Logical Calculi for Reasoning with Binding

Aad Mathijssen

Department of Mathematics and Computer Science
Technische Universiteit Eindhoven

Joint work with Murdoch J. Gabbay

Computer Science Colloquium
Technische Universiteit Eindhoven

7th February 2008

Logic

Logic

Logic studies ***reasoning****.*

Logic

Logic studies ***reasoning****.*

Example

"If I obtain my degree I will give a talk at the symposium."

Logic

Logic studies ***reasoning****.*

Example

"If I obtain my degree I will give a talk at the symposium."

Expressed in logic by the following formula:

$$obtain\_degree \;\Rightarrow\; give\_talk$$

Logic

*Logic studies **reasoning**.*

Example

"If I obtain my degree I will give a talk at the symposium."

Expressed in logic by the following formula:

$$obtain\_degree \;\Rightarrow\; give\_talk$$

"If all human beings are mortal then Socrates is mortal."

Logic

Logic studies ***reasoning****.*

Example

"If I obtain my degree I will give a talk at the symposium."

Expressed in logic by the following formula:

$$obtain\_degree \Rightarrow give\_talk$$

"If all human beings are mortal then Socrates is mortal."

Expressed in logic by the following formula:

$$\forall_{x \in Humans}\, mortal(x) \Rightarrow mortal(Socrates)$$

Reasoning about logics

Reasoning about logics

In many cases we reason about logics:

$$\phi \Rightarrow (\psi \Rightarrow \phi)$$

Reasoning about logics

In many cases we reason about logics:

$$\phi \ \Rightarrow\ (\psi \ \Rightarrow\ \phi)$$

ϕ *and* ψ *are* ***meta-variables*** *ranging over arbitrary formulas.*

We have a ***schema*** *of formulas, one for each instantiation of* ϕ *and* ψ*.*

Reasoning about logics

In many cases we reason about logics:

$$\phi \;\Rightarrow\; (\psi \;\Rightarrow\; \phi)$$

*ϕ and ψ are **meta-variables** ranging over arbitrary formulas.*

*We have a **schema** of formulas, one for each instantiation of ϕ and ψ.*

Example

Take obtain_degree for ϕ and give_talk for ψ:

$$obtain\_degree \;\Rightarrow\; (give\_talk \;\Rightarrow\; obtain\_degree)$$

Reasoning about logics with binders

Reasoning about logics with binders

In many cases we reason about logics with ***binders****, such as* $\forall$*:*

$$\phi \;\Rightarrow\; \forall x.\phi \qquad \textit{if } x \textit{ does not occur free in } \phi$$

$$\forall x.\phi \;\Rightarrow\; \phi[t/x]$$

ϕ is a meta-variable ranging over formulas.
t is a meta-variable ranging over terms.

Reasoning about logics with binders

In many cases we reason about logics with ***binders****, such as* $\forall$*:*

$$\phi \;\Rightarrow\; \forall x.\phi \qquad \textit{if } x \textit{ does not occur free in } \phi$$
$$\forall x.\phi \;\Rightarrow\; \phi[t/x]$$

ϕ *is a meta-variable ranging over formulas.*
t *is a meta-variable ranging over terms.*

We need to define the following concepts:

- *freshness conditions: if* x *does not occur free in* ϕ
- *substitution* $\phi[t/x]$

Observation

If logic teaches us to study reasoning,
we should also study reasoning about logics.

Formalise reasoning about logics with binders

How can we formalise assertions like:

$$\phi \;\Rightarrow\; \forall x.\phi \qquad \text{if } x \text{ does not occur free in } \phi$$

$$\forall x.\phi \;\Rightarrow\; \phi[t/x]$$

Formalise reasoning about logics with binders

How can we formalise assertions like:

$$\phi \;\Rightarrow\; \forall x.\phi \qquad \text{if } x \text{ does not occur free in } \phi$$

$$\forall x.\phi \;\Rightarrow\; \phi[t/x]$$

Model the difference between object- and meta-variables using ***a hierarchy of types****:*

$$P \;\Rightarrow\; \forall(\lambda x.P)$$

$$\forall(\lambda x.F(x)) \;\Rightarrow\; F(T)$$

Formalise reasoning about logics with binders

How can we formalise assertions like:

$$\phi \;\Rightarrow\; \forall x.\phi \qquad \textit{if } x \textit{ does not occur free in } \phi$$

$$\forall x.\phi \;\Rightarrow\; \phi[t/x]$$

Model the difference between object- and meta-variables using ***a hierarchy of types****:*

$$P \;\Rightarrow\; \forall(\lambda x.P)$$

$$\forall(\lambda x.F(x)) \;\Rightarrow\; F(T)$$

Drawbacks:

- *substitution of terms for object-variables is* ***capture-avoiding***
- *representation of meta-variables depends on their* ***context***
- *need unification up to* ***substitution*** *(and* ***extensionality****)*

Formalise reasoning about logics with binders

How can we formalise assertions like:

$\phi \;\Rightarrow\; \forall x.\phi$ *if x does not occur free in ϕ*

$\forall x.\phi \;\Rightarrow\; \phi[t/x]$

Formalise reasoning about logics with binders

How can we formalise assertions like:

$$\phi \;\Rightarrow\; \forall x.\phi \qquad \textit{if } x \textit{ does not occur free in } \phi$$

$$\forall x.\phi \;\Rightarrow\; \phi[t/x]$$

Embrace meta-variables and reject object-variables
by ***adding term-formers and axioms****:*

$$P \;\Rightarrow\; \forall(\mathsf{c}(P)), \qquad \mathsf{c} \textit{ is a constant such that } \mathsf{c}(P)(x) = P$$

$$\forall(\mathsf{d}(F)) \;\Rightarrow\; F(T), \quad \mathsf{d} \textit{ is a constant such that } \mathsf{d}(F)(x) = F(x)$$

Formalise reasoning about logics with binders

How can we formalise assertions like:

$$\phi \;\Rightarrow\; \forall x.\phi \qquad \textit{if } x \textit{ does not occur free in } \phi$$
$$\forall x.\phi \;\Rightarrow\; \phi[t/x]$$

Embrace meta-variables and reject object-variables
by ***adding term-formers and axioms****:*

$$P \;\Rightarrow\; \forall(\mathsf{c}(P)), \qquad \mathsf{c} \textit{ is a constant such that } \mathsf{c}(P)(x) = P$$
$$\forall(\mathsf{d}(F)) \;\Rightarrow\; F(T), \quad \mathsf{d} \textit{ is a constant such that } \mathsf{d}(F)(x) = F(x)$$

Drawbacks:

- *cannot* ***explicitly manipulate*** *bound object-variables*
- *freshness information is* ***encoded*** *in the term structure*

Formalise reasoning about logics with binders

How can we formalise assertions like:

$\phi \Rightarrow \forall x.\phi$ *if x does not occur free in* ϕ

$\forall x.\phi \Rightarrow \phi[t/x]$

Formalise reasoning about logics with binders

How can we formalise assertions like:

$\phi \;\Rightarrow\; \forall x.\phi$ *if x does not occur free in* ϕ

$\forall x.\phi \;\Rightarrow\; \phi[t/x]$

Embrace the difference between object- and meta-variables using ***nominal terms*** *(Urban, Pitts & Gabbay, 2004):*

$a\#P \vdash P \;\Rightarrow\; \forall[a]P$

$\vdash \forall[a]P \Rightarrow\; P[a \mapsto T]$

Formalise reasoning about logics with binders

How can we formalise assertions like:

$$\phi \;\Rightarrow\; \forall x.\phi \qquad \textit{if } x \textit{ does not occur free in } \phi$$

$$\forall x.\phi \;\Rightarrow\; \phi[t/x]$$

Embrace the difference between object- and meta-variables using ***nominal terms*** *(Urban, Pitts & Gabbay, 2004):*

$$a\#P \vdash P \;\Rightarrow\; \forall[a]P$$

$$\vdash \forall[a]P \Rightarrow\; P[a \mapsto T]$$

Drawback:

- *relative new technique: logical frameworks were not available*

Our contribution

Developed two logics to reason about logics with binders based on nominal terms:

Equational logic *with binders and meta-variables:*

- *natural deduction calculus*
- *axiomatisation of the lambda calculus*
- *axiomatisation of capture-avoiding substitution*
- *semantics in nominal sets*

First-order logic *with binders and meta-variables:*

- *sequent calculus*
- *axiomatisation of the sequent calculus*

Nominal terms

Definition:

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

Here we fix:

- ***atoms*** $a, b, c, \ldots$ *(to represent object-variables* x, y*)*
- ***unknowns*** $X, Y, Z, \ldots$ *(to represent meta-variables* ϕ*,* ψ*,* t*)*
- ***term-formers*** $\mathsf{f}, \mathsf{g}, \mathsf{h}, \ldots$ *(for obtain_degree, mortal,* $\Rightarrow$*,* $\forall$*,* _[_ $\mapsto$ _]*)*

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

π *represents a* ***permutation of atoms****:*

- *needed for* α-***conversion***
- *we write id* $\cdot X$ *as* X *where id is the* ***identity permutation***

Freshness on nominal terms

Representation of 'x does not occur free in ϕ':

- ***primitive freshnesses*** $a\#X$
- ***freshness contexts*** Δ*: finite set of primitive freshnesses.*

Freshness on nominal terms

Representation of 'x does not occur free in ϕ':

- ***primitive freshnesses*** *$a\#X$*
- ***freshness contexts*** *Δ: finite set of primitive freshnesses.*

Decidability of freshness:

- ***freshness*** *$a\#t$, where t is a nominal term.*
- *natural deduction rules for freshness:*

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

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

Examples: $\vdash a\#b$ $\quad \vdash a\#\lambda[a]X \quad a\#X \vdash a\#\lambda[b]X$

$\nvdash a\#a \quad \nvdash a\#\lambda[b]X \quad a\#X \nvdash a\#Y$

Equational logic on nominal terms

Natural deduction rules for equality between nominal terms.

Equational logic on nominal terms

Natural deduction rules for equality between nominal terms.

Equivalence and congruence:

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

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

Equational logic on nominal terms

Natural deduction rules for equality between nominal terms.

α-***conversion:***

$$\frac{a\#t \quad b\#t}{(a\ b)\cdot t = t}\,\textbf{(perm)} \quad (a \neq b)$$

Examples:

$$\cfrac{\cfrac{\cfrac{}{a\#b}\,\textbf{(\#ab)}}{a\#[b]b}\,\textbf{(\#[]b)} \qquad \cfrac{}{b\#[b]b}\,\textbf{(\#[]a)}}{[a]a = [b]b}\,\textbf{(perm)} \qquad\qquad \cfrac{\cfrac{a\#X}{a\#[b]X}\,\textbf{(\#[]b)} \qquad \cfrac{}{b\#[b]X}\,\textbf{(\#[]a)}}{[a](b\ a)\cdot X = [b]X}\,\textbf{(perm)}$$

Equational logic on nominal terms

Natural deduction rules for equality between nominal terms.

Instantiation of axioms:

$$\frac{\pi \cdot \Delta\sigma}{\pi \cdot t\sigma = \pi \cdot u\sigma}\,(\mathbf{ax}_{\Delta\,\vdash\,t\,=\,u})$$

Instantiation σ of unknowns is ***capturing****,*
but we need to verify the ***capture-avoiding constraints****.*

Examples:

$$\frac{\dfrac{}{c\#b}\,(\#\mathbf{ab})}{[c]\mathsf{app}(b,c) = b}\,(\mathbf{ax}_{\mathbf{a}\#\mathbf{X}\vdash[\mathbf{a}]\mathsf{app}(\mathbf{X},\mathbf{a})=\mathbf{X}}) \qquad \frac{c\#c}{[c](\mathsf{app}(c,c)) = c}\,(\mathbf{ax}_{\mathbf{a}\#\mathbf{X}\vdash[\mathbf{a}]\mathsf{app}(\mathbf{X},\mathbf{a})=\mathbf{X}})$$

The left derivation is valid but the right one is ***not****, since $\nvdash c\#c$.*

Equational logic on nominal terms

Natural deduction rules for equality between nominal terms.

***Introduce fresh atoms**:*

$$\dfrac{\begin{matrix}[a\#X_1,\ldots,a\#X_n] & \quad \Delta \\ \vdots & \\ t=u & \end{matrix}}{t=u}\,(\mathbf{fr}) \quad (a\notin t,u,\Delta)$$

Example:

$$\dfrac{\dfrac{[a\#X]^1}{X=a}(\mathbf{ax_{a\#X\vdash X=a}}) \qquad \dfrac{\dfrac{[a\#Y]^1}{Y=a}(\mathbf{ax_{a\#X\vdash X=a}})}{a=Y}(\mathbf{symm})}{\dfrac{X=Y}{X=Y}(\mathbf{fr})^1}(\mathbf{tran})$$

Axiomatising the lambda calculus

Term-formers:

- *binary application term-former* app
- *constant term-formers* $\mathsf{c}_1, \ldots, \mathsf{c}_n$

Five axioms:

$(\mathbf{var}\mapsto)$		$\vdash$	$\mathsf{app}([a]a, X) = X$
$(\#\mapsto)$	$a\#Z$	$\vdash$	$\mathsf{app}([a]Z, X) = Z$
$(\mathbf{app}\mapsto)$		$\vdash$	$\mathsf{app}([a](\mathsf{app}(Z', Z), X) = \mathsf{app}(\mathsf{app}([a]Z', X), \mathsf{app}([a]Z, X))$
$(\mathbf{abs}\mapsto)$	$b\#X$	$\vdash$	$\mathsf{app}([a][b]Z, X) = [b]\mathsf{app}([a]Z, X)$
$(\mathbf{id}\mapsto)$		$\vdash$	$\mathsf{app}([a]Z, a) = Z$

Axiomatising the lambda calculus

Term-formers:

- *binary application term-former* app
- *constant term-formers* $\mathsf{c}_1, \ldots, \mathsf{c}_n$

Five axioms:

$(\mathbf{var}\mapsto)$	$\vdash$	$\mathsf{app}([a]a, X) = X$
$(\#\mapsto)$	$a\#Z \vdash$	$\mathsf{app}([a]Z, X) = Z$
$(\mathbf{app}\mapsto)$	$\vdash$	$\mathsf{app}([a](\mathsf{app}(Z', Z), X) = \mathsf{app}(\mathsf{app}([a]Z', X), \mathsf{app}([a]Z, X))$
$(\mathbf{abs}\mapsto)$	$b\#X \vdash$	$\mathsf{app}([a][b]Z, X) = [b]\mathsf{app}([a]Z, X)$
$(\mathbf{id}\mapsto)$	$\vdash$	$\mathsf{app}([a]Z, a) = Z$

Derivability using these axioms is ***sound and complete*** *with respect to a model constructed out of* ***lambda-terms quotiented by*** $\alpha\beta$***-equivalence****.*

A semantics in nominal sets

Nominal sets *(Gabbay & Pitts, 1999):*

- *A set-based model with built-in atoms*
- *Support for binding and freshness*
- *Inspired the development of nominal terms*

Axiomatisations in the equational logic
have a semantics in nominal sets:

- *Derivability of equality is* ***sound*** *and* ***complete***
- *Derivability of freshness is* ***sound*** *but* ***incomplete***
  $a \# \mathsf{app}([a]b, a)$ *is not derivable: independent of axioms*
  $a \# \mathsf{app}([a]b, a)$ *is valid:* $\mathsf{app}([a]b, a) = b$ *is derivable*
  Semantic freshness can be expressed using equalities
- *The semantics satisfies a variant of Birkhoff's theorem:*
  HSPA, where A stands for ***atoms-abstraction***

First-order logic with meta-variables

Terms:

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

Formulas:

$$\begin{array}{lll} \phi & ::= & \pi \cdot P \ \mid \ \bot \ \mid \ \phi \Rightarrow \psi \ \mid \ \forall [a]\phi \ \mid \ \phi[a \mapsto t] \\ & \mid & t \approx u \ \mid \ \mathsf{p}(t_1, \ldots, t_n) \end{array}$$

Sequents: triples $\Phi \vdash_{\Delta} \Psi$ of finite sets of formulas Φ, Ψ and a freshness context Δ

First-order logic with meta-variables

Sequent calculus for first-order logic with meta-variables.

