a-logic with arrows

Murdoch J. Gabbay, Heriot-Watt University, Scotland

WFLP, RDP, Paris
Monday, 24 June 2007

Joint work with Michael Gabbay, King's College, London

a-logic

Your mission is to axiomatise substitution, the λ-calculus, and first- (or higher-)order logic.

Your tool is first-order logic.

Go!

a-logic

Why is my mission to axiomatise substitution, the λ-calculus, and first- (or higher-)order logic?

Because they're at the heart of logic and programming.

You can do unification too, if you like.

Axioms of substitution

Axioms for substitution in first-order logic should look something like this:

$$x[x{:=}y] = y \qquad\qquad x{\neq}y \Rightarrow y[x{:=}z] = y$$

$$f(x_1, \ldots, x_n)[x{:=}y] = f(x_1[x{:=}y], \ldots, x_n[x{:=}y])$$

Here x, y, z, x_i are variables.

Axioms for substitution

$=$ is equality.

Now there is a problem.

'$x \neq y$' in

$$x{\neq}y \Rightarrow y[x{:=}z] = y$$

means that the values of x and y are not equal, and that's not what we meant when we wrote '$x \neq y$'.

So introduce a predicate $\mathtt{at}$ and let $\mathtt{at}(t)$ mean intuitively 't is a variable and it has not been associated to a value'.

That is, read $\mathtt{at}(t)$ as 't is a variable symbol'.

Inference rule for $\mathbf{at}$

$$\frac{[t \text{ not a variable}]}{\Gamma,\ \mathbf{at}\, t \ \vdash\ \Delta}\,(\mathbf{at}\,\mathbf{L})$$

Here is a valid derivation:

$$\frac{}{\mathbf{at}\,(2) \vdash 2+2=3}\,(\mathbf{at}\,\mathbf{L})$$

Axioms for substitution

Assume a ternary term-former $s\langle u \mapsto t\rangle$ explicit substitution and a binary predicate $\#$ freshness. Axioms are:

$$
\begin{array}{lcl}
a\#s & \Leftrightarrow & \mathbf{at}\, a \;\wedge\; \forall t. s\langle a \mapsto t\rangle = s \\
\mathbf{at}\, a & \Rightarrow & s\langle a \mapsto a\rangle = s \\
\mathbf{at}\, a & \Rightarrow & a\langle a \mapsto s\rangle = s \\
\mathbf{at}\, a \wedge b\#s & \Rightarrow & s\langle a \mapsto b\rangle\langle b \mapsto t\rangle = s\langle a \mapsto t\rangle \\
\mathbf{at}\, b \wedge a\#b \wedge a\#v & \Rightarrow & s\langle a \mapsto u\rangle\langle b \mapsto v\rangle = s\langle b \mapsto v\rangle\langle a \mapsto u\langle b \mapsto v\rangle\rangle
\end{array}
$$

Denotations

$\mathbf{at}$ is a unary predicate. The denotation of a unary predicate is uncontroversial; it identifies a subclass of the domain.

An easy denotation for a-logic is just a first-order structure; a set with elements and a subset of that to interpret $\mathbf{at}$. It's not hard.

We hypothesise term-formers such as $s\langle a{\mapsto}t\rangle$ and $a\#s$, and λ and whatever else pleases us, and write axioms for them.

A model is a first-order structure with functions, and stuff, to interpret the term-formers and predicates, and stuff. The usual story.

A catch

NOT.

Assume $\mathbf{at}\ a$ and consider $a\langle a{\mapsto}a\rangle$.

We can prove $a = a\langle a{\mapsto}a\rangle$. Traditionally equal things can be interchanged freely.

We assumed $\mathbf{at}\ a$. Do we want $\mathbf{at}\ (a\langle a{\mapsto}a\rangle)$?

No we do not; it is not a variable symbol — the top-level term-formers is explicit substitution. We can use $(\mathbf{at\ L})$.

(Later when we study the λ-calculus, we also have $\mathbf{at}\ ((\lambda a.a)a)$; the issue is not with substitution itself.)

A catch

Inference rules should be syntax-directed and give meaning to connectives independently of axioms. $(\mathbf{at\,L})$ does that.

This is incompatible with the usual treatment of equality, which can replace a term without a top-level term-former (a variable) with a term with a top-level term-former.

A solution

Orient equality:

$$\mathbf{at}\, a \Rightarrow a\langle a \mapsto a\rangle \rightsquigarrow a.$$

Think of this as a reduction relation. Call it ayquality.

Now $\mathbf{at}\ (a\langle a \mapsto a\rangle)$ can be false and $\mathbf{at}\ (a)$ can be true, and this is not a problem.

Inference rules

$$\frac{\Gamma, P \vdash Q, \Delta}{\Gamma \vdash P \Rightarrow Q, \Delta}(\Rightarrow \mathbf{R}) \qquad \frac{\Gamma \vdash P, \Delta \quad \Gamma, Q \vdash \Delta}{\Gamma, P \Rightarrow Q \vdash \Delta}(\Rightarrow \mathbf{L})$$

$$\frac{}{\Gamma, P \vdash P, \Delta}(\mathbf{Ax}) \quad \frac{}{\Gamma, \bot \vdash \Delta}(\bot\mathbf{L}) \quad \frac{\Gamma \vdash P, \Delta \quad \Gamma, P \vdash \Delta}{\Gamma \vdash \Delta}(\mathbf{Cut})$$

$$\frac{\Gamma \vdash P, \Delta \quad [a \notin \Gamma, \Delta]}{\Gamma \vdash \forall a.P, \Delta}(\forall\mathbf{R}) \qquad \frac{\Gamma, P[a{:=}t] \vdash \Delta}{\Gamma, \forall a.P \vdash \Delta}(\forall\mathbf{L})$$

