Nominal Terms with a Hierarchy of Variables
Or . . . **when** are unknowns?

Murdoch J. Gabbay

Joint work with Giulio Manzonetto and Antonino Salibra

Technical University Eindhoven, 31 October 2005.

Health warning:

I’m still busy inventing this stuff.

This material probably consists mostly of errors.

If this doesn’t make sense, tell me — and wait for the paper.

Thanks for finding time to come.

Motivation

'Normal' substitution is capture-avoiding on bound variables. E.g.

$$(\forall x.\ x = y)[y \mapsto x] \equiv \forall x'.\ x' = x \qquad (\lambda x.(\lambda y.xy))y \to \lambda y'.yy'$$

'Context' substitution is not. E.g.

$P \cong Q$ when in all process contexts C, $C[P]\downarrow$ if and only if $C[Q]\downarrow$.

Normally these are understood as phenomena related purely to syntax. I would like some semantic account. (This means technically that I have to give x a semantics independently of some ambient evaluation to closed terms, and a similar but not identical one to the hole $[\text{-}]$ in $C[\text{-}]$, as well as to C itself.)

$\forall$ and $\exists$ from first-order logic have symmetric intro-rules:

$$\frac{\Gamma \vdash P, \Delta}{\Gamma \vdash \forall x.\, P, \Delta}(\forall R) \quad (x \notin \Gamma, \Delta) \qquad \frac{\Gamma, P \vdash \Delta}{\Gamma, \exists x.\, P \vdash \Delta}(\exists L) \quad (x \notin \Gamma, \Delta)$$

$$\frac{\Gamma, P[x{\mapsto}s] \vdash \Delta}{\Gamma, \forall x.\, P \vdash \Delta}(\forall L) \qquad \frac{\Gamma \vdash P[x{\mapsto}s], \Delta}{\Gamma \vdash \forall x.\, P, \Delta}(\exists R)$$

There are explanations of where these symmetries come from, a great example of which is adjunctions in category theory. These presume a typed environment and introduce, in effect, functions — but weren't we doing first-order logic?

I want something inherently type- and function-free. I also want to decompose $\forall$ and $\exists$ into a simpler self-dual quantifier (self-dual meaning that the intro-rules on left and right are identical except for the side of the sequent they act on).

Motivation

Records and unstructured datatypes, for example customer.ID : $\mathbb{N}$ are generally modelled using either table lookups, lists, or functions. I want a notion of unstructured data which is *atomic*, that is, relies on *no implementational overhead*. This exists, e.g. in the Cardelli-Abadi object calculus, but I want it in a generic framework which is not specifically tailored to this one job.

I also want to model component-based semantics which involve graphs being substituted into other graphs, possibly with rewiring of edges during the substitution. Again, I want this with no specific implementational overhead (e.g., explicitly modelling graphs!).

Can we do this?

Is there a single system which will exhibit all of these disparate phenomena as aspects of a single, preferably rather elementary, system? Can we give this system a simple semantics.

Yes.

Very simple. I'll give an equational system (the only judgement is equality $s = t$).

If you know Nominal Algebraic Specifications (work with Aad Mathijssen), you can think of what you are about to see as NAS on steroids and speed. (Though that is only a first approximation.)

Assume base data sorts δ, one of which is propositions o. Sorts σ, τ and arities ρ are:

$$\sigma, \tau ::= \delta \mid [\sigma]\sigma \qquad \rho ::= (\sigma_1, \ldots, \sigma_n)\sigma.$$

Here n may equal zero.

For $i \geq 1$ and sort σ assume variable symbols $a^i_\sigma, b^i_\sigma, c^i_\sigma$ of level i and sort σ — we may drop the annotations.

Assume term-formers $f : \rho$.

