Nominal Algebra

Aad Mathijssen Murdoch J. Gabbay

Department of Mathematics and Computer Science
Technische Universiteit Eindhoven
The Netherlands

18th Nordic Workshop on Programming Theory (NWPT'06)
Reykjavík University, Iceland
18-20th October 2006

Motivation

In informal mathematical usage, we often encounter properties like the following:

- λ-calculus: $\lambda x.(tx) = t$ — if $x \notin fv(t)$.
- First-order logic: $(\forall x.\phi) \wedge \psi = \forall x.(\phi \wedge \psi)$ — if $x \notin fv(\psi)$.
- π-calculus: $(\nu x.P) \mid Q = \nu x.(P \mid Q)$ — if $x \notin fv(Q)$.

And for any binder $\xi \in \{\lambda, \forall, \nu\}$:

- $(\xi x.t)[y \mapsto u] = \xi x.(t[y \mapsto u])$ — if $x \notin fv(u)$.
- α-equivalence: $\xi x.t = \xi y.(t[x \mapsto y])$ — if $y \notin fv(t)$.

Motivation

In informal mathematical usage, we often encounter properties like the following:

- λ-calculus: $\lambda x.(tx) = t$ — if $x \notin fv(t)$.
- First-order logic: $(\forall x.\phi) \wedge \psi = \forall x.(\phi \wedge \psi)$ — if $x \notin fv(\psi)$.
- π-calculus: $(\nu x.P) \mid Q = \nu x.(P \mid Q)$ — if $x \notin fv(Q)$.

And for any binder $\xi \in \{\lambda, \forall, \nu\}$:

- $(\xi x.t)[y \mapsto u] = \xi x.(t[y \mapsto u])$ — if $x \notin fv(u)$.
- α-equivalence: $\xi x.t = \xi y.(t[x \mapsto y])$ — if $y \notin fv(t)$.

Here:

- t, u, ϕ, ψ, P, Q are meta-variables ranging over terms.

Motivation

In informal mathematical usage, we often encounter properties like the following:

- λ-calculus: $\lambda x.(tx) = t$ — if $x \notin fv(t)$.
- First-order logic: $(\forall x.\phi) \wedge \psi = \forall x.(\phi \wedge \psi)$ — if $x \notin fv(\psi)$.
- π-calculus: $(\nu x.P) \mid Q = \nu x.(P \mid Q)$ — if $x \notin fv(Q)$.

And for any binder $\xi \in \{\lambda, \forall, \nu\}$:

- $(\xi x.t)[y \mapsto u] = \xi x.(t[y \mapsto u])$ — if $x \notin fv(u)$.
- α-equivalence: $\xi x.t = \xi y.(t[x \mapsto y])$ — if $y \notin fv(t)$.

Here:

- t, u, ϕ, ψ, P, Q are meta-variables ranging over terms.
- Freshness occurs in the presence of meta-variables.

Motivation (2)

Question: Is it possible to formalise these meta-level properties in a direct way?

Motivation (2)

Question: Is it possible to formalise these meta-level properties in a direct way?

Answer: Yes, using a universal algebra on nominal terms.

Motivation (2)

Question: Is it possible to formalise these meta-level properties in a direct way?

Answer: Yes, using a universal algebra on nominal terms.

Explanation:

- Universal algebra, or equational logic, is one of the simplest languages to study properties of mathematical structures.
- Nominal terms are a syntax designed to naturally express binding and freshness in the presence of meta-variables.

Nominal Terms
Definition

Nominal terms are inductively defined by:

$$t \quad ::= \quad a \mid X \mid \mathsf{f}(t_1, \ldots, t_n) \mid [a]t$$

Here we fix:

- atoms $a, b, c, \ldots$ (for x, y).
- unknowns $X, Y, Z, \ldots$ (for t, u, ϕ, ψ, P and Q).
- term-formers $\mathsf{f}, \mathsf{g}, \mathsf{h}, \ldots$ (for λ, $\_\,\_$, $\forall$, $\wedge$, ν, $|$, $\_[\_ \mapsto \_]$).

