Extensions of Nominal Terms.

Murdoch J. Gabbay
Work with/thanks to: Maribel Fernández, Ian Mackie, Alexis Saurin, and Antonino Salibra

March 11, 2004

Introduction

My esteemed colleague Maribel has just given an outstanding presentation of Nominal Rewriting.

Nominal Rewriting can be described as “rewriting for Nominal Signatures and Terms”. These were originally introduced for Nominal Unification in work with Christian Urban and Andrew Pitts.

Nominal Signatures Σ

Fix $\mathcal{S}$ base data sorts typically called s, for example integer, boolean.

A Nominal Signature Σ is:

1. A set of sorts of atoms typically written ν.
2. A set of data sorts typically written δ. δ can be a base data sort or a product:
$$\delta ::= s \mid \delta \times \delta.$$
3. Compound (data) sorts typically written τ are then *defined* by the following grammar:
$$\tau ::= \nu \mid \delta \mid 1 \mid \tau \times \tau \mid [\nu]\tau.$$

So nominal signatures are like ordinary signatures, though some sorts are sorts of atoms (whatever they are), and we can abstract over them (whatever that means).

Examples of nominal signatures Σ

λ-calculus: A data sort Λ and constants $\lambda : [\nu]\Lambda \rightarrow \Lambda$ and $app : \Lambda \times \Lambda \rightarrow \Lambda$.

π-calculus (1): A data sort Π and constants $res : [\nu]\Pi \rightarrow \Pi$, $\mid : \Pi \times \Pi \rightarrow \Pi$, $in : \nu \times [\nu]\Pi \rightarrow \Pi$, ...

Fix some Σ.

Terms

For each sort τ fix $\mathcal{X}_\tau$ term variables X_τ, Y_τ, Z_τ.

For each *atoms* sort ν fix $\mathcal{A}_\nu$ atoms $a_\nu, b_\nu, c_\nu, f_\nu, g_\nu, h_\nu, \ldots$.

A swapping is a pair $(a_\nu\ b_\nu)$. Permutations π are lists of swappings, write **Id** for the empty list.

Nominal Terms are generated by the following grammar:

$$t ::= a_\nu \mid *_1 \mid \langle t_\tau, t'_{\tau'} \rangle_{\tau\times\tau'} \mid (f_{\tau\to\delta} t_\tau)_\delta \mid ([a_\nu]t_\tau)_{[\nu]\tau} \mid \pi \cdot X_\tau$$

Again without sorts:

$$t ::= a \mid * \mid \langle t, t' \rangle \mid f\ t \mid [a]t \mid \pi \cdot X$$

Terms

Write X for **Id** $\cdot\, X$. Informally $(a\ b) \cdot X$ is "swap a and b in X when instantiated". Informally, $[a]t$ means "t with a bound".

An apartness condition is a pair $a\#X$. $a\#X$ means "a does not occur in X, when it is instantiated". Apartness contexts $\Delta, \nabla, \Gamma, \ldots$ are finite sets of apartness conditions.

Nominal rewriting/unification work with terms-in-context, $\nabla \vdash s$. We get a critical pair theorem, decidable unification, most general unifiers. It's nice.

What? An example? The unification problem $b\#X \vdash [a]X \stackrel{?}{=}_{?} [b]Y$ has solution $X = (a\ b) \cdot Y$. E.g. take $X = ac$ (as might arise unifying $\lambda[a]ac$ and $\lambda[b]bc$). Then $Y = bc$.

Enrich the apartness context

So what next?

Enrich the contexts. Possible judgements are:

1. $a\#X$ read 'a fresh for X' as before.
2. $a \in_{\#} X$, meaning 'a is *not* fresh for X'. Similar to the HOAS Xa, which says that X *may* use a. Useful for occurs-checks, e.g. in ripplying and rewriting.
3. $a \in_1 X$ meaning 'a occurs precisely once in X'.
4. $a \in_{\leq 1} X$.
5. $ev(X)$ meaning X contains no (unabstracted) at all; 'is closed'.

