Available online at www.sciencedirect.com
ScienceDirect
Electronic Notes in Theoretical Computer Science
ELSEVIER
Electronic Notes in Theoretical Computer Science 265 (2010) 259–276
www.elsevier.com/locate/entcs

Binding in Nominal Equational Logic

Ranald Clouston[1,2]

Computer Laboratory, University of Cambridge, Cambridge CB3 0DF, United Kingdom

Abstract

Many formal systems, particularly in computer science, may be expressed through equations modulated by assertions regarding the 'freshness of names'. It is the presence of binding operators that make such structure non-trivial. Clouston and Pitts's Nominal Equational Logic presented a formalism for this style of reasoning in which support for name binding was implicit. This paper extends this logic to offer explicit support for binding and then demonstrates that such an extension does not in fact add expressivity.

Keywords: Equational logic, nominal sets, name binding, alpha-conversion, nominal signatures.

1 Introduction

Formal languages in logic and computer science frequently utilise *names*, and equations that are modulated by side conditions concerning names, such as η-equivalence in the λ-calculus:

$$(1) \qquad \lambda\, a.f\, a \;=_\eta\; f \qquad \text{if } a \text{ does not appear free in } f\;.$$

In the language of this paper, this side condition requires that a *is fresh for* f.

Such freshness conditions are trivial without the presence of operators that bind names in their arguments. For example, the λ-operator above binds one name in its argument. Given such operators we will generally want to work modulo α-equivalence, identifying terms that only differ uniformly in their bound names.

A variety of approaches have been suggested to formalise these ideas, most notably higher order abstract syntax (HOAS) [16] and the nameless approach of de Bruijn [8]. These approaches have their advantages, but have been criticised [4,2] for departing too far from intuitive pen-and-paper reasoning, which often involves the explicit manipulation of bound variables.

[1] The author gratefully acknowledges the supervision and advice of Andrew Pitts, the comments of the reviewers, and the financial and pastoral support of the Woolf Fisher Trust.

[2] Email: ranald@cloustons.org

doi:10.1016/j.entcs.2010.08.016

The Gabbay-Pitts *FM-sets* model [13], and its subsequent refinement to the *nominal sets* model [17], consider names as first class syntactic entities that may be *permuted*. This notion of the permutation of names turns out to be sufficient to formalise freshness, binding and α-equivalence, while remaining close to intuitive practice and allowing these ubiquitous concepts to be studied as mathematical objects in their own right. A variety of work has been done within this model, including Nominal Equational Logic (NEL) [7], a logic designed to express equations modulated by freshness conditions, as with (1).

One difference between NEL and other approaches to these problems, such as Nominal Algebra [12], is that NEL employs a very simple notion of signature that does not explicitly support binding structure. Examples were given in [7] to indicate that such structure could often be encoded through the axioms of a NEL-theory, but it is an original contribution of this paper that this can be done in general. This is of interest because it shows us that binding is not logically fundamental to reasoning with names, however practically important it may be, so we may do much basic work without having to worry about such structure. The simplicity of NEL's signatures simplify the many proofs by induction on the structure of terms that are necessary in developing the metatheory of NEL, while their closeness to the signatures of standard equational logic make it easier to develop algebraic accounts of NEL by analogy with the standard development, as is done extensively in [6].

This paper proceeds by defining *binding NEL*, an extension of NEL with explicit support for binding based on the binding signatures of [10]. Sec. 6 then defines a translation from binding NEL back to NEL, and the main result of this paper, Thm. 6.9, shows that this translation maintains expressivity with respect to semantic consequence. Sec. 7 then presents an analogous result where binding is represented through nominal signatures [20]. While these extensions of NEL offer no more expressivity, it may sometimes be desirable to have binding structure built in at the lowest level of the logic on grounds of efficiency or usability, so these definitions are of interest in their own rights.

Finally, having proved that NEL may express operators with binding structure, Sec. 8 shows that the related but stronger property of characterising the nominal sets that model binding structure is not possible, and briefly discusses how NEL could be extended with that expressivity.

This paper is based on Chap. 4 of the author's thesis [6]. There, NEL is defined in a slightly more general setting where the sorts of a signature form a nominal set rather than a set, but this generalisation has no bearing on the results of this paper, so we will work with NEL as first defined in [7].

2 Nominal sets

Definition 2.1 We start by fixing a countably infinite set $\mathbb{A}$ of *atoms*, which will capture the notion of names in applications of NEL. We assume that $\mathbb{A}$ is partitioned into countably infinitely many different *atom sorts*, each countably infinite themselves. Call this set of atom sorts $\mathbb{A}\mathsf{Sort}$, with typical member ν, and write the set of all atoms of sort ν as $\mathbb{A}_\nu$. Note that we will not always specify the sort of an atom where that is clear from context.

Definition 2.2 The set Perm of (finite, sort-respecting) *permutations* of atoms consists of all bijections $\pi : \mathbb{A} \to \mathbb{A}$ such that

- $\{a \in \mathbb{A} \mid \pi(a) \neq a\}$ is finite;
- For each atom a, the atom $\pi(a)$ has the same sort as a.

Perm is generated by transpositions $(a\ a')$ that send a to a', a' to a and leave all other atoms unchanged. It is a group whose multiplication is functional composition, $\pi'\pi(a) = \pi'(\pi(a))$, and identity is the permutation ι, where $\iota(a) = a$ for all a.

As such we can define Perm-sets in the usual way, as a set X equipped with a Perm-*action* $\mathsf{Perm} \times X \to X$ mapping $(\pi, x) \mapsto \pi \cdot x$ so that $\iota \cdot x = x$ and $\pi' \cdot (\pi \cdot x) = \pi'\pi \cdot x$.

Definition 2.3 Given a Perm-set X, a set of atoms $\overline{a} \subseteq \mathbb{A}$ is said to *support* $x \in X$ if for all $a, a' \notin \overline{a}$, $(aa') \cdot x = x$. x is *finitely supported* if there exists a finite set $\overline{a}$ supporting x.

A Perm-set X is a *nominal set* if every $x \in X$ is finitely supported.