We call $[a]t$ an abstraction (for the $x.\_$).

Nominal Terms

Examples

Representation of mathematical syntax in nominal terms:

mathematics	nominal terms	
	unsugared	sugared
$\lambda x.t$	$\lambda([a]X)$	$\lambda[a]X$
$\lambda x.(tx)$	$\lambda([a]\mathrm{app}(X, a))$	$\lambda[a](Xa)$
$(\forall x.\phi) \wedge \psi$	$\wedge(\forall([a]X), Y)$	$(\forall[a]X) \wedge Y$
$(\nu x.P) \mid Q$	$\mid(\nu([a]X), Y)$	$(\nu[a]X) \mid Y$
$t[x \mapsto u]$	$\mathrm{sub}([a]X, Y)$	$X[a \mapsto Y]$

Nominal algebra
Definition

Nominal algebra is a theory of equality between nominal terms:

- $t = u$ is an equality.
- $a\#X$ is a primitive freshness (for $x \notin fv(t)$).
- A freshness context Δ is a *finite set* of primitive freshnesses.
- $\Delta \to t = u$ is a judgement (for '$t = u$ if $x \notin fv(v)$').
  If $\Delta = \emptyset$, write $t = u$.

Nominal algebra
Example judgements

Meta-level properties as judgements in nominal algebra:

- λ-calculus: $a\#X \to \lambda[a](Xa) = X$.
- First-order logic: $a\#Y \to (\forall[a]X) \wedge Y = \forall[a](X \wedge Y)$.
- π-calculus: $a\#Y \to (\nu[a]X) \mid Y = \nu[a](X \mid Y)$.

And for any binder $\xi \in \{\forall, \lambda, \nu\}$:

- $a\#Y \to (\xi[a]X)[b \mapsto Y] = \xi[a](X[b \mapsto Y])$.
- α-equivalence: $b\#X \to \xi[a]X = \xi[b](X[a \mapsto b])$.

Nominal algebra

Theories

A theory in nominal algebra consists of:

- a set of term-formers;
- a set of axioms: judgements $\Delta \longrightarrow t = u$.

Nominal Algebra
LAM: the lambda-calculus

A theory LAM for the lambda-calculus with meta-variables:

- Term-formers λ, app and sub
  (recall that $t[a \mapsto u]$ is just sugar for $\mathrm{sub}([a]t, u)$).
- An axiom for β-reduction:

$$(\beta) \quad (\lambda[a]Y)X = Y[a \mapsto X]$$

Example judgements in LAM:

$$(\lambda[a]Y)X = Y[a \mapsto X] \qquad (\lambda[a]b)c = b[a \mapsto c]$$

$$(\lambda[a]a)c = a[a \mapsto c] \qquad (\lambda[b]a)c = a[b \mapsto c]$$

$$(\lambda[a](\lambda[b]Z)Y)X = ((\lambda[b]Z)Y)[a \mapsto X] = Z[b \mapsto Y][a \mapsto X]$$

Nominal Algebra
FOL: first-order logic

A theory FOL for first-order logic with meta-variables, also called *one-and-a-halfth-order logic*:

- Term-formers:
  - $\bot$, $\supset$, $\forall$, $\approx$ and sub for the basic operators ($\top$, $\neg$, $\wedge$, $\vee$, $\Leftrightarrow$, $\exists$ are sugar);
  - $p_1, \ldots, p_m$ and $f_1, \ldots, f_n$ for object-level predicates and terms.
- Axioms: ...