Inference rules

$$\frac{[t \text{ not a variable}]}{\Gamma,\ \mathbf{at}\, t \ \vdash\ \Delta}\,(\mathbf{at}\,\mathbf{L}) \qquad \frac{\Gamma,\ \mathbf{at}\, a \ \vdash\ \Delta \qquad [a \notin \Gamma,\ \Delta]}{\Gamma \ \vdash\ \Delta}\,(\mathbf{Fresh})$$

$$\frac{}{\Gamma \ \vdash\ t \rightsquigarrow t,\ \Delta}\,(\rightsquigarrow\mathbf{R})$$

$$\frac{\Gamma,\ p(ts)[a{:=}s] \ \vdash\ \Delta \qquad [a{\downarrow}p(ts)]}{\Gamma,\ s' \rightsquigarrow s,\ p(ts)[a{:=}s'] \ \vdash\ \Delta}\,(\rightsquigarrow\mathbf{L}{\downarrow})$$

$$\frac{\Gamma,\ p(ts)[a{:=}s'] \ \vdash\ \Delta \qquad [a{\uparrow}p(ts)]}{\Gamma,\ s' \rightsquigarrow s,\ p(ts)[a{:=}s] \ \vdash\ \Delta}\,(\rightsquigarrow\mathbf{L}{\uparrow})$$

Arrowdown and arrowup

Once we orient equality we have to worry about whether an instance of the term occurs in positive or negative position.

Suppose that $s \rightsquigarrow t$ and $P[a{:=}s]$ implies $P[a{:=}t]$ as a result.

Then $P[a{:=}t] \Rightarrow Q$ implies $P[a{:=}s] \Rightarrow Q$.

So we have to keep track of positive and negative positions.

Arrowdown and arrowup

We must do this also at the level of terms.

If $t \rightsquigarrow t'$ and $s \rightsquigarrow t$ then $s \rightsquigarrow t'$. The right-hand side of $\rightsquigarrow$ is positive.

If $t \rightsquigarrow t'$ and $t' \rightsquigarrow u$ then $t \rightsquigarrow u$. The left-hand side of $\rightsquigarrow$ is negative.

So term-formers take an arity which is not just a number, but a list of directions.

The arity of $\rightsquigarrow$ is $(\uparrow, \downarrow)$.

The arity of $\mathbf{at}$ is $(\downarrow)$.

Arrowdown and arrowup

Define $a{\uparrow}P$, $a{\downarrow}P$, and $a\circlearrowleft P$ by:

$$\frac{a\circlearrowleft P}{a{\uparrow}P} \quad \frac{a\circlearrowleft P}{a{\downarrow}P}$$

$$\frac{a{\uparrow}P \quad a{\downarrow}Q}{a{\downarrow}(P{\Rightarrow}Q)} \quad \frac{a{\downarrow}P \quad a{\uparrow}Q}{a{\uparrow}(P{\Rightarrow}Q)} \quad \frac{a\circlearrowleft P \quad a\circlearrowleft Q}{a\circlearrowleft(P{\Rightarrow}Q)}$$

$$\frac{a{\uparrow}P}{a{\uparrow}\forall a.P} \quad \frac{a{\downarrow}P}{a{\downarrow}\forall a.P} \quad \frac{a\circlearrowleft P}{a\circlearrowleft\forall a.P}$$

Our axioms, again

$$
\begin{array}{lcl}
a\#s & \Leftrightarrow & \mathbf{at}\, a \ \wedge\ \forall t. s\langle a \mapsto t\rangle \rightsquigarrow s \\
\mathbf{at}\, a & \Rightarrow & s\langle a \mapsto a\rangle \rightsquigarrow s \\
\mathbf{at}\, a & \Rightarrow & a\langle a \mapsto s\rangle \rightsquigarrow s \\
\mathbf{at}\, a \wedge b\#s & \Rightarrow & s\langle a \mapsto b\rangle\langle b \mapsto t\rangle \rightsquigarrow s\langle a \mapsto t\rangle \\
\mathbf{at}\, b \wedge a\#b \wedge a\#v & \Rightarrow & s\langle a \mapsto u\rangle\langle b \mapsto v\rangle \rightsquigarrow s\langle b \mapsto v\rangle\langle a \mapsto u\langle b \mapsto v\rangle\rangle
\end{array}
$$

Shallow embedding

$$\mathbf{at}\, a \wedge b\#s \Rightarrow \lambda a.s = \lambda b.s\langle a{\mapsto}b\rangle \qquad \mathbf{at}\, a \Rightarrow (\lambda a.s)\cdot t \rightsquigarrow s\langle a{\mapsto}t\rangle.$$

So the λ-calculus can be embedded in a-logic with arrows.

This is not a deep embedding; the notion of equality is not syntactic identity, or even α-equivalence.

It is a shallow embedding. Terms are ayqual up to $\alpha\beta$-equivalence.

What is really going on here?

We are trying to reconcile the difference between syntactic identity and semantic identity.

Usually this is handled by types:

$\mathtt{Expr} \longrightarrow \mathbb{N}$ for an evaluation function, for example. $Prop$ is a type of propositions, o is a type of truth-values representing the denotation of elements in $Prop$, for example.

Somehow, the property of 'being a variable' doesn't sit terribly well with types.

What is really going on here?

a-logic internalises just enough of this property to give reasonably sensible shallow embeddings. Interestingly a notion of reduction is forced on us.

This is not a finished work. There is much to explore. (See my other papers.)