Definition 2.4 In fact, given any Perm-set X and $x \in X$, if x is finitely supported then there is a least such finite support [13, Prop. 3.4]. Call this set *the support of* x and write it $supp(x)$.

If $a \notin supp(x)$ then we say that a *is fresh for* x and write $a \# x$. If $a \# x$ for all a in some set of atoms $\overline{a}$ then we write $\overline{a} \# x$.

Example 2.5 The set of atoms $\mathbb{A}$, and any subset $\mathbb{A}_\nu$ of a particular sort, are nominal sets under the action $\pi \cdot a \triangleq \pi(a)$; then $supp(a) = \{a\}$. Similarly $\mathcal{P}_{fin}(\mathbb{A})$, the set of finite sets of atoms, is a nominal set under the action $\pi \cdot \overline{a} \triangleq \{\pi(a) \mid a \in \overline{a}\}$, so $supp(\overline{a}) = \overline{a}$.

Example 2.6 Given nominal sets X, Y the Cartesian product $X \times Y$ is a nominal set with the obvious Perm-action, $\pi \cdot (x, y) = (\pi \cdot x, \pi \cdot y)$, so that $supp(x, y) = supp(x) \cup supp(y)$. There is another notion of a tensor product of nominal sets however, that of the *separated tensor*

$$X \otimes Y \triangleq \{(x, y) \in X \times Y \mid supp(x) \cap supp(y) = \emptyset\} \tag{2}$$

with Perm-action and support sets defined as for Cartesian product.

Example 2.7 Nominal sets of lists of atoms can be constructed via Exs. 2.5 and 2.6. Given a list of atom sorts $\vec{\nu} = (\nu_1, \ldots \nu_n)$, let $\mathbb{A}_{\vec{\nu}}$ be the set of all lists of atoms $(a_1 \ldots, a_n)$ such that $a_i \in \mathbb{A}_{\nu_i}$ for $1 \leq i \leq n$. This is a nominal set given the pointwise Perm-action

$$\pi \cdot (a_1, \ldots, a_n) \triangleq (\pi(a_1), \ldots, \pi(a_n)) \ . \tag{3}$$

Clearly $supp(a_1, \ldots, a_n) = \{a_1, \ldots, a_n\}$. The nominal set $\mathbb{A}^{(*)}_{\vec{\nu}}$ is defined via the separated tensor (2) as the subset of $\mathbb{A}_{\vec{\nu}}$ of lists of atoms whose members are mutually distinct, with Perm-action and support as above.

Given two lists $\vec{a} = (a_1, \ldots, a_n)$, $\vec{a}' = (a'_1, \ldots, a'_n)$ in $\mathbb{A}_{\vec{\nu}}$ we define their *generalised transposition* as

$$(\vec{a}\ \vec{a}') \;\triangleq\; (a_1\ a'_1) \cdots (a_n\ a'_n) \ . \tag{4}$$

Definition 2.8 A *finitely supported function* is a function between nominal sets, $f : X \to Y$, that is finitely supported with respect to the Perm-action

$$(\pi \cdot f)(x) \;\triangleq\; \pi \cdot f(\pi^{-1} \cdot x) \ .$$

We call a function that is emptily supported under this definition an *equivariant function*; that is, $\pi \cdot (f(x)) = f(\pi \cdot x)$. The category $\mathcal{N}om$ of nominal sets has equivariant functions as its morphisms. Equivariant functions also have the property that they do not increase the support of their arguments:

$$a \# x \Rightarrow a \# f(x) \ . \tag{5}$$

3 Atom Abstractions

The nominal sets model supports a well established notion to describe binding operators, that of *atom abstraction* [17, Sec. 7], generalising the equational rules for α-equivalence over λ-operators presented in [15, Chap. 2].

Definition 3.1 Given an atom sort ν and nominal set X we define a relation on $\mathbb{A}_\nu \times X$, $(a, x) \sim (a', x')$, by

$$(a, x) \sim (a', x') \;\Leftrightarrow\; (a\ b) \cdot x = (a'\ b) \cdot x'$$

for some atom $b \# (a, a', x, x')$. This clearly defines an equivalence relation on $\mathbb{A}_\nu \times X$; write the equivalence class containing (a, x) as $\langle a \rangle x$ and call such a class the *atom abstraction of a in x*.

In fact this relation is not only an equivalence but also *equivariant*, in that

$$(a, x) \sim (a', x') \;\Rightarrow\; (\pi(a), \pi \cdot x) \sim (\pi(a'), \pi \cdot x') \ . \tag{6}$$

We may hence construct a new nominal set by taking the quotient over this relation:

Definition 3.2 Given an atom sort ν and nominal set X the *nominal set of atom abstractions of sort ν on X* is defined by

$$[\mathbb{A}_\nu]X \;\triangleq\; \{\langle a \rangle x \mid a \in \mathbb{A}_\nu \wedge x \in X\}$$

with Perm-action $\pi \cdot (\langle a \rangle x) \;\triangleq\; \langle \pi(a) \rangle (\pi \cdot x)$, which is well-defined by (6).

Lemma 3.3 *Take a nominal set X, element $x \in X$ and atom a. Then*

$$supp(\langle a \rangle x) = supp(x) - \{a\} \ .$$

Proof. [17, Prop. 5]. □

It will be necessary to consider situations when a finite list of atoms, rather than a single atom, is abstracted:

Definition 3.4 Suppose we have a list of atom sorts $\vec{\nu} = (\nu_1, \ldots, \nu_n)$ and a list of atoms $\vec{a} = (a_1, \ldots, a_n) \in \mathbb{A}_{\vec{\nu}}$. Then we will write

$$\langle a_1 \rangle (\cdots (\langle a_n \rangle x) \cdots) \in [\mathbb{A}_{\nu_1}](\cdots ([\mathbb{A}_{\nu_n}]X) \cdots)$$

as $\langle \vec{a} \rangle x \in [\mathbb{A}_{\vec{\nu}}]X$.