Then terms are:

$$s, t, u, v ::= a^i_\sigma \mid f_{(\sigma_1,\ldots,\sigma_n)\sigma}(s_{\sigma_1}, \ldots, s_{\sigma_n}) \mid ([a^i_\sigma]s_\tau)_{[\sigma]\tau} \mid \text{И} a_\sigma.s_o.$$

Or, without annotations:

$$s, t, u, v ::= a \mid f(s, \ldots, s) \mid [a]s \mid \text{И} a.s.$$

Sorts and terms

$$s, t, u, v ::= a^i_\sigma \mid f_{(\sigma_1,\ldots,\sigma_n)\sigma}(s^{i_1}_{\sigma_1}, \ldots, s^{i_n}_{\sigma_n}) \mid ([a^i_\sigma]s_\tau)_{[\sigma]\tau} \mid \text{И} a_\sigma.s_o.$$

Assume term-formers:

- $\supset_{(o,o)o}$ implication.
- $=_{(\sigma,\sigma)o}$ equality (one for each σ).
- $\bot_{()o}$ false.
- $\sigma_{([\sigma']\sigma,\sigma')\sigma}$ explicit substitution (one for each σ, σ').

Use standard sugar. $[a]s$ is called abstract a in s, $\text{И}a.s$ is new (or fresh) a in s. И binds, abstraction does not. Write $s[a{\mapsto}t]$ for $\sigma([a]s, t)$.

Some examples

1. $\bot_o$ is a truth value. We know what it is and it represents itself.
2. a_o^1 is a variable. It represents a truth value we do not know today, but we will learn whether it is $\top \equiv \bot \supset \bot$ or $\bot$, tomorrow.
3. X_o^2 is also a variable. We will only find out what it is this evening — we may, for example, learn that it is $\bot$, but we may also learn that it is a.
4. $\mathcal{X}_o^3$ is also a variable. We will find out what it is, oh, sometime early this afternoon.
5. a_o^{1000} is also a variable. Let's find out what it is right now. Any suggestions?

It's lonely standing up here all alone. Look at my sad face. Make me smile!

Some examples

- $[a]a_o$ is, well never mind what it is, but write it $*$. It lives in $[o]o$.
- $[a]b$ is just b.
- $[a]X$ waits to find out what X is; if X becomes a then it becomes $[a]a$, if X becomes b it becomes $[a]b$, if X becomes $\bot$ it becomes $[a]\bot$, and so on.
- $f(s_1, \ldots, s_n)$ is f applied to s_1 to s_n. No tricks. However, f may become something else as a function of its arguments becoming something else.
- $\text{И}a.s$ generates a *fresh* a. This will never become anything, but we can use it as a 'generic unknown', e.g. to build $[a]a$ or $[a]X$.

Some examples

- $a[a{\mapsto}b]$ is b. $X[a{\mapsto}b]$ is X which has been told that a maps to b. This evening when X becomes something, the substitution $[a{\mapsto}b]$ will pounce on it.
- $X[a{\mapsto}Y]$ *is* allowed by the syntax. We simply learn what X and Y become, and whatever the substitution becomes acts on whatever X becomes. No sweat.
- $a[X{\mapsto}b]$ is a. By the time a becomes anything else, X is long gone.

$\supset$, $=$, and $\perp$ are as usual.

Axioms

We now say that all in axioms:

$$P \supset (Q \supset P) = \top \qquad (Q \supset R) \supset (P \supset Q) \supset (P \supset R) = \top$$

$$\neg\neg P = P$$

$$f(a_1[a \mapsto x], \ldots, a_n[a \mapsto x]) = f(a_1, \ldots, a_n)[a \mapsto x]$$

$$([a^i]x)[b^j \mapsto y] = [a^i](x[b^j \mapsto y]) \qquad j > i$$

$$a^i \# y \supset (([a^i]x)[b^j \mapsto y] = [a^i](x[b^j \mapsto y])) = \top \qquad i \leq j$$

$$([a]x)[a \mapsto y] = [a]x$$

$$(b\#x \supset [a]x = [b](x[a \mapsto b])) = \top$$

$$(b\#x \supset x[a \mapsto b][b \mapsto y] = x[a \mapsto y]) = \top$$

$$a[a \mapsto x] = x \qquad b^i[a^i \mapsto x] = b$$

Axioms

...just a few more:

$$a\#a = \bot \qquad (\text{И}y.a\#(X[x\mapsto y])) = a\#[x]X$$

$$(\text{И}a.\bot) = \bot$$

$$((\text{И}a.P) \supset Q) = (\text{И}a.(P \supset Q)) \quad a \notin Q$$

$$(P \supset (\text{И}a.Q)) = (\text{И}a.(P \supset Q)) \quad a \notin P$$

$$(\text{И}a.P)[b\mapsto Q] = \text{И}a.(P[b\mapsto Q]) \quad a \notin Q$$

$$(\text{И}a.P) = (\text{И}a.a\#x \wedge P) \qquad (\text{И}a.P) = (\text{И}a.x\#a \wedge P)$$

$$(a\#Y \supset (Y = Y[a\mapsto X])) = \top \qquad (X = X) = \top$$

$$((X = Y) \wedge C[X]) = ((X = Y) \wedge C[Y])$$

Our first definition

Use $a\#s$ as a macro for $(\text{И}c.s[a\mapsto c]) = s$ and say a is fresh for s.