Enrich the apartness context

An apartness context ∇ is now a conjunction of disjunctions. Example deduction rules are:

$$\frac{a \in_{\#} t \vee a \in_{\#} t'}{a \in_{\#} \langle t, t' \rangle} \qquad \frac{a \in_1 t \quad a\#t'}{a \in_1 \langle t, t' \rangle} \qquad \frac{a \in_{\leq 1} t \quad a\#t'}{a \in_{\leq 1} \langle t, t' \rangle} \qquad \frac{ev(t)}{a\#t}$$

$a \in_{\geq 1}$ is not so easy, what is the deduction rule for judgements of the form $a \in_{\geq 1} [a]t$?

The nice thing is that this is entirely technical: work out what the deduction rules must be, verify that a known list of lemmas and algorithms is not broken (culminating in a decidable algorithm for calculating MGUs). No imagination required.

Summary

"Nominal Terms (in context)" equals "first-order analysis of α-conversion". Associated is a semantics FM sets (Schanuel topos), so we even have a useful sanity check.

This can be applied to:

1. Unification (Nominal Unification).
2. Rewriting (Nominal Rewriting).
3. Universal Algebra (We know what to do).

Note that via the FM semantics, any universal algebra theory automatically gets a semantics for free. Let's look at this more closely.

An nominally universal algebraic theory for the λ-calculus

So let's look at a Nominal Universal Algebra theory. One sort of atoms ν and one data sort Λ. The signature has three constructors:

$$[\text{-}\mapsto\text{-}] : [\nu]\Lambda \times \Lambda \to \Lambda \quad \lambda : [\nu]\Lambda \to \Lambda \quad app : \Lambda \times \Lambda \to \Lambda$$

Axioms of substitution:

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

Distributivity axioms:

$$\vdash (XX')[a \mapsto Y] = (X[a \mapsto Y])(X'[a \mapsto Y])$$
$$a\#X' \vdash (\lambda[a]X)[b \mapsto X'] = \lambda[a](X[b \mapsto X'])$$

Other axioms:

$$(\beta) \quad \lambda([a]X)X' = X[a \mapsto X'] \qquad (\eta) \quad a\#X \vdash X = \lambda[a](Xa)$$

Lambda-abstraction algebras by Salibra

Theorems of Nominal Universal Algebra (yet to be proved) say “any model is a homomorphic image of a subalgebra of a direct product of the free term algebra”.

Already done for signature above: Salibra’s Lambda Abstraction Algebras. Who knows what else is out there.

1. Is the category of models of the algebraic theory of the λ/π-calculus above cartesian closed?
2. What is it for a class of algebras to be ‘Turing complete/computationally non-trivial’.
3. In this context, what is an ‘evaluation function’.
4. What is the behaviour of nominal unification/rewriting *modulo* nominal equational theories?

Let’s look at the last point.

Extend equalities

Consider the Nominal Unification algorithm modulo fragments of theories such as that for substitution and λ above. How decidable are those fragments? MGUs? Relation to HO unification and friends?

Difference from existing work: we have the apartness contexts, so we can work 'modulo equalities *in context*'.

Extend atoms

Finally, we can extend atoms. For every sort of atoms ν take another sort of atoms ν_ω representing a countably infinite stream of distinct atoms.

I have developed the semantics for this. What happens to nominal terms? There are functions $cons : \nu \times \nu_\omega \rightarrow \nu_\omega$ and deduction rules such as $\dfrac{a\#\alpha}{a\#b :: \alpha}$

Here is a signature for the $\lambda\mu$-calculus (recent work with Alexis Saurin):

$$t_\Lambda[a_\nu \mapsto t'_\Lambda] \quad t\{\alpha_{\nu_\omega} \mapsto t'_{\nu_\omega}\} \quad \lambda[a]t \quad tt' \quad \mu[\alpha]t \quad t\alpha.$$

This raises the strange question of what sense, if any, does adding a theory of streams of atoms make nominal terms 'classical'?

Conclusions

Nominal Terms (in apartness context) give a framework to analyse α-equivalence.

1. We can probably enrich the contexts to analyse ‘occurs’ properties.
2. I claim that substitution is precisely what makes β-reduction (and computation) difficult to express algebraically. Therefore, nominal terms are a good place to analyse computation!
3. Put another way, it is interesting to investigate nominal terms up to nominal equational theories; computationally (unification, matching) as well as mathematically (algebras, categories).
4. Other interesting extensions are possible, such as streams of atoms, and they are not gratuitous, in the sense that surprising connections exist with other work.