Lemma 3.5 *Take $\langle \vec{a} \rangle x, \langle \vec{a}' \rangle x' \in [\mathbb{A}_{\vec{\nu}}]X$ as above. Then*

$$\langle \vec{a} \rangle x = \langle \vec{a}' \rangle x' \;\Leftrightarrow\; (\vec{a}\;\vec{b}) \cdot x = (\vec{a}'\;\vec{b}) \cdot x'$$

where $\vec{b} \in \mathbb{A}^{(*)}_{\vec{\nu}}$ is a list of distinct atoms such that $supp(\vec{b}) \mathrel{\#} (\vec{a}, \vec{a}', x, x')$ and $(\vec{a}\;\vec{b})$, $(\vec{a}'\;\vec{b})$ are generalised transpositions as in (4).

Proof. By induction of the length of $\vec{a}$; see [6, Lem. 4.1.6] for details. □

An immediate corollary of this is that for all $\vec{a} \in \mathbb{A}_{\vec{\nu}}$,

$$\langle \vec{a} \rangle x \;=\; \langle \vec{b} \rangle ((\vec{a}\;\vec{b}) \cdot x) \tag{7}$$

where $\vec{b} \in \mathbb{A}^{(*)}_{\vec{\nu}}$ and $supp(\vec{b}) \mathrel{\#} (\vec{a}, x)$.

4 Nominal Equational Logic

This section presents Nominal Equational Logic (NEL) [7]. We will postpone giving examples of theories of this logic until Sec. 6, after we have shown how binding may be encoded through the simple signatures of NEL.

Definition 4.1 A *NEL-signature* Σ is an ordinary algebraic signature whose operation symbols form a nominal set and whose typing function is equivariant. Fixing our notation, Σ comprises of

- a set Sort_Σ, whose elements are called the *sorts of* Σ;
- a nominal set Op_Σ, whose elements are called the *operation symbols of* Σ; and
- an equivariant *typing function* that assigns to each $op \in \mathsf{Op}_\Sigma$ a *type* consisting of a finite (possibly empty) list $\vec{\mathsf{s}} = [\mathsf{s}_1, \ldots, \mathsf{s}_n]$ of sorts of Σ and a sort s of Σ. As usual, the list $\vec{\mathsf{s}}$ indicates the number and sort of arguments that op accepts and s indicates the sort of result it returns. We write

  $$op : \vec{\mathsf{s}} \to \mathsf{s}$$

  to indicate this typing information.

Definition 4.2 Fixing a countably infinite set Var of *variables*, the grammar of terms over a NEL-signature Σ is given by terms

$$t \quad ::= \quad \pi\, x \mid op\, t \cdots t$$

where $\pi \in \mathsf{Perm}$, $x \in \mathsf{Var}$ and $op \in \mathsf{Op}_\Sigma$. We call $\pi\, x$ a *suspension* and $op\, t_1 \cdots t_n$ a *constructed term*.

All occurrences of variables x in terms are preceded by a suspended permutation π, but where π is the identity permutation ι we shall abbreviate $\iota\, x$ just to x.

Definition 4.3 A *sorting environment* over a NEL-signature Σ is a partial function $\Gamma : \mathsf{Var} \rightharpoonup \mathsf{Sort}_\Sigma$ with finite domain $dom(\Gamma) \subseteq \mathsf{Var}$.

Definition 4.4 The sets $\Sigma_{\mathsf{s}}(\Gamma)$ of *terms of sort* $\mathsf{s} \in \mathsf{Sort}_\Sigma$ *in a sorting environment* Γ are inductively defined by:

- if $\pi \in \mathsf{Perm}$, $x \in dom(\Gamma)$ and $\Gamma(x) = \mathsf{s}$, then $\pi\, x \in \Sigma_{\mathsf{s}}(\Gamma)$;
- if $op \in \mathsf{Op}_\Sigma$ has type $[\mathsf{s}_1, \ldots, \mathsf{s}_n] \to \mathsf{s}$ and $t_i \in \Sigma_{\mathsf{s}_i}(\Gamma)$ for $1 \le i \le n$, then $op\, t_1 \cdots t_n \in \Sigma_{\mathsf{s}}(\Gamma)$.

We wish to be able to modulate our equations by assertions regarding the freshness of names for variables, so we need to define a richer notion of environment than the sorting environments of Def. 4.3 before we define our judgements and semantics.

Note that from here on we will use the symbols "$\approx$" and "$\not\approx\!\!\#$" to refer to the notions of equality and freshness defined by NEL, and continue to use "$=$" and "$\#$" for the actual equality and "not-in-the-support-of" relations in $\mathcal{N}om$.

Definition 4.5 A *freshness environment* over a NEL-signature Σ is a partial function $\nabla : \mathsf{Var} \rightharpoonup \mathsf{Sort}_\Sigma \times \mathcal{P}_{fin}(\mathbb{A})$ with finite domain, mapping each variable in $dom(\nabla)$ to a sort of Σ and finite set of atoms. If $dom(\nabla)$ is $\{x_1, \ldots, x_n\}$, and $\nabla(x_i) = (\mathsf{s}_i, \overline{a}_i)$ for $1 \le i \le n$, then we write ∇ as

$$[\overline{a}_1 \not\approx\!\!\# \, x_1 : \mathsf{s}_1, \ldots, \overline{a}_n \not\approx\!\!\# \, x_n : \mathsf{s}_n] \ . \tag{8}$$

Given any freshness environment ∇ we can derive a sorting environment $\nabla^{:}$ by taking the first projection.

Definition 4.6 We wish NEL to be expressive enough to reason about both equality between terms and the freshness of atoms for terms. Rather than use separate judgements for equality and freshness, it is convenient to roll both into a single judgement form. So we define a *NEL-judgement* over a signature Σ to have the form

$$\nabla \vdash \overline{a} \not\approx\!\!\# \, t \approx t' : \mathsf{s}$$

where ∇ is a freshness environment over Σ (Def. 4.5), $\overline{a} \in \mathcal{P}_{fin}(\mathbb{A})$, $\mathsf{s} \in \mathsf{Sort}_\Sigma$ and $t, t' \in \Sigma_{\mathsf{s}}(\nabla^{:})$ (Def. 4.4), where $\nabla^{:}$ is defined as above.