As a term-former $\#$ would have arity $([\sigma]o)o$ (one $\#$ for each σ).

Intuitively it is clear (*is it?*) that $a\#s$ means 'a does not occur unabstracted in s'. This statement may transcend syntactic fact, e.g. $a\#X$ is an assertion about what happens this evening.

Then the conditions $a\#x$ scattered about the axioms are simply 'capture-avoidance' conditions.

Implementing $\forall$ and $\exists$

$$\forall a.\,\phi \equiv \mathsf{N}c.(a\#\phi \wedge \phi[a\mapsto c]) \qquad \exists a.\,\phi \equiv \mathsf{N}c.(a\#\phi \supset \phi[a\mapsto c])$$

These have the expected behaviour, for example:

$$\mathsf{N}c.(a\#\phi \wedge \phi[a\mapsto c]) \supset \phi[a\mapsto s] \quad =$$
$$\mathsf{N}c.(a\#\phi \wedge (\phi[a\mapsto s]=\phi) \wedge (\phi[a\mapsto c]=\phi) \wedge \phi[a\mapsto c]) \supset \phi[a\mapsto s] \quad =$$
$$\mathsf{N}c.\top = \top.$$

It is possible (and quite interesting!) to verify that $\forall a.\,\phi = \neg\exists a.\,\neg\phi$.

Implementing the λ-calculus (algebraically!)

Introduce a term-former $\cdot_{([\sigma']\sigma,\sigma')\sigma}$ and an axiom

$$([a]X) \cdot Y = X[a \mapsto Y].$$

The rest of the system takes care of substitution.

Also possible to directly implement the NEW calculus of contexts, i.e. to add in И and abstract over the full hierarchy of variables. Thus, this 'λ-calculus' is *actually* a *λ-calculus of contexts*, with stronger variables playing the contexts.

Records

Fix constants 1 and 2. l and m have level 1, X has level 2.

Here is a record:

$$X[l \mapsto 1][m \mapsto 2]$$

Here is record lookup:

$$\begin{aligned} X[l \mapsto 1][m \mapsto 2][X \mapsto m] &= X[l \mapsto 1][X \mapsto m][m \mapsto 2] \\ &= X[X \mapsto m][l \mapsto 1][m \mapsto 2] \\ &= m[l \mapsto 1][m \mapsto 2] \\ &= m[m \mapsto 2] \\ &= 2. \end{aligned}$$

In-place update

$$
\begin{aligned}
X[l\mapsto 1][m\mapsto 2][X\mapsto X[l\mapsto 2]] &= X[l\mapsto 1][X\mapsto X[l\mapsto 2]][m\mapsto 2] \\
&= X[X\mapsto X[l\mapsto 2]][l\mapsto 1][m\mapsto 2] \\
&= X[l\mapsto 2][l\mapsto 1][m\mapsto 2] \\
&= X[l\mapsto 2][m\mapsto 2]
\end{aligned}
$$

Substitution-as-a-term

$(\lambda X.X[l \mapsto \lambda n.n])$ applied to lm

$$(\lambda X.X[l \mapsto \lambda n.n])lm = X[l \mapsto \lambda n.n][X \mapsto lm] = (\lambda n.n)m$$

In-place update as a term

$$\lambda \mathcal{W}.\mathcal{W}[X \mapsto X[l \mapsto 2]] \quad \text{applied to} \quad X[l \mapsto 1][m \mapsto 2]$$

…and so on ($\mathcal{W}$ has level 3).

I'm *telling* you we can proceed to global state (the world is a big hole with state suspended on it, just like a record), and Abadi-Cardelli imp-ε object calculus.

Graphs are speculative; I haven't implemented them. I mentioned it only to give you some idea of the directions I am thinking in. However, this work may have applications to component-based systems, with strong variables controlling how components are 'plugged in'.

