# A Formal Calculus for Informal Equality with Binding

Aad Mathijssen

Department of Mathematics and Computer Science
Technische Universiteit Eindhoven

Joint work with Murdoch J. Gabbay

Mathematical Theories of Abstraction, Substitution and Naming in Computer Science
International Centre for Mathematical Sciences (ICMS), Edinburgh

27th May 2007

# Motivation

## The $\lambda$-calculus

The $\lambda$-calculus:

$$t \quad ::= \quad x \mid tt \mid \lambda x.t$$

Axioms:

$$\begin{array}{lll} (\alpha) & \lambda x.t = \lambda y.(t[x \mapsto y]) & \text{if } y \notin fv(t) \\ (\beta) & (\lambda x.t)u = t[x \mapsto u] & \\ (\eta) & \lambda x.(tx) = t & \text{if } x \notin fv(t) \end{array}$$

Free variables function $fv$:

$$fv(x) = \{x\} \qquad fv(tu) = fv(t) \cup fv(u) \qquad fv(\lambda x.t) = fv(t) \backslash \{x\}$$

# Motivation

## The $\lambda$-calculus

The $\lambda$-calculus:

$$t \quad ::= \quad x \mid tt \mid \lambda x.t$$

Axiom schemata:

$$\begin{array}{llll} (\alpha) & \lambda x.t & = \lambda y.(t[x \mapsto y]) & \text{if } y \notin fv(t) \\ (\beta) & (\lambda x.t)u & = t[x \mapsto u] & \\ (\eta) & \lambda x.(tx) & = t & \text{if } x \notin fv(t) \end{array}$$

Free variables function $fv$:

$$fv(x) = \{x\} \qquad fv(tu) = fv(t) \cup fv(u) \qquad fv(\lambda x.t) = fv(t) \backslash \{x\}$$

$t$ and $u$ are meta-variables ranging over terms.

# Motivation

## The $\lambda$-calculus

The $\lambda$-calculus with meta-variables:

$$t \quad ::= \quad x \mid tt \mid \lambda x.t \mid X$$

Axioms:

$$\begin{array}{lllll}
(\alpha) & \lambda x.X & = \lambda y.(X[x \mapsto y]) & \text{if } y \notin fv(X) \\
(\beta) & (\lambda x.X)Y & = X[x \mapsto Y] & \\
(\eta) & \lambda x.(Xx) & = X & \text{if } x \notin fv(X)
\end{array}$$

Free variables function $fv$:

$$fv(x) = \{x\} \qquad fv(tu) = fv(t) \cup fv(u) \qquad fv(\lambda x.t) = fv(t) \backslash \{x\}$$

# Motivation

## The $\lambda$-calculus

The $\lambda$-calculus with meta-variables:

$$t \quad ::= \quad x \mid tt \mid \lambda x.t \mid X$$

Axioms:

$$\begin{array}{lll} (\alpha) & \lambda x.X = \lambda y.(X[x \mapsto y]) & \text{if } y \notin fv(X) \\ (\beta) & (\lambda x.X)Y = X[x \mapsto Y] & \\ (\eta) & \lambda x.(Xx) = X & \text{if } x \notin fv(X) \end{array}$$

Free variables function $fv$:

$$fv(x) = \{x\} \qquad fv(tu) = fv(t) \cup fv(u) \qquad fv(\lambda x.t) = fv(t) \backslash \{x\}$$

Freshness occurs in the presence of meta-variables:
We only know if $x \notin fv(X)$ when $X$ is instantiated.