Although this single form of judgement combining equality and freshness is usefully compact, in particular cases it is clearer to use the following abbreviations:

- $t \approx t' : \mathsf{s}$ means $\emptyset \not\approx\!\!\# \, t \approx t' : \mathsf{s}$ and, in a freshness environment, $x : \mathsf{s}$ means $\emptyset \not\approx\!\!\# \, x : \mathsf{s}$.
- $\overline{a} \not\approx\!\!\# \, t : \mathsf{s}$ means $\overline{a} \not\approx\!\!\# \, t \approx t : \mathsf{s}$.
- $a \not\approx\!\!\# \, t \approx t' : \mathsf{s}$ means $\{a\} \not\approx\!\!\# \, t \approx t' : \mathsf{s}$ and $a \not\approx\!\!\# \, x : \mathsf{s}$ means $\{a\} \not\approx\!\!\# \, x : \mathsf{s}$.

Definition 4.7 A *NEL-theory* $\mathbb{T}$ consists of a NEL-signature Σ together with a set of NEL-judgements over Σ, which we call *axioms*.

Definition 4.8 Given a NEL-signature Σ, a Σ-*structure* M in the category $\mathcal{N}om$ is specified by

- a nominal set $M[\![\mathsf{s}]\!]$ for each sort s of Σ;
- an equivariant map from each operation symbol $op \in \mathsf{Op}_\Sigma$ with type $\vec{\mathsf{s}} \to \mathsf{s}$ to a finitely supported function

$$M[\![op]\!] : M[\![\vec{\mathsf{s}}]\!] \to M[\![\mathsf{s}]\!]$$

where if $\vec{\mathsf{s}} = [\mathsf{s}_1, \ldots, \mathsf{s}_n]$ then $M[\![\vec{\mathsf{s}}]\!] \triangleq M[\![\mathsf{s}_1]\!] \times \cdots \times M[\![\mathsf{s}_n]\!]$.

Where the name of the structure is not important we will omit it, writing $M[\![\mathsf{s}]\!]$ as $[\![\mathsf{s}]\!]$ and so forth.

Definition 4.9 Take a a NEL-signature Σ, a sorting environment Γ for that signature and some Σ-structure. Let the set $[\![\Gamma]\!]$ of Γ-*valuations* be the functions ρ mapping each $x \in dom(\Gamma)$ to elements $\rho(x)$ of $[\![\Gamma(x)]\!]$.

The intended meaning of a freshness environment (8) is to assert not only that each variable x_i has sort s_i, but also that it stands for an element of the corresponding nominal set whose support is disjoint from $\overline{a}_i$. Therefore we define the set of ∇-*valuations* as

$$[\![\nabla]\!] \triangleq \{\rho \in [\![\nabla^{:}]\!] \mid \overline{a}_1 \# \rho(x_1) \wedge \cdots \wedge \overline{a}_n \# \rho(x_n)\}$$

where $\#$ is the freshness relation on each $[\![\mathsf{s}_i]\!]$ and $\nabla^{:}$ is the sorting environment associated with ∇ by Def. 4.5.

Definition 4.10 The *value* $[\![t]\!]\rho$ of a well-sorted term $t \in \Sigma_{\mathsf{s}}(\Gamma)$ with respect to a valuation $\rho \in [\![\Gamma]\!]$ is an element of $[\![\mathsf{s}]\!]$ defined by

$$\begin{aligned} [\![\pi\, x]\!]\rho &\triangleq \pi \cdot \rho(x) \ ; \\ [\![op\, t_1 \cdots t_n]\!]\rho &\triangleq [\![op]\!]([\![t_1]\!]\rho, \ldots, [\![t_n]\!]\rho) \ . \end{aligned} \tag{9}$$

Definition 4.11 Let Σ be a NEL-signature. A Σ-structure *satisfies* a NEL-judgement $\nabla \vdash \overline{a} \not\# t \approx t' : \mathsf{s}$ if for all $\rho \in [\![\nabla]\!]$ it is the case both that $\overline{a} \# [\![t]\!]\rho$ and that $[\![t]\!]\rho = [\![t']\!]\rho$ in $[\![\mathsf{s}]\!]$.

Given a NEL-theory $\mathbb{T}$ over Σ, a $\mathbb{T}$-*algebra* is a Σ-structure that satisfies the axioms of $\mathbb{T}$. Given a judgement $\nabla \vdash \overline{a} \not\# t \approx t' : \mathsf{s}$, the *semantic consequence relation*

$$\nabla \vDash_{\mathbb{T}} \overline{a} \not\# t \approx t' : \mathsf{s}$$

is defined to hold if all $\mathbb{T}$-algebras satisfy the judgement.

Sound and complete proof rules for NEL are provided in [7], but we will prove the results of this paper directly on the semantics.

5 NEL with binders

This section will extend the definitions of the previous section to deal with binding operators, defining *binding NEL*. This presentation, inspired by the binding signatures of [10], will use the atom abstractions of Sec. 3.

Definition 5.1 A *binding NEL-signature* Σ is specified by a set of sorts Sort_Σ and nominal set of operation symbols Op_Σ as in Def. 4.1. The equivariant typing function now assigns to each $op \in \mathsf{Op}_\Sigma$ a type consisting of a finite (possibly empty) list of pairs $(\vec{\nu}_i, \mathsf{s}_i)$, where $\vec{\nu}_i$ is a finite (possibly empty) list of atom sorts and $\mathsf{s}_i \in \mathsf{Sort}_\Sigma$, along with a result sort s. If op is assigned n arguments we write this

$$op : [\vec{\nu}_1.\,\mathsf{s}_1, \ldots, \vec{\nu}_n.\,\mathsf{s}_n] \to \mathsf{s} \ . \tag{10}$$

The intended meaning is that op binds a list of atoms of sort $\vec{\nu}_i$ in its i'th argument. We use the abbreviations s for $().\,\mathsf{s}$ and $\nu.\,\mathsf{s}$ for $(\nu).\,\mathsf{s}$.