The usual higher-order version of the axiom of choice:

$$\forall x.\, \exists y.\, (\phi x y) \Leftrightarrow \exists f.\, \forall x.\, (\phi x (f x)).$$

May be true of individual ϕ, but the general assertion over all ϕ asserts that there always exists a function picking out *some* y for each x.

Our 'hierarchy'-based version (*I think*):

$$\forall x^1.\, \exists y^1.\, \phi \Leftrightarrow \exists Y^2.\, \forall x.\, (\phi[y \mapsto Y]).$$

Here ϕ is a sufficiently strong variable of sort o. For example (this works even without an axiom, because we have the terms to do it):

$$(\forall x.\, \exists y.\, x{=}y) = \top. \qquad (\exists Y.\, \forall x.\, x{=}Y) = \top$$

calculations omitted, but basically we substitute Y for x on the right-hand side. This gets captured by the $\forall$, which is for a weaker (later) variable.

Semantics

A model of a theory consists of the following data:

1. For each base data sort D, a set $[\![D]\!]^\bullet$. Extend this to all sorts as follows:

$$[\![S' \times S]\!]^\bullet = [\![S']\!]^\bullet \times [\![S]\!]^\bullet \qquad [\![[S']S]\!]^\bullet = [\![S]\!]^\bullet \cup \{*\}.$$

   $[\![S]\!]^\bullet \cup \{*\}$ adjoins a new element to $[\![S]\!]^\bullet$.

2. For $f : (S')S$ choose $[\![f]\!]^\bullet$ a function from $[\![S']\!]^\bullet$ to $[\![S]\!]^\bullet$.

Call $[\![\text{-}]\!]^\bullet$ the closed section.

Semantics

Define

$$\mathbb{T}^0_\sigma = [\![\sigma]\!]^\bullet \qquad \mathbb{T}^{i+1}_\sigma = (\mathbb{V}^{i+1} \stackrel{fin}{\longrightarrow} \mathbb{T}^i) \stackrel{fin}{\longrightarrow} \mathbb{T}^i_\sigma$$

Write $\mathbb{T}^i$ for $\bigcup_i \mathbb{T}^i_\sigma$, $\mathbb{T}_\sigma$ for $\bigcup_\sigma \mathbb{T}^i_\sigma$, and $\mathbb{T}$ for $\bigcup_{i,\sigma} \mathbb{T}^i_\sigma$.

I am afraid that all the hard work is hidden in the definition of $\stackrel{fin}{\longrightarrow}$, and in proving its (excellent) properties. That's what took me six months to work out. I shall conclude by sketching the construction. . .

Permutations

Write $\pi \in \mathbb{P}$ for level-preserving finitely supported bijections on variables. π is a bijection such that:

- $\pi(a)$ has the same level as a always (whence 'level-preserving').
- $\pi(a) = a$ for all variables, *except* for some finite set (whence 'finitely supported').

Write **Id** for the identity permutation mapping a to a always. Write composition of permutations $\pi \circ \pi'$. This is given by functional composition.

A $\mathbb{P}$-action on $\mathbb{S}$ is $\mathbb{P} \times \mathbb{S} \longrightarrow \mathbb{S}$, write it infix as $\pi \cdot s$, such that $\pi \cdot (\pi' \cdot s) = (\pi \circ \pi') \cdot s$ and $\mathbf{Id} \cdot s = s$.

Permutations

Say $s \in \mathbb{S}$ is supported by A a set of variables when if $\pi(a) = \pi'(a)$ for all $a \in A$ then $\pi \cdot s = \pi' \cdot s$ (say that A supports s). Say $\mathbb{S}$ has finite support when all its elements have a *finite* supporting set.

Lemma: *Suppose S has a finitely supported $\mathbb{P}$-action and $s \in S$. Then:*

1. *s has a unique smallest finite supporting set; call it the support of s and write it supp(s).*
2. *$a \in$ supp(s) if and only if for all but finitely many b, $(b\ a) \cdot s = s$.*
3. *$a \in$ supp(s) if and only if for any other $b \notin$ supp(s), $(c\ b) \cdot s = s$.*

Permutations

The typical example of a set with a $\mathbb{P}$-action is a set of syntax, where π acts literally on the variables mentioned in the syntax; then (finite) syntax is obviously supported by the finite set of variables mentioned in its syntax.