# Motivation

## Other examples

In informal mathematical usage, we see equalities like:

- First-order logic: $(\forall x.\phi) \wedge \psi \quad = \forall x.(\phi \wedge \psi) \quad$ if $x \notin fv(\psi)$
- $\pi$-calculus: $(\nu x.P) \mid Q \quad = \nu x.(P \mid Q) \quad$ if $x \notin fv(Q)$
- $\mu$CRL/mCRL2: $\sum_x .p \quad = p \quad$ if $x \notin fv(p)$

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

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

# Motivation

## Other examples

In informal mathematical usage, we see equalities like:

- First-order logic: $(\forall x.\phi) \wedge \psi \quad = \forall x.(\phi \wedge \psi) \quad$ if $x \notin fv(\psi)$
- $\pi$-calculus: $(\nu x.P) \mid Q \quad = \nu x.(P \mid Q) \quad$ if $x \notin fv(Q)$
- $\mu$CRL/mCRL2: $\sum_x .p \quad = p \quad$ if $x \notin fv(p)$

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

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

Here:

- $\phi, \psi, P, Q, p, t, u$ are meta-variables ranging over terms.

# Motivation

## Other examples

In informal mathematical usage, we see equalities like:

- First-order logic: $(\forall x.\phi) \wedge \psi \quad = \forall x.(\phi \wedge \psi) \quad$ if $x \notin fv(\psi)$
- $\pi$-calculus: $(\nu x.P) \mid Q \quad = \nu x.(P \mid Q) \quad$ if $x \notin fv(Q)$
- $\mu$CRL/mCRL2: $\sum_x .p \quad = p \quad$ if $x \notin fv(p)$

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

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

Here:

- $\phi, \psi, P, Q, p, t, u$ are meta-variables ranging over terms.
- Freshness occurs in the presence of meta-variables.

# Motivation

## Formalisation

Question: Can we formalise binding and freshness in the presence of meta-variables?

# Motivation

## Formalisation

Question: Can we formalise binding and freshness in the presence of meta-variables?

Answer: Yes, using Nominal Terms (Urban, Gabbay, Pitts)

# Motivation

## Formalisation

Question: Can we formalise binding and freshness in the presence of meta-variables?

Answer: Yes, using Nominal Terms (Urban, Gabbay, Pitts)

Question: Can we formalise equality with binding in the presence of meta-variables?

# Motivation

## Formalisation

Question: Can we formalise binding and freshness in the presence of meta-variables?

Answer: Yes, using Nominal Terms (Urban, Gabbay, Pitts)

Question: Can we formalise equality with binding in the presence of meta-variables?

Answer: Yes, using Nominal Algebra...

# Overview

Overview:

- Nominal terms
- Nominal algebra:
  - Definitions
  - Examples
- $\alpha$-conversion
- Derivability of equality
- A semantics in nominal sets
- Related work
- Conclusions and future work

# Nominal Terms

## Definition

Nominal terms are inductively defined by:

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

Here we fix:

- atoms $a, b, c, \ldots$ (for $x, y$)
- unknowns $X, Y, Z, \ldots$ (for $t$, $u$, $\phi$, $\psi$, $P$, $Q$, $p$)
- term-formers $\mathsf{f}, \mathsf{g}, \mathsf{h}, \ldots$ (for $\lambda$, $\_\,\_$, $\forall$, $\wedge$, $\nu$, $|$, $\sum$, $\_[\_ \mapsto \_]$)

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

# Nominal Terms

## Definition

Nominal terms are inductively defined by:

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

Here we fix:

- atoms $a, b, c, \ldots$ (for $x, y$)
- unknowns $X, Y, Z, \ldots$ (for $t$, $u$, $\phi$, $\psi$, $P$, $Q$, $p$)
- term-formers $\mathsf{f}, \mathsf{g}, \mathsf{h}, \ldots$ (for $\lambda$, $\_\,\_$, $\forall$, $\wedge$, $\nu$, $|$, $\sum$, $\_[\_ \mapsto \_]$)

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

We can impose a sorting system on nominal terms ...
but we don't do that here.

# Nominal Terms

## Examples

Representation of mathematical syntax in nominal terms:

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

# Nominal Terms

## Freshness

Definition:

- Call $a\#X$ a primitive freshness (for '$x \notin fv(t)$').
- A freshness context $\Delta$ is a *finite set* of primitive freshnesses.