Definition 5.2 Given a binding NEL-signature Σ, the sets $\Sigma_{\mathsf{s}}(\Gamma)$ of terms of sort $\mathsf{s} \in \mathsf{Sort}_\Sigma$ in a sorting environment Γ are inductively defined by:

- if $\pi \in \mathsf{Perm}$, $x \in dom(\Gamma)$ and $\Gamma(x) = \mathsf{s}$, then $\pi\, x \in \Sigma_{\mathsf{s}}(\Gamma)$;
- if $op \in \mathsf{Op}_\Sigma$ has type $[\vec{\nu}_1.\,\mathsf{s}_1, \ldots, \vec{\nu}_n.\,\mathsf{s}_n] \to \mathsf{s}$, $t_i \in \Sigma_{\mathsf{s}_i}(\Gamma)$, and $\vec{a}_i \in \mathbb{A}_{\vec{\nu}_i}$ for $1 \le i \le n$ then $op\, \vec{a}_1.\, t_1 \cdots \vec{a}_n.\, t_n \in \Sigma_{\mathsf{s}}(\Gamma)$. We use the abbreviations t for $().\,t$ and $a.\,t$ for $(a).\,t$.

Judgements and theories may then be defined for binding NEL as in the previous section.

Example 5.3 A binding NEL-signature for the untyped λ-calculus [1] may be defined as follows. Fix a single atom sort $\nu \in \mathbb{A}\mathsf{Sort}$ and a single sort tm representing λ-terms. Then the operation symbols are

$$\{\, V_a \mid a \in \mathbb{A}_\nu \} \ \cup \ \{L\} \ \cup \ \{A\}$$

representing variables, λ-abstractions and application respectively. The Perm-action on these operation symbols is $\pi \cdot V_a = V_{\pi(a)}$ and the identity on L and A. The typing function is defined by

$$V_a : [\,] \to \mathsf{tm}, \quad L : [\nu.\,\mathsf{tm}] \to \mathsf{tm}, \quad A : [\mathsf{tm}, \mathsf{tm}] \to \mathsf{tm} \ .$$

The binding NEL-theory for $\alpha\beta\eta$-equivalence on the untyped λ-calculus can then be defined, following [7, Fig. 4], as:

$$a \# x : \mathsf{tm}, x' : \mathsf{tm} \vdash A\,(L\,a.\,x)\,x' \approx x : \mathsf{tm} \qquad (\beta\text{–}1)$$

$$x' : \mathsf{tm} \vdash A\,(L\,a.\,V_a)\,x' \approx x' : \mathsf{tm} \qquad (\beta\text{–}2)$$

$$x : \mathsf{tm}, a' \# x' : \mathsf{tm} \vdash A\,(L\,a.\,(L\,a'\,x))\,x' \approx L\,a'\,(A\,(L\,a.\,x)\,x') : \mathsf{tm} \qquad (\beta\text{–}3)$$

$$x_1 : \mathsf{tm}, x_2 : \mathsf{tm}, x' : \mathsf{tm} \vdash A\,(L\,a.\,(A\,x_1\,x_2))\,x' \approx A\,(A\,(L\,a.\,x_1)\,x')\,(A\,(L\,a.\,x_2)\,x') : \mathsf{tm} \qquad (\beta\text{–}4)$$

$$a' \# x : \mathsf{tm} \vdash A\,(L\,a.\,x)\,V_{a'} \approx (a\ a')\,x : \mathsf{tm} \qquad (\beta\text{–}5)$$

$$a \# x : \mathsf{tm} \vdash x \approx L\,a.\,(A\,x\,V_a) : \mathsf{tm} \qquad (\eta)$$

In [7, Fig. 4] we needed to define an additional axiom, for α-equivalence, as NEL does not offer explicit support for binding. With the binding NEL of this section we have gained the ability to specify binding structure through the typing of operation symbols such as L. Therefore equality in the empty theory over a binding NEL-signature will be interpreted as α-equivalence rather than the literal equality of terms.

Example 5.4 Nominal Substitution [11, Sec. 1.4] is a theory of name-for-name substitution. Fixing ν, tm as above, we have the operation symbols

$$\{sub_a : [\nu.\,\mathsf{tm}] \to \mathsf{tm} \mid a \in \mathbb{A}_\nu\}$$

with Perm-action $\pi \cdot sub_a = sub_{\pi(a)}$. The intended meaning of the term $sub_a\,b.\,t$ is to map b to a in t. The theory of Nominal Substitution is then given by:

$$x : \mathsf{tm} \vdash sub_a\,a.\,x \approx x : \mathsf{tm} \qquad \text{(Identity)}$$

$$a \# x : \mathsf{tm} \vdash sub_b\,a.\,x \approx x : \mathsf{tm} \qquad \text{(Weakening)}$$

$$x : \mathsf{tm} \vdash sub_c\,b.\,sub_b\,a.\,x \approx sub_c\,b.\,sub_c\,a.\,x : \mathsf{tm} \qquad \text{(Contraction-1)}$$

$$x : \mathsf{tm} \vdash sub_a\,b.\,sub_b\,a.\,x \approx sub_a\,b.\,x : \mathsf{tm} \qquad \text{(Contraction-2)}$$

$$x : \mathsf{tm} \vdash sub_d\,b.\,sub_c\,a.\,x \approx sub_c\,a.\,sub_d\,b.\,x : \mathsf{tm} \qquad \text{(Permutation-1)}$$

$$x : \mathsf{tm} \vdash sub_c\,b.\,sub_c\,a.\,x \approx sub_c\,a.\,sub_c\,b.\,x : \mathsf{tm} \qquad \text{(Permutation-2)}$$

Definition 5.5 Given a binding NEL-signature Σ, a Σ-structure is specified by sending sorts s to nominal sets $[\![\mathsf{s}]\!]$ and operation symbols op to finitely supported functions $[\![op]\!]$ as with Def. 4.8, except that if op has type (10) then

$$[\![op]\!] : [\mathbb{A}_{\vec{\nu}_1}][\![\mathsf{s}_1]\!] \times \cdots \times [\mathbb{A}_{\vec{\nu}_n}][\![\mathsf{s}_n]\!] \to [\![\mathsf{s}]\!] \ .$$

Definition 5.6 The value $[\![t]\!]\rho$ of a well-sorted binding NEL term $t \in \Sigma_{\mathsf{s}}(\Gamma)$ with respect to a valuation $\rho \in [\![\Gamma]\!]$ is defined to be an element of $[\![\mathsf{s}]\!]$ by:

$$[\![\pi\, x]\!]\rho \;\triangleq\; \pi \cdot \rho(x)$$
$$[\![op\, \vec{a}_1.\, t_1 \cdots \vec{a}_n.\, t_n]\!]\rho \;\triangleq\; [\![op]\!](\langle\vec{a}_1\rangle([\![t_1]\!]\rho), \ldots, \langle\vec{a}_n\rangle([\![t_n]\!]\rho)) \ .$$

where each $\langle\vec{a}_i\rangle([\![t_i]\!]\rho)$ is an atom abstraction (Def. 3.4).