Functions $f \in \mathbb{S} \longrightarrow \mathbb{S}'$ have a $\mathbb{P}$-action given by $(\pi \cdot f)(\pi \cdot s) = \pi \cdot (f(s))$.

E.g. in $\mathbb{T}^i$ above $\kappa \in \mathbb{V}^i \longrightarrow \mathbb{S}$ has a natural permutation action given by $(\pi \cdot \kappa)(\pi \cdot a) = \pi \cdot \kappa(a)$.

Write $\mathbb{V}^i \overset{fin}{\longrightarrow} \mathbb{S}$ for the set of finitely supported functions from $\mathbb{V}^i$ to $\mathbb{S}$.

Permutations

$\tau \in (\mathbb{V}^i \stackrel{fin}{\longrightarrow} \mathbb{S}) \longrightarrow \mathbb{S}'$ also has a natural permutation action. Write $(\mathbb{V}^i \stackrel{fin}{\longrightarrow} \mathbb{S}) \stackrel{fin}{\longrightarrow} \mathbb{S}'$ for the set of τ such that:

1. τ has a finite supporting set.
2. There exists some finite $A \subseteq \mathbb{V}^i$ such that if κ and κ' agree on A then $\tau(\kappa) = \tau(\kappa')$ (we say that τ has no asymptotic behaviour).

The intuition is that τ only examines a finite part of κ; it must 'ignore what κ does to most variables'.

The fundamental result

If τ examines arguments κ at a then for all but finitely many b, $(a\,b) \cdot \tau \neq \tau$.

Semantics (sketched)

- $[\![a^i]\!]^i = \lambda\kappa^i.\kappa(a)$.
- $[\![a^i]\!]^j = \lambda\kappa^j.[\![a]\!]^{j-1}$ for $j > i$.
- $[\![a^i]\!]^j$ is not defined for $j < i$.
- Write $[\![f]\!]^0$ for $[\![f]\!]^\bullet$.
- Define $[\![f]\!]^i : [\![S_1]\!]^i \times \cdots \times [\![S_n]\!]^i \longrightarrow [\![S]\!]^i$ by

$$[\![f]\!]^i\kappa^i = \lambda\tau_1 \in [\![S_1]\!]^i \cdots \tau_n \in [\![S_n]\!]^i.[\![f]\!]^{i-1}(\tau_1\kappa, \ldots, \tau_n\kappa).$$

- Then $[\![f(t_1, \ldots, t_n)]\!]^i\kappa^i = [\![f]\!]^i([\![t_1]\!]^i\kappa, \ldots, [\![t_n]\!]^i\kappa)$ whenever $[\![t_1]\!]^i, \ldots, [\![t_n]\!]^i$ are all defined, and is undefined otherwise.

Semantics (sketched some more)

- $[\![[a]t]\!]^i = [a][\![t]\!]^i$ provided that a has level at most i, and $[\![t]\!]^i$ is defined.
- Write $\bot$ for $[\![\bot]\!]^i$, for any i. Write $\top$ for $[\![\bot \supset \bot]\!]^i$.
- $[\![s = t]\!]^i$ is undefined if $[\![s]\!]^i$ or $[\![t]\!]^i$ is undefined. If they *are* defined then $[\![s = t]\!]^i = \top$ if $[\![s]\!]^i = [\![t]\!]^i$ and $[\![s = t]\!]^i = \bot$ otherwise.
- If $[\![t]\!]^i$ is not defined or $j > i$ then $[\![a^j \# t]\!]^i$ is not defined. Otherwise: $[\![a\#t]\!] = \top$ if $a\#[\![t]\!]^i$, and $[\![a\#t]\!] = \bot$ if $a \not\# [\![t]\!]^i$.

Summary

I have given an equational system with the power to express some sophisticated concepts in terms of a few basic primities (abstraction and И, plus predicate logic, and of course the hierarchy of variables).

The real difference from higher-order frameworks (e.g. HOL) is that in HOL we say what *can* appear in a term (by applying that term to the argument). In this system (*what to call it?*), we say what *cannot* appear in a term (by asserting a freshness $\#$ or choosing the levels right).

Finally, I have given a semantics which has a simple intuition (tomorrow, this evening, this afternoon) but really quite subtle to get right — though I have not gone into details.