# Nominal Terms

## Freshness

Definition:

- Call $a\#X$ a primitive freshness (for '$x \notin fv(t)$').
- A freshness context $\Delta$ is a *finite set* of primitive freshnesses.

Generalise freshness on unknowns $X$ to terms $t$:

- Call $a\#t$ a freshness, where $t$ is a nominal term.
- Write $\Delta \vdash a\#t$ when $a\#t$ is derivable from $\Delta$ using

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

Examples:

| | | |
|---|---|---|
| $\vdash a\#b$ | $\vdash a\#\lambda[a]X$ | $a\#X \vdash a\#\lambda[b]X$ |
| $\nvdash a\#a$ | $\nvdash a\#\lambda[b]X$ | $a\#X \nvdash a\#Y$ |

# Nominal Algebra

## Definition

Nominal algebra is a theory of equality between nominal terms:

- $t = u$ is an equality where $t$ and $u$ are nominal terms.
- $\Delta \vdash t = u$ is an equality-in-context
  (for '$t' = u'$ if $x \notin fv(v')$').

# Nominal Algebra

## Example equalities-in-context

Meta-level properties as equalities-in-context in nominal algebra:

- $\lambda$-calculus: $a\#X \vdash \lambda[a](Xa) = X$
- First-order logic: $a\#Y \vdash (\forall[a]X) \wedge Y = \forall[a](X \wedge Y)$
- $\pi$-calculus: $a\#Y \vdash (\nu[a]X) \mid Y = \nu[a](X \mid Y)$
- $\mu$CRL/mCRL2: $a\#X \vdash \sum[a]X = X$

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

- $a\#Y \vdash (\xi[a]X)[b \mapsto Y] = \xi[a](X[b \mapsto Y])$
- $\alpha$-equivalence: $b\#X \vdash \xi[a]X = \xi[b](X[a \mapsto b])$

# Nominal algebra

## Theories

A theory in nominal algebra consists of:

- a set of term-formers
- a set of axioms: equalities-in-context $\Delta \vdash t = u$

# Nominal Algebra

## LAM: the $\lambda$-calculus

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

- term-formers $\lambda$, app and sub
  (recall that $t[a \mapsto u]$ is just sugar for $\mathsf{sub}([a]t, u)$)
- axioms:

$$
\begin{array}{llcllcl}
(\alpha) & b\#X & \vdash & \lambda[a]X & = & \lambda[b](X[a \mapsto b]) \\
(\beta) & & \vdash & (\lambda[a]Y)X & = & Y[a \mapsto X] \\
(\eta) & a\#X & \vdash & \lambda[a](Xa) & = & X
\end{array}
$$

# 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

Axioms of one-and-a-halfth-order logic:

$$
\begin{array}{rl}
\textbf{(MP)} & \vdash \top \supset P \;=\; P \\
\textbf{(M)} & \vdash ((((P \supset Q) \supset (\neg R \supset \neg S)) \supset R) \supset T) \\
& \qquad\qquad \supset ((T \supset P) \supset (S \supset P)) \quad = \top \\
\textbf{(Q1)} & \vdash \forall[a]P \supset P[a \mapsto T] \;=\; \top \\
\textbf{(Q2)} & \vdash \forall[a](P \wedge Q) \;=\; \forall[a]P \wedge \forall[a]Q \\
\textbf{(Q3)} & a\#P \;\vdash\; \forall[a](P \supset Q) \;=\; P \supset \forall[a]Q \\
\textbf{(E1)} & \vdash T \approx T \;=\; \top \\
\textbf{(E2)} & \vdash U \approx T \wedge P[a \mapsto T] \supset P[a \mapsto U] \;=\; \top
\end{array}
$$

# Nominal Algebra

## SUB: a theory of capture-avoiding substitution

A theory SUB for capture-avoiding substitution with meta-variables:

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

# $\alpha$-conversion

## Problem

Formalising binding implies formalising $\alpha$-conversion.