Satisfaction and algebra may then be defined for binding NEL as in the previous section.

6 Binders do not add expressivity

We now define a translation T from the binding NEL defined in the previous section to the NEL of Sec. 4. We will show that this translation does not sacrifice expressivity.

Definition 6.1 Given a binding NEL-signature Σ, define the NEL-signature $T(\Sigma)$ by

- $\mathsf{Sort}_{T(\Sigma)} \;\triangleq\; \mathsf{Sort}_{\Sigma}$;
- For $\mathsf{Op}_{T(\Sigma)}$, replace each $op : [\vec{\nu}_1.\, \mathsf{s}_1, \ldots, \vec{\nu}_n.\, \mathsf{s}_n] \to \mathsf{s} \in \mathsf{Op}_{\Sigma}$ with the set

  (11) $$\{op_{\vec{a}_1,\ldots,\vec{a}_n} \mid \vec{a}_i \in \mathbb{A}_{\vec{\nu}_i} \text{ for } 1 \leq i \leq n\} \ .$$

  All operation symbols in this set have type $[\mathsf{s}_1, \ldots, \mathsf{s}_n] \to \mathsf{s}$. Defining the Perm-action on $\mathsf{Op}_{T(\Sigma)}$ as $\pi \cdot (op_{\vec{a}_1,\ldots,\vec{a}_n}) \;\triangleq\; (\pi \cdot op)_{\pi\cdot\vec{a}_1,\ldots,\pi\cdot\vec{a}_n}$ ensures that $\mathsf{Op}_{T(\Sigma)}$ is a nominal set and that the typing function is equivariant.

Given a term $t \in \Sigma_{\mathsf{s}}(\Gamma)$, where Σ is a binding NEL-signature, define $T(t) \in T(\Sigma)_{\mathsf{s}}(\Gamma)$ by

$$T(\pi\, x) \;\triangleq\; \pi\, x$$
$$T(op\, \vec{a}_1.\, t_1 \cdots \vec{a}_n.\, t_n) \;\triangleq\; op_{\vec{a}_1,\ldots,\vec{a}_n} T(t_1) \cdots T(t_n) \ .$$

Given a Σ-structure M, where Σ is a binding NEL-signature, define the $T(\Sigma)$-structure $T(M)$ by

- $T(M)[\![\mathsf{s}]\!] \;\triangleq\; M[\![\mathsf{s}]\!]$ for all $\mathsf{s} \in \mathsf{Sort}_{\Sigma} = \mathsf{Sort}_{T(\Sigma)}$;
- Given $op_{\vec{a}_1,\ldots,\vec{a}_n} : [\mathsf{s}_1, \ldots, \mathsf{s}_n] \to \mathsf{s}$ as in (11) and $x_i \in T(M)[\![\mathsf{s}_i]\!] = M[\![\mathsf{s}_i]\!]$ for $1 \leq i \leq n$,

  (12) $$T(M)[\![op_{\vec{a}_1,\ldots,\vec{a}_n}]\!](x_1, \ldots, x_n) \;\triangleq\; M[\![op]\!](\langle\vec{a}_1\rangle x_1, \ldots, \langle\vec{a}_n\rangle x_n) \ .$$

Lemma 6.2 *Take a binding signature Σ, sorting environment Γ, term $t \in \Sigma_{\mathsf{s}}(\Gamma)$, Σ-structure M and valuation $\rho \in M[\![\Gamma]\!] = T(M)[\![\Gamma]\!]$. Then*

$$T(M)[\![T(t)]\!]\rho = M[\![t]\!]\rho \ .$$

Proof. By induction on the structure of t. □

Definition 6.3 Given a theory $\mathbb{T}$ over a binding signature Σ we define a theory $T(\mathbb{T})$ over $T(\Sigma)$ as follows. First we replace each axiom $\nabla \vdash \overline{a} \not\# t \approx t' : \mathsf{s}$ of $\mathbb{T}$ with the axiom

$$\nabla \vdash \overline{a} \not\# T(t) \approx T(t') : \mathsf{s} \tag{13}$$

Then we add a *binding axiom* for each $op_{\vec{a}_1,\ldots,\vec{a}_n} : [\mathsf{s}_1,\ldots,\mathsf{s}_n] \to \mathsf{s} \in \mathsf{Op}_{T(\Sigma)}$, where $\vec{a}_i \in \mathbb{A}_{\vec{\nu}_i}$ for $1 \le i \le n$, as follows. Take distinct lists of atoms $\vec{b}_1,\ldots,\vec{b}_n$, where $\vec{b}_i \in \mathbb{A}^{(*)}_{\vec{\nu}_i}$ and $supp(\vec{b}_i) \# (op, \vec{a}_i)$. Then the binding axiom is

$$\begin{gathered} supp(\vec{b}_1) \not\# x_1 : \mathsf{s}_1, \ldots, supp(\vec{b}_n) \not\# x_n : \mathsf{s}_n \vdash \\ op_{\vec{a}_1,\ldots,\vec{a}_n} x_1 \cdots x_n \approx op_{\vec{b}_1,\ldots,\vec{b}_n} (\vec{a}_1\ \vec{b}_1)\, x_1 \cdots (\vec{a}_n\ \vec{b}_n)\, x_n : \mathsf{s} \ . \end{gathered} \tag{14}$$

(14) may look complicated, but that is largely because it is robust enough to handle any binding structure in the arguments of an operation symbol. In practice binding structures will tend to be simpler, as in the following examples.