Nominal Algebra
Axioms of FOL

$$
\begin{array}{rl}
\textbf{(MP)} & \top \supset P \;=\; P \\
\textbf{(SwapL)} & P \supset (Q \supset R) \;=\; Q \supset (P \supset R) \\
\textbf{(CP)} & \neg P \supset Q \;=\; \neg Q \supset P \\
\textbf{(BotE)} & \bot \supset P \;=\; \top \\
\textbf{(OrIdem)} & \neg P \supset P \;=\; P \\
\textbf{(Triv)} & P \supset P \;=\; \top \\
\\
\textbf{(Q1)} & \forall[a]P \supset P[a \mapsto T] \;=\; \top \\
\textbf{(Q2)} & \forall[a](P \wedge Q) \;=\; \forall[a]P \wedge \forall[a]Q \\
\textbf{(Q3)} & a\#P \;\rightarrow\; \forall[a](P \supset Q) \;=\; P \supset \forall[a]Q \\
\\
\textbf{(E1)} & T \approx T \;=\; \top \\
\textbf{(E2)} & U \approx T \wedge P[a \mapsto T] \supset P[a \mapsto U] \;=\; \top
\end{array}
$$

Nominal Algebra

Axiems of FOL: (Q3)

$$(\mathbf{Q3}) \quad a\#P \;\to\; \forall[a](P \supset Q) \;=\; P \supset \forall[a]Q$$

Inst. P	Resulting judgement
$P := p(a)$	violation of freshness context
$P := p(b)$	$\forall[a](p(b) \supset Q) = p(b) \supset \forall[a]Q$
$P := \forall[a]R$	$\forall[a](\forall[a]R \supset Q) = \forall[a]R \supset \forall[a]Q$
$P := \forall[b]R$	$a\#R \to \forall[a](\forall[b]R \supset Q) = \forall[b]R \supset \forall[a]Q$
$P := R \supset S$	$a\#R, a\#S \to$ $\forall[a]((R \supset S) \supset Q) = (R \supset S) \supset \forall[a]Q$

Nominal Algebra

SUB: a theory of explicit substitution

A theory SUB for explicit substitution is:

$$
\begin{array}{rrcl}
(\mathbf{var}{\mapsto}) & a[a \mapsto T] & = & T \\
(\#{\mapsto}) & a\#X \rightarrow X[a \mapsto T] & = & X \\
(\mathbf{f}{\mapsto}) & \mathsf{f}(X_1, \ldots, X_n)[a \mapsto T] & = & \mathsf{f}(X_1[a \mapsto T], \ldots, X_n[a \mapsto T]) \\
(\mathbf{abs}{\mapsto}) & b\#T \rightarrow ([b]X)[a \mapsto T] & = & [b](X[a \mapsto T]) \\
(\mathbf{ren}{\mapsto}) & b\#X \rightarrow X[a \mapsto b] & = & (b\ a) \cdot X
\end{array}
$$

Nominal algebra
Results

Results on nominal algebra:

- it has a semantics in *nominal sets*;
- it has a notion of derivability:
  - sound and complete with respect to the semantics;
  - fresh atoms can be introduced within a derivation.
- α-equivalence of terms with meta-variables:
  - permutations of atoms are stuck on unknowns;
  - unification up to α-equivalence is decidable.

Nominal algebra

Results on the theories (other work)

Results on theory SUB:

- actual capture-avoiding substitution on closed terms;
- extending to open terms: omega-completeness.

Results on theory FOL:

- first-order logic on closed terms;
- has an equivalent sequent calculus:
  - representing schemas of derivations in first-order logic;
  - satisfies cut-elimination.

Conclusions

Nominal algebra:

- is a theory of algebraic equality on nominal terms;
- allows us to reason about systems with binding;
- closely mirrors informal mathematical usage:
  - we can manipulate variables directly
  - natural notion of instantiation of meta-variables:

    informal notation: instantiating t to x in $\lambda x.t$ yields $\lambda x.x$.
    nominal terms: instantiating X to a in $\lambda[a]X$ yields $\lambda[a]a$.

Nominal terms revisited
Permutations

Nominal terms are inductively defined by:

$$t \quad ::= \quad a \mid \pi \cdot X \mid \mathsf{f}(t_1, \ldots, t_n) \mid [a]t$$

Here:

- π a permutation of atoms.
- we call $\pi \cdot X$ a moderated unknown;
  write X when π is the trivial permutation **Id**.

Nominal algebra revisited

α-equivalence

Permutations essentially capture α-equivalence on nominal terms:

$$a\#X \rightarrow [a]X = [b](b\ a) \cdot X$$

For any binder $\xi \in \{\forall, \lambda, \nu\}$:

$$a\#X \rightarrow \xi[a]X = \xi[b](b\ a) \cdot X$$

Sorts

Nominal algebra is sorted.

Sorts τ, inductively defined by:

$$\tau \quad ::= \quad \mathbb{A} \mid \delta \mid [\mathbb{A}]\tau$$

Here:

- a set $\mathbb{A}$ is the set of all atoms $a, b, c, \ldots$;
- we fix base sorts δ;
- $[\mathbb{A}]\tau$ represents an abstraction set:
  the set consisting of elements of τ with an atom abstracted.

Sorting assertions

Assign to each

- unknown X a sort τ, write this as $X : \tau$;
- term-former f an arity $(\tau_1, \ldots, \tau_n)\tau$,
  write this as $\mathsf{f} : (\tau_1, \ldots, \tau_n)\tau$.

Define sorting assertions on nominal terms, inductively by:

$$\frac{}{a : \mathbb{A}} \qquad \frac{}{\pi \cdot X_\tau : \tau} \qquad \frac{t : \tau}{[a]t : [\mathbb{A}]\tau}$$

$$\frac{\mathsf{f} : (\tau_1, \ldots, \tau_n)\tau \qquad t_1 : \tau_1 \quad \cdots \quad t_n : \tau_n}{\mathsf{f}(t_1, \ldots, t_n) : \tau}$$

In equalities $t = u$, t and u should have the same sort.

Freshness on terms

Definition and derivability

Recall that a *primitive* freshness is a pair $a\#X$.
A freshness $a\#t$ is a pair of an atom a and a term t.

Write $\Delta \vdash a\#t$ when $a\#t$ is derivable from Δ using the following inference rules:

$$\frac{}{a\#b}\,(\#\mathbf{ab}) \qquad \frac{\pi^{-1}(a)\#X}{a\#\pi\cdot X}\,(\#\mathbf{X})$$

$$\frac{}{a\#[a]t}\,(\#[]\mathbf{a}) \qquad \frac{a\#t}{a\#[b]t}\,(\#[]\mathbf{b}) \qquad \frac{a\#t_1 \quad \cdots \quad a\#t_n}{a\#\mathsf{f}(t_1,\ldots,t_n)}\,(\#\mathbf{f})$$

Examples:

$$\vdash a\#b \qquad \vdash a\#\lambda[a]X \qquad a\#X \vdash a\#\lambda[b]X$$

Derivability of equalities

Write $\Delta \vdash_{\mathsf{T}} t = u$ when $t = u$ is derivable from the rules below, s.t.

- only assumptions used are from Δ;
- each axiom used in $(\mathbf{ax}_{\Delta' \to t' = u'})$ is from T only.

$$\frac{}{t = t}\,(\mathbf{refl}) \qquad \frac{t = u}{u = t}\,(\mathbf{symm}) \qquad \frac{t = u \quad u = v}{t = v}\,(\mathbf{tran})$$

$$\frac{t = u}{C[t] = C[u]}\,(\mathbf{cong}) \qquad \frac{a\#t \quad b\#t}{(a\ b)\cdot t = t}\,(\mathbf{perm})$$

$$\frac{\Delta^{\pi}\sigma}{t^{\pi}\sigma = u^{\pi}\sigma}\,(\mathbf{ax}_{\Delta \,\to\, t = u}) \qquad \begin{array}{c} [a\#X_1, \ldots, a\#X_n] \qquad \Delta \\ \vdots \\ \dfrac{t = u}{t = u}\,(\mathbf{fr}) \quad (a \notin t, u, \Delta) \end{array}$$

Related work

Related work to Nominal Algebra (NA):

- Higher-Order Algebra (HOA)
- Cylindric Algebra and Lambda-Abstraction Algebra (CA/LAA)

These do not mirror informal mathematical usage like NA does:

- Non-capturing substitution cannot be defined HOA/CA/LAA. It is the default notion of (meta-level) substitution in NA.
- Variables are encoded:
  - by higher-order functions in HOA;
  - by De Bruijn indices in CA/LAA.