First-order logic with meta-variables

Sequent calculus for first-order logic with meta-variables.

Basic rules:

$$\frac{}{\phi, \Phi \vdash_{\Delta} \Psi, \phi}\,(\mathbf{Ax}) \qquad \frac{}{\bot, \Phi \vdash_{\Delta} \Psi}\,(\bot\mathbf{L})$$

$$\frac{\Phi \vdash_{\Delta} \Psi, \phi \quad \psi, \Phi \vdash_{\Delta} \Psi}{\phi \Rightarrow \psi, \Phi \vdash_{\Delta} \Psi}\,(\Rightarrow\mathbf{L}) \qquad \frac{\phi, \Phi \vdash_{\Delta} \Psi, \psi}{\Phi \vdash_{\Delta} \Psi, \phi \Rightarrow \psi}\,(\Rightarrow\mathbf{R})$$

$$\frac{\phi[a \mapsto t], \Phi \vdash_{\Delta} \Psi}{\forall[a]\phi, \Phi \vdash_{\Delta} \Psi}\,(\forall\mathbf{L}) \qquad \frac{\Phi \vdash_{\Delta} \Psi, \psi}{\Phi \vdash_{\Delta} \Psi, \forall[a]\psi}\,(\forall\mathbf{R}) \quad (\Delta \vdash a\#\Phi, \Psi)$$

$$\frac{\phi[a \mapsto t'],\ \Phi \vdash_{\Delta} \Psi}{t' \approx t,\ \phi[a \mapsto t],\ \Phi \vdash_{\Delta} \Psi}\,(\approx\mathbf{L}) \qquad \frac{}{\Phi \vdash_{\Delta} \Psi,\ t \approx t}\,(\approx\mathbf{R})$$

First-order logic with meta-variables

Sequent calculus for first-order logic with meta-variables.

Special rules:

$$\frac{\phi', \Phi \vdash_{\Delta} \Psi}{\phi, \Phi \vdash_{\Delta} \Psi}\,(\mathbf{StructL}) \quad (\Delta \vdash_{\mathsf{SUB}} \phi' = \phi)$$

$$\frac{\Phi \vdash_{\Delta} \Psi, \psi'}{\Phi \vdash_{\Delta} \Psi, \psi}\,(\mathbf{StructR}) \quad (\Delta \vdash_{\mathsf{SUB}} \psi' = \psi)$$

$$\frac{\Phi \vdash_{\Delta} \Psi, \phi \qquad \phi', \Phi \vdash_{\Delta} \Psi}{\Phi \vdash_{\Delta} \Psi}\,(\mathbf{Cut}) \quad (\Delta \vdash_{\mathsf{SUB}} \phi = \phi')$$

$$\frac{\Phi \vdash_{\Delta, a\#X_1, \ldots, a\#X_n} \Psi}{\Phi \vdash_{\Delta} \Psi}\,(\mathbf{Fr}) \quad (n \geq 1,\ a \notin \Phi, \Psi, \Delta)$$

First-order logic with meta-variables

Sequent calculus for first-order logic with meta-variables.

Example

Meta-level sequent:

$$\phi, \psi \vdash \forall x.\phi, \quad \textit{if } x \textit{ does not occur free in } \phi$$

Formal derivation:

$$\dfrac{\dfrac{\dfrac{\dfrac{}{P, Q \vdash_{a\#P, b\#P, Q} P}\,(\mathbf{Ax})}{P, Q \vdash_{a\#P, b\#P, b\#Q} \forall[b]P}\,(\forall\mathbf{R}) \quad (a\#P, b\#P, b\#Q \vdash b\#P, Q)}{P, Q \vdash_{a\#P, b\#P, b\#Q} \forall[a]P}\,(\mathbf{StructR}) \quad (a\#P, b\#P, b\#Q \vdash_{\mathsf{SUB}} \forall[b]P = \forall[a]P)}{P, Q \vdash_{a\#P} \forall[a]P}\,(\mathbf{Fr}) \quad (b \notin P,\ Q,\ \forall[a]P,\ a\#P)$$

First-order logic with meta-variables

Proof-theoretical results:

- *In derivations we may* ***permute*** *atoms and* ***instantiate*** *unknowns*
- *The sequent calculus satisfies* ***cut-elimination****, and is* ***consistent***
- *Without unknowns or explicit substitutions, the sequent calculus is* ***equivalent*** *to* ***Gentzen's sequent calculus*** *for first-order logic*

An axiomatisation of first-order logic

Consider the following axioms:

- *Substitution axioms: similar to those for the lambda calculus*
- *Propositional axioms, e.g. axioms of boolean algebra*
- *Quantifier axioms:*

(Qinst)		$\vdash$	$\forall[a]P \Rightarrow P[a \mapsto T] = \top$
(Qdist)		$\vdash$	$\forall[a](P \wedge Q) \Leftrightarrow \forall[a]P \wedge \forall[a]Q = \top$
(Qextr)	$a\#P$	$\vdash$	$\forall[a](P \Rightarrow Q) \Leftrightarrow P \Rightarrow \forall[a]Q = \top$

- *Equality axioms:*

(Esubst)		$\vdash$	$U \approx T \wedge P[a \mapsto T] \Rightarrow P[a \mapsto U] = \top$
(Erefl)		$\vdash$	$T \approx T = \top$

An axiomatisation of first-order logic

Consider the following axioms:

- *Substitution axioms: similar to those for the lambda calculus*
- *Propositional axioms, e.g. axioms of boolean algebra*
- *Quantifier axioms:*

$$\begin{array}{lrl} (\textbf{Qinst}) & & \vdash \forall[a]P \Rightarrow P[a \mapsto T] = \top \\ (\textbf{Qdist}) & & \vdash \forall[a](P \wedge Q) \Leftrightarrow \forall[a]P \wedge \forall[a]Q = \top \\ (\textbf{Qextr}) & a\#P & \vdash \forall[a](P \Rightarrow Q) \Leftrightarrow P \Rightarrow \forall[a]Q = \top \end{array}$$

- *Equality axioms:*

$$\begin{array}{lrl} (\textbf{Esubst}) & & \vdash U \approx T \wedge P[a \mapsto T] \Rightarrow P[a \mapsto U] = \top \\ (\textbf{Erefl}) & & \vdash T \approx T = \top \end{array}$$

This is a ***sound and complete axiomatisation*** *of the sequent calculus for first-order logic with meta-variables.*

Conclusions

Using nominal terms we can ***formalise the meta-level*** *of* ***logics with binding*** *in a way that is* ***close to informal practice****:*

- *Developed calculi for equational logic and first-order logic with binders and meta-variables.*
- *Established proof-theoretical and algebraic results.*

Conclusions

*Using nominal terms we can **formalise the meta-level** of **logics with binding** in a way that is **close to informal practice**:*

- *Developed calculi for equational logic and first-order logic with binders and meta-variables.*
- *Established proof-theoretical and algebraic results.*

We're not there yet:

- ***Usability**: extend the logics with more features to support reasoning*
- ***Implementation**: develop a theorem prover*
- ***Methodology**: apply the technique to other systems with binding*

If your interested

Aad Mathijssen:
Logical Calculi for Reasoning with Binding.
PhD Thesis.

Murdoch J. Gabbay, Aad Mathijssen:
A Formal Calculus for Informal Equality with Binding.
In: Proc. WoLLIC'07.

Murdoch J. Gabbay, Aad Mathijssen:
Capture-Avoiding Substitution as a Nominal Algebra.
In: Proc. ICTAC'06.
Extended version in: Formal Aspects of Computing (in print).

Murdoch J. Gabbay, Aad Mathijssen:
One-and-a-halfth-order Logic.
In: Proc. PPDP'06.
Extended version in: Journal of Logic and Computation (in print).