Idea: use theory SUB:

$$b\#X \vdash [a]X = [b](X[a \mapsto b])$$

# $\alpha$-conversion

## Problem

Formalising binding implies formalising $\alpha$-conversion.

Idea: use theory SUB:

$$b\#X \vdash [a]X = [b](X[a \mapsto b])$$

This destroys the proof theory:

- When proving properties by induction on the size of terms, you often want to freshen up a term using $\alpha$-conversion.
- Freshening using the above $\alpha$-conversion increases term size.

# $\alpha$-conversion

## Problem

Formalising binding implies formalising $\alpha$-conversion.

Idea: use theory SUB:

$$b\#X \vdash [a]X = [b](X[a \mapsto b])$$

This destroys the proof theory:

- When proving properties by induction on the size of terms, you often want to freshen up a term using $\alpha$-conversion.
- Freshening using the above $\alpha$-conversion increases term size.

Not all systems need substitution of terms for atoms, e.g. the $\pi$-calculus.

# $\alpha$-conversion

## Solution

Solution: use permutations of atoms:

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

# $\alpha$-conversion

## Solution

Solution: use permutations of atoms:

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

Redefine nominal terms:

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

Here:

- we call $\pi \cdot X$ a moderated unknown
- write $X$ when $\pi$ is the trivial permutation **Id**
- instantiation of $X$ to $t$ in $\pi \cdot X$ gives us $\pi \cdot t$:

$$\pi \cdot a \equiv \pi(a) \qquad \pi \cdot (\pi' \cdot X) \equiv (\pi \circ \pi') \cdot X \qquad \pi \cdot [a]t \equiv [\pi(a)](\pi \cdot t)$$

$$\pi \cdot \mathsf{f}(t_1, \ldots, t_n) \equiv \mathsf{f}(\pi \cdot t_1, \ldots, \pi \cdot t_n)$$

# $\alpha$-conversion

## Consequence

Add freshness derivation rule:

$$\frac{\pi^{-1}(a)\#X}{a\#\pi\cdot X}\ (\#\mathbf{X})\quad(\pi\neq\mathbf{Id})$$

Redefine theory SUB for capture-avoiding substitution:

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

# Derivability of equalities

## Definition

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

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

$$\frac{}{t = t}\,(\mathbf{refl}) \qquad \frac{t = u}{u = t}\,(\mathbf{symm}) \qquad \frac{t = u \quad u = v}{t = v}\,(\mathbf{tran}) \qquad \frac{a\#t \quad b\#t}{(a\ b)\cdot t = t}\,(\mathbf{perm})$$

$$\frac{t = u}{[a]t = [a]u}\,(\mathbf{cong}[]) \qquad \frac{t = u}{\mathsf{f}(t_1, \ldots, t, \ldots, t_n) = \mathsf{f}(t_1, \ldots, u, \ldots, t_n)}\,(\mathbf{cong}\mathsf{f})$$

$$\frac{\pi \cdot \Delta'\sigma}{\pi \cdot t'\sigma = \pi \cdot u'\sigma}\,(\mathbf{ax}_{\Delta' \vdash t' = u'}) \qquad \frac{\begin{array}{c}[a\#X_1, \ldots, a\#X_n] \qquad \Delta \\ \vdots \\ t = u\end{array}}{t = u}\,(\mathbf{fr}) \quad (a \notin t, u, \Delta)$$

# Derivability of equalities

## Instantiation of $(\beta)$ in LAM

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

Instantiation of the $(\beta)$ axiom:

| $\sigma$ | $\pi$ | Result |
|---|---|---|
| $[]$ | **Id** | $\vdash (\lambda[a]Y)X = Y[a \mapsto X]$ |
| $[b/Y, c/X]$ | **Id** | $\vdash (\lambda[a]b)c = b[a \mapsto c]$ |
| $[a/Y, c/X]$ | **Id** | $\vdash (\lambda[a]a)c = a[a \mapsto c]$ |
| $[a/Y, c/X]$ | $(a\ b)$ | $\vdash (\lambda[b]b)c = b[b \mapsto c]$ |
| $[(\lambda[b]Z)Y/Y]$ | **Id** | $\vdash (\lambda[a](\lambda[b]Z)Y)X = ((\lambda[b]Z)Y)[a \mapsto X]$ |

# Derivability of equalities

## Instantiation of $(\eta)$ in LAM

$$(\eta) \quad a\#X \vdash \lambda[a](Xa) = X$$

Instantiation of the $(\eta)$ axiom:

| $\sigma$ | $\pi$ | Resulting equality-in-context |
|---|---|---|
| $[a/X]$ | **Id** | none, since $\nvdash a\#a$ |
| $[b/X]$ | **Id** | $\vdash \lambda[a](ba) = b$ |
| $[YZ/X]$ | **Id** | $a\#Y, a\#Z \vdash \lambda[a]((YZ)a) = YZ$ |
| $[\lambda[a]Y/X]$ | **Id** | $\vdash \lambda[a]((\lambda[a]Y)a) = \lambda[a]Y$ |
| $[\lambda[b]Y/X]$ | **Id** | $a\#Y \vdash \lambda[a]((\lambda[b]Y)a) = \lambda[b]Y$ |

# Derivability of equalities

## An example derivation

A derivation of $\vdash_{\mathsf{SUB}} X[a \mapsto a] = X$:

$$
\dfrac{
\dfrac{
\dfrac{
\dfrac{
\dfrac{}{a\#[a]X}\,(\#[]\mathbf{a})
\qquad
\dfrac{[b\#X]^1}{b\#[a]X}\,(\#[]\mathbf{b})
}{[b](b\ a)\cdot X = [a]X}\,(\mathbf{perm})
}{[a]X = [b](b\ a)\cdot X}\,(\mathbf{symm})
}{X[a \mapsto a] = ((b\ a)\cdot X)[b \mapsto a]}\,(\mathbf{congf})
\qquad
\dfrac{
\dfrac{[b\#X]^1}{a\#(b\ a)\cdot X}\,(\#\mathbf{X})
}{((b\ a)\cdot X)[b \mapsto a] = X}\,(\mathbf{ax}_{\mathbf{ren}\mapsto})
}{
\dfrac{X[a \mapsto a] = X}{X[a \mapsto a] = X}\,(\mathbf{fr})^1
}\,(\mathbf{tran})
$$

# Derivability of equalities

## Results for specific theories

Results on the CORE theory with no axioms:

- Syntactic criteria for deciding equality between terms
- Equivalent to $\alpha$-equality in Nominal Unification and Rewriting

Results on theory SUB:

- It is decidable whether $\Delta \vdash_{\mathsf{SUB}} t = u$
- Omega-complete: sound and complete w.r.t. the term model

Results on theory FOL:

- has an equivalent sequent calculus:
  - representing schemas of derivations in first-order logic
  - satisfies cut-elimination
- equivalent to first-order logic for terms without unknowns

# A semantics in nominal sets

## Definitions

Nominal algebra theories have a semantics in nominal sets:

- An interpretation $[\![\_]\!]_\varsigma$ of terms under a valuation $\varsigma$:

$$[\![a]\!]_\varsigma = a \qquad [\![\pi{\cdot}X]\!]_\varsigma = \pi{\cdot}\varsigma(X) \qquad [\![[a]t]\!]_\varsigma = [a][\![t]\!]_\varsigma$$

$$[\![\mathsf{f}(t_1,\ldots,t_n)]\!]_\varsigma = [\![\mathsf{f}]\!]([\![t_1]\!]_\varsigma,\ldots,[\![t_n]\!]_\varsigma)$$

- Validity of freshness and equality:

$$[\![\Delta]\!]_\varsigma \text{ when } a\#\varsigma(X) \text{ for each } a\#X \in \Delta$$

$$[\![\Delta \vdash a\#t]\!] \text{ when } [\![\Delta]\!]_\varsigma \text{ implies } a\#[\![t]\!]_\varsigma \text{ for all } \varsigma$$

$$[\![\Delta \vdash t = u]\!] \text{ when } [\![\Delta]\!]_\varsigma \text{ implies } [\![t]\!]_\varsigma = [\![u]\!]_\varsigma \text{ for all } \varsigma$$

- A model of a theory $\mathsf{T}$ is an interpretation $[\![\_]\!]$ such that $[\![\Delta \vdash t = u]\!]$ for all axioms $\Delta \vdash t = u$ of $\mathsf{T}$.
- Write $\Delta \models_{\mathsf{T}} a\#t$ when $[\![\Delta \vdash a\#t]\!]$ for all models $[\![\_]\!]$ of $\mathsf{T}$.
  Write $\Delta \models_{\mathsf{T}} t = u$ when $[\![\Delta \vdash t = u]\!]$ for all models $[\![\_]\!]$ of $\mathsf{T}$.

# A semantics in nominal sets

## Soundness and completeness

Derivability of equality is sound and complete:

$$\Delta \vdash_{\mathsf{T}} t = u \quad \text{if and only if} \quad \Delta \models_{\mathsf{T}} t = u.$$

Derivability of freshness is sound:

$$\text{If} \quad \Delta \vdash a\#t \quad \text{then} \quad \Delta \models_{\mathsf{T}} a\#t.$$

... but not complete, e.g.:

$$\models_{\mathsf{LAM}} a\#(\lambda[a]b)a \quad \text{but not} \quad \vdash a\#(\lambda[a]b)a.$$

This is no loss in power:

$$\Delta \models_{\mathsf{T}} a\#t \quad \text{if and only if} \quad \Delta, b\#X_1, \ldots, b\#X_n \vdash_{\mathsf{T}} (b\ a) \cdot t = t,$$

where $b$ is fresh and the $X_i$ are all unknowns mentioned in $t, \Delta$.

# Related work

## Nominal Equational Logic

Closely related to Nominal Algebra:

- Nominal Equational Logic (NEL) by Pitts and Clouston

Derivability of freshness is semantic and not syntactic:

- In NEL freshness derivability is complete
- Potentially undecidable
- Expressing syntactic freshness is impossible:

  $x \notin fv(t)$ does not correspond to $\vdash a \# t'$

# Related work

## Non-nominal approaches

Other related work:

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

These do not mirror informal equality like NA does:

- Binding and freshness are encoded:
  - by higher-order functions in HOA
  - by replacing $t$ by $c_i t$ to ensure $x_i \notin fv(t)$ in CA/LAA
- Reasoning about binding becomes different.
- Non-capturing substitution cannot be defined HOA/CA/LAA. It is the default notion of (meta-level) substitution in NA.

# 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:
  - existing axioma schemata can be expressed directly
  - equational proofs carry over 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 $\lambda[a]a$

# Future work

Future work on nominal algebra:

- further develop theory on:
  - the $\lambda$-calculus
  - choice quantification in $\mu$CRL/mCRL2
  - $\pi$-calculus and its variants
  - reversibility
- investigate other kinds of semantics
- formalise meta-level reasoning, meta-meta-level reasoning,... a hierarchy of variables.
- develop a theorem prover

# Further reading


- Murdoch J. Gabbay, Aad Mathijssen:
  A Formal Calculus for Informal Equality with Binding.
  WoLLIC'07.
- Murdoch J. Gabbay, Aad Mathijssen:
  Capture-Avoiding Substitution as a Nominal Algebra.
  ICTAC'06.
- Murdoch J. Gabbay, Aad Mathijssen:
  One-and-a-halfth-order Logic.
  PPDP'06.


Papers and slides of talks can be found on my web page:
`http://www.win.tue.nl/~amathijs`