Example 6.4 In Ex. 5.3 we defined a binding NEL-signature for the untyped λ-calculus, in particular representing λ-abstractions by the operation symbol $L : [\nu.\,\mathsf{tm}] \to \mathsf{tm}$. Applying the translation T gives us a nominal set of operation symbols $\{L_a \mid a \in \mathbb{A}_\nu\}$, each having type $[\mathsf{tm}] \to \mathsf{tm}$. The binding axiom for each L_a is, following (14),

$$b \not\# x : \mathsf{tm} \vdash L_a\, x \approx L_b\, (a\ b)\, x : \mathsf{tm} \tag{15}$$

where $b \neq a$. In fact by the equivariance of NEL [7, Thm. 8.1] we only need one copy of this axiom as all others could be reached by applying some permutation. Further, (15) is in fact equivalent to

$$x : \mathsf{tm} \vdash a \not\# L_a\, x : \mathsf{tm} \ .$$

This is the axiom that was presented for α-equivalence in [7, Fig. 4].

Example 6.5 The operation symbols of Nominal Substitution (Ex. 5.4) under this translation will be

$$\{sub_{a,b} : [\mathsf{tm}] \to \mathsf{tm} \mid (a,b) \in \mathbb{A}_\nu \times \mathbb{A}_\nu\}$$

with the evident Perm-action, where $sub_{a,b}\, t$ is understood as mapping b to a in t. We have the binding axioms

$$\begin{gathered} c \not\# x : \mathsf{tm} \vdash sub_{a,b}\, x \approx sub_{a,c}\, (b\ c)\, x : \mathsf{tm} \ ; \\ c \not\# x : \mathsf{tm} \vdash sub_{a,a}\, x \approx sub_{a,c}\, (a\ c)\, x : \mathsf{tm} \ . \end{gathered}$$

where a, b, c are disjoint. The first axiom is equivalent to $x \vdash b \not\# sub_{a,b}\, x$ while the second turns out to be a consequence of this and the other axioms presented in Ex. 5.4, so may be omitted from the full theory.

Lemma 6.6 *If $\mathbb{T}$ is a theory over a binding NEL-signature Σ and M is a $\mathbb{T}$-algebra, then $T(M)$ is a $T(\mathbb{T})$-algebra.*

Proof. If $\nabla \vdash \overline{a} \,\#\!\!\!/\, t \approx t' : \mathsf{s}$ is an axiom of $\mathbb{T}$ then (13) is an axiom of $T(\mathbb{T})$. Take $\rho \in T(M)[\![\nabla]\!] = M[\![\nabla]\!]$. $\overline{a} \# M[\![t]\!]\rho = M[\![t']\!]\rho$ because M is a $\mathbb{T}$-algebra, so by Lem. 6.2, $\overline{a} \# T(M)[\![T(t)]\!]\rho = T(M)[\![T(t')]\!]\rho$.

We now turn to the binding axioms (14). Given an appropriate valuation ρ, the value of the left hand side is

$$T(M)[\![op_{\vec{a}_1,\dots,\vec{a}_n}]\!](\rho(x_1),\dots\rho(x_n)) = M[\![op]\!](\langle\vec{a}_1\rangle\rho(x_1),\dots,\langle\vec{a}_n\rangle\rho(x_n))$$

by (9) and (12). The right hand side is

$$M[\![op]\!](\langle\vec{b}_1\rangle((\vec{a}_1\ \vec{b}_1)\cdot\rho(x_1))\dots\langle\vec{b}_n\rangle((\vec{a}_n\ \vec{b}_n)\cdot\rho(x_n)))\ .$$

These are equal by (7). □

So that our translation T is sound and complete, we also need to be able to translate algebras in the other direction, from $T(\mathbb{T})$-algebras M to $\mathbb{T}$-algebras $U(M)$:

Lemma 6.7 *Suppose that Σ is a binding NEL-signature and that M is a $T(\Sigma)$-structure that satisfies all the binding axioms for the translation. Then we can define a Σ-structure $U(M)$ by*

- $U(M)[\![\mathsf{s}]\!] \triangleq M[\![\mathsf{s}]\!]$;
- *Given* $op : [\vec{\nu}_1.\mathsf{s}_1,\dots,\vec{\nu}_n.\mathsf{s}_n] \to \mathsf{s}$, $x_i \in U(M)[\![\mathsf{s}_i]\!] = M[\![\mathsf{s}_i]\!]$ *and* $\vec{a}_i \in \mathbb{A}_{\vec{\nu}_i}$ *for* $1 \le i \le n$,

  (16) $$U(M)[\![op]\!](\langle\vec{a}_1\rangle x_1,\dots,\langle\vec{a}_n\rangle x_n) \triangleq M[\![op_{\vec{a}_1,\dots,\vec{a}_n}]\!](x_1,\dots,x_n)\ .$$

Proof. The step that requires proof is that (16) defines a function; that is, if we have $\vec{a}_i, \vec{a}'_i \in \mathbb{A}_{\vec{\nu}_i}$ and $x_i, x'_i \in M[\![\mathsf{s}_i]\!]$ such that $\langle\vec{a}_i\rangle x_i = \langle\vec{a}'_1\rangle x'_i$ for $1 \le i \le n$ then $M[\![op_{\vec{a}_1,\dots,\vec{a}_n}]\!](x_1,\dots,x_n) = M[\![op_{\vec{a}'_1,\dots,\vec{a}'_n}]\!](x'_1,\dots,x'_n)$. Taking suitably fresh and distinct $\vec{b}_i \in \mathbb{A}^{(*)}_{\vec{\nu}_i}$,

$$M[\![op_{\vec{a}_1,\dots,\vec{a}_n}]\!](x_1,\dots,x_n) = M[\![op_{\vec{b}_1,\dots,\vec{b}_n}]\!]((\vec{a}_1\ \vec{b}_1)\cdot x_1,\dots,(\vec{a}_n\ \vec{b}_n)\cdot x_n)$$

because M satisfies the relevant binding axiom. This equals

$$M[\![op_{\vec{b}_1,\dots,\vec{b}_n}]\!]((\vec{a}'_1\ \vec{b}_1)\cdot x'_1,\dots,(\vec{a}'_n\ \vec{b}_n)\cdot x'_n)$$

because $(\vec{a}_i\ \vec{b}_i)\cdot x = (\vec{a}'_i\ \vec{b}_i)\cdot x'$ for $1 \le i \le n$ by Lem. 3.5. This in turn equals $M[\![op_{\vec{a}'_1,\dots,\vec{a}'_n}]\!](x'_1,\dots,x'_n)$ by satisfaction of another binding axiom. □

Lemma 6.8 *[ref. Lems. 6.2 and 6.6] Take a binding signature Σ. Then*

(i) *Given a sorting environment Γ, term $t \in \Sigma_{\mathsf{s}}(\Gamma)$, $T(\Sigma)$-structure M and valuation $\rho \in M[\![\Gamma]\!]$,*

$$U(M)[\![t]\!]\rho = M[\![T(t)]\!]\rho\ .$$

(ii) *If* $\mathbb{T}$ *is a theory for* Σ *and* M *is a* $T(\mathbb{T})$*-algebra then* $U(M)$ *is a* $\mathbb{T}$*-algebra.*

Proof. (i) follows by induction on the structure of t, while (ii) follows by (i) and the translated axioms (13). $\square$

The final theorem of this section shows that given any theory over a binding NEL-signature, the translation T to a standard NEL-signature maintains expressivity with respect to the semantic consequence relation of Def. 4.11.

Theorem 6.9 *Given any theory* $\mathbb{T}$ *over a binding signature* Σ *and any NEL-judgement* $\nabla \vdash \overline{a} \not\# t \approx t' : \mathsf{s}$ *over* Σ*,*

$$\nabla \vDash_{\mathbb{T}} \overline{a} \not\# t \approx t' : \mathsf{s} \;\Leftrightarrow\; \nabla \vDash_{T(\mathbb{T})} \overline{a} \not\# T(t) \approx T(t') : \mathsf{s} \;.$$

Proof. Right-to-left: Take any $\mathbb{T}$-algebra M and $\rho \in M[\![\nabla]\!]$. $T(M)$ is a $T(\mathbb{T})$-algebra by Lem. 6.6, so $\overline{a} \# T(M)[\![T(t)]\!]\rho = T(M)[\![T(t')]\!]\rho$, which gives our result by Lem. 6.2. The converse holds similarly, by Lem. 6.8. $\square$

7 Nominal Signatures

Any work that attempts to capture the notion of binding has to confront the fact that many different definitions of binding exist. Many of the more elaborate notions, such as the binding specifications of Cαml [19], seek to extend practical, rather than theoretical, expressivity so can be translated into NEL-signatures as in the previous section at the cost of a loss of elegance. However it is an interesting open question to what extent other elaborate binders can be accommodated; see [5] for discussion.

For now we will restrict our attentions to a notion of binding structure that has been extensively used in the nominal sets literature, that of *nominal signatures* [20]. An analogous result to that of this section has been independently developed for Nominal Algebra using a variation of nominal signatures that is unsorted [12, Sec. 5.1]. As we will see, utilising a sorting system introduces new subtleties into the proof.

Nominal signatures encode binding through their sort structure and use explicit atom abstraction terms. The sorts of a nominal signature start with a set of base sorts with typical member b, and are defined by

$$\mathsf{s} \;::=\; \mathsf{b} \mid 1 \mid \mathsf{s} \times \mathsf{s} \mid \nu \mid \nu.\mathsf{s} \tag{17}$$

for all $\nu \in \mathbb{A}\mathsf{Sort}$. A nominal signature specifies the base sorts and a set of operation symbols, each with type $\mathsf{s} \to \mathsf{b}$, where s is as in (17). The terms then have the following form and sort:

- *Unit:* $() : 1$;
- *Pair:* $(t_1, t_2) : \mathsf{s}_1 \times \mathsf{s}_2$ if $t_1 : \mathsf{s}_1$ and $t_2 : \mathsf{s}_2$;
- *Constructed term:* $op\, t : \mathsf{b}$ if $op : \mathsf{s} \to \mathsf{b}$ and $t : \mathsf{s}$;
- *Atom:* $a : \nu$ for $a \in \mathbb{A}_\nu$;
- *Atom abstraction:* $a.t : \nu.\mathsf{s}$ if $a \in \mathbb{A}_\nu$ and $t : \mathsf{s}$;

- *Suspension:* $\pi\, x : \mathsf{s}$ if $\pi \in \mathsf{Perm}$ and x has base sort or atom sort s according to some sorting environment.

Nominal signatures may include variables of atom sort, but while such variables may be convenient they do not add expressivity: a judgement whose freshness environment contains $\overline{a} \mathrel{\#} x : \nu$ can be replaced in a theory by judgements without that assertion in the freshness environment, applying the substitution $\{a/x\}$ to the judgement's terms for all $a \in \mathbb{A} - \overline{a}$. Therefore we will only consider variables of base sort b.

An algebra for such a nominal signature involves first assigning a nominal set $[\![\mathsf{b}]\!]$ to each base sort b, allowing us to define $[\![\mathsf{s}]\!]$ for each sort s:

$$(18) \qquad [\![1]\!] \triangleq \{*\}, \quad [\![\mathsf{s} \times \mathsf{s}']\!] \triangleq [\![\mathsf{s}]\!] \times [\![\mathsf{s}']\!], \quad [\![\nu]\!] \triangleq \mathbb{A}_\nu, \quad [\![\nu.\,\mathsf{s}]\!] \triangleq [\mathbb{A}_\nu][\![\mathsf{s}]\!] \ .$$

Next we assign an equivariant function $[\![op]\!] : [\![\mathsf{s}]\!] \to [\![\mathsf{b}]\!]$ to each $op : \mathsf{s} \to \mathsