $\forall$.
- We need to build enough maximal filters (or points). In particular we need to know that every $x \in \mathcal{B}$ is contained in some point.
- We need to prove that there are enough open sets to discriminate points (our topology is no use if it has e.g. only two open sets).

Structure of a topology for first-order logic

There will be an underlying set of points $\mathcal{P}$.

There will be sets intersection and sets union on open sets.

The empty set will be open and represent 'false'. The entire underlying set of the topology will be open and represent 'true'.

Universal quantification and existential quantification will be interpreted as follows:

$$\begin{aligned} \mathsf{all}\,\mathsf{a}.\mathsf{X} &= \textstyle\bigcap_{u\in|\mathcal{U}|} X[a\mapsto u] \\ \mathsf{exist}\,\mathsf{a}.\mathsf{X} &= \textstyle\bigcup_{u\in|\mathcal{U}|} X[a\mapsto u] \end{aligned}$$

It remains to decide what "$X[a\mapsto u]$" should mean.

The correct notion of filter

Suppose $\mathcal{B}$ is a FOL-algebra. A filter is a finitely-supported subset $p \subseteq |\mathcal{B}|$ such that:

1. $\bot \notin p$
2. $\forall x, y.(x \in p \wedge y \in p) \Leftrightarrow (x \wedge y \in p)$.
3. $\mathsf{И}a.\forall x.(x \in p \Rightarrow \forall a.x \in p)$.

The first two conditions are standard. The third condition accounts for $\forall$. $\mathsf{И}$ is the new-quantifier meaning 'for fresh' or 'for all but finitely many'.

There is something strange; the natural condition corresponding to all would be $(\forall u {\in} |\mathcal{U}|.x[a {\mapsto} u] \in p) \Rightarrow \forall a.x \in p$. Remarkably, this apparently stronger condition follows from the condition above.

Quick sanity check

Let's just rehearse what we're doing.

There's a map Boolean algebras $\Rightarrow$ Stone spaces. We take maximal filters and the natural topology.

We're pulling the same stunt with FOL. To do this we used a nominal axiomatisation of FOL; the advantage of this is that we eliminate valuations and work just with a simple algebraic structure.

I haven't shown you any detailed proofs, but I have said what the correct notion of filter is to make all the proofs work.

Amgis-algebras

Now we need to decide what $X[a{\mapsto}u]$ means where X is an open set.

We give this pointwise meaning:

$$X[a{\mapsto}u] = \{p[u{\mapsfrom}a] \mid p \in X\}$$

where $[u{\mapsfrom}a]$ is an amgis-action, written $\reflectbox{$\sigma$}$-action.

Filters naturally have an $\reflectbox{$\sigma$}$-action.

Amgis-algebras

An $\rotatebox[origin=c]{180}{σ}$-algebra over $\mathcal{U}$ is a tuple $\mathcal{P} = (|\mathcal{P}|, \cdot, \rotatebox[origin=c]{180}{σ}, \mathcal{U})$ of an underlying nominal set $(|\mathcal{P}|, \cdot)$ which we may write just $\mathcal{P}$, and an amgis-action $\rotatebox[origin=c]{180}{σ} : |\mathcal{P}| \times \mathbb{A} \times |\mathcal{U}| \to |\mathcal{P}|$ written infix $p[u \mapsfrom a]$, such that:

$$
\begin{array}{lrcl}
(\textbf{Busid}) & p[a \mapsfrom a] & = & p \\
(\textbf{Bus}\sigma) & a\#v \Rightarrow p[v \mapsfrom b][u \mapsfrom a] & = & p[u[b \mapsto v] \mapsfrom a][v \mapsfrom b]
\end{array}
$$

- It is not necessarily the case that if p is a filter and $b\#p$ then $p[u \mapsfrom a] = ((b\ a)\cdot p)[u \mapsfrom b]$.
- It is not necessarily the case that if $a\#p$ then $p[u \mapsfrom a] = p$.

$\rotatebox[origin=c]{180}{σ}$-algebras are a partial dual to σ-algebras. Points (maximal filters) have a natural $\rotatebox[origin=c]{180}{σ}$-action given by:

$$p[u \mapsfrom a] = \{x \mid x[a \mapsto u] \in p\}$$

Back to sigma-algebras

Going in the other direction, from topology on ʊ-algebra to σ-algebra (by taking clopens), how do we restore (**Sub**α) and (**Sub**$\#$)?

Simplifying slightly, we impose these conditions on the notions of set that we build out of points. We build a σ-powerset out of those sets of points that satisfy (**Sub**α) and (**Sub**$\#$). Then (**Subid**) and (**Sub**σ) arise directly from the ʊ-action on points.

So (**Subid**) and (**Sub**σ) are 'pointwise' axioms, and (**Sub**α) and (**Sub**$\#$) are 'setwise' axioms.

This was a conceptual barrier that held me up for a while.

Topological conditions

The information I have given you so far suffices to prove a representation theorem. Every FOL-algebra $\mathcal{B}$ can be injected into the σ-powerset $pow(points(\mathcal{B}))$ of maximal filters of $\mathcal{B}$.

There remains the question of what notion of topological space corresponds to the images of FOL-algebras.

I will just state the main result:

- Call $\mathcal{U} \in pow(Opens_{\mathcal{T}})$ ∃-closed when $\text{И} a.\forall U.(U{\in}\mathcal{U} \Rightarrow \text{exist}\,\text{a.U}{\in}\mathcal{U})$ (dual to the third filter axiom).
- Call $\mathcal{U} \in pow(Opens_{\mathcal{T}})$ a cover when $\bigcup \mathcal{U} = ||\mathcal{T}||$.
- Call $\mathcal{U}$ a ∃-cover when $\mathcal{U}$ is a cover and is ∃-closed.
- Call $\mathcal{T}$ ∃-compact when every ∃-cover has a finite subcover.

The duality is between FOL-algebras and totally separated ∃-compact σ-topological spaces.

Conceptual map 1

Predicate	*Boolean Alg.*	*Usual sem. FO logic*	*Nominal semantics*
$\phi \wedge \psi$	$[\![\phi]\!] \cap [\![\psi]\!]$	$\lambda\varsigma.([\![\phi]\!]_\varsigma \cap [\![\psi]\!]_\varsigma)$	$[\![\phi]\!] \cap [\![\psi]\!]$
$\neg\phi$	$\mathcal{U}\backslash[\![\phi]\!]$	$\lambda\varsigma.(\mathcal{U}\backslash[\![\phi]\!]_\varsigma)$	$\mathcal{U}\backslash[\![\phi]\!]$
$\forall a.\phi$	n/a	$\lambda\varsigma.\bigcap_{x\in\mathcal{U}} [\![\phi]\!]_{\varsigma[a:=x]}$	$\bigcap_{x\in\mathcal{U}} [\![\phi]\!][a\mapsto x]$

$\mathcal{U}$ is some 'domain of points'. Boolean algebra does not have a universal quantification, whence the 'n/a'. Usual FOL semantics interprets $\forall$ using valuations ς; sets structure interacts indirectly with $\forall$.

Conceptual map 2

FOL-algebra	$\Rightarrow$	$\mathfrak{o}$-algebra by set of points (max. filters).
FOL-algebra	$\Rightarrow$	σ-topological space by $\{p \mid x \in p\}$ clopen.
σ-topological space	$\Rightarrow$	FOL-algebra by taking clopens.
$\mathfrak{o}$-algebra	$\Rightarrow$	FOL-algebra (by σ-powersets)

On complexity

Is this complicated?

Well, yes and no.

There are quite a few things stacked on top of each other here. Let's look at the conceptual stack:

- Nominal sets.
- Nominal algebras for σ-algebra, FOL-algebra, and ꝺ-algebra.
- The concrete constructions of the duality; notably filters, ꝺ-algebra action, σ-powerset, $\exists$-compactness, and $\forall$-Stone space.

That's a lot of stuff. Furthermore, not one of the elements above is obvious.

It's ab-so-lutely amazing how it all fits together.

On complexity

When we build the usual FOL semantics using valuations we assume a bunch of stuff too:

- Sets, so we can take an underlying domain $\mathcal{U}$.
- Higher-order functions, so we can build valuations of type $\mathit{Vars} \to \mathcal{U}$ and then semantics of predicates $(\mathit{Vars} \to \mathcal{U}) \to \{\bot, \top\}$.
- Inductively-defined syntax, α-equivalence, capture-avoiding substitution.
- The Stone duality proof itself; filters, powerset, and Stone space.

It seems to me that this is no simpler. More familiar, certainly. But simpler? Probably not.

Back to the high level

Why do this, aside from the fact that we can?

One of my long-term research goals is to argue that we took a wrong turn with valuations. Valuations make formal that 'a variable is something that refers to something in the denotation' and adds '(but is not itself in the denotation)'.

This is a very limited and restricted view. It made sense and was sufficient once upon a time, but not now.

Computer science is full of structures in which the internal pattern of references is itself important information. (Think of the π-calculus.)

So I am trying to argue that variables and their properties are a special case of referents, and referents are best understood in terms of denotational, not syntactic, behaviour.

Back to the high level

Now first-order logic is a (the) basic logic of the foundations of computer science.

(Second-order and higher-order logic are arguably already ‘set theory in disguise’—unless you do Henkin semantics, in which case see my recent paper with Dominic Mulligan on ‘nominal Henkin semantics’!)

To argue that valuations are just one way, and not necessarily the best way, of understanding variables, I need to show how the quantifiable variables of FOL can be approached as ‘nominal atoms plus extra structure’.

For me, this is the real breakthrough of this paper. For me, it definitively takes nominal atoms out of syntax and into FOL semantics.

Future work

Now we know it can be done, we can try to do it again. For instance:

- λa from the λ-calculus,
- second-order logic, higher-order logic,
- the propositional quantifier $\bigwedge\alpha$ from System F, and
- generalised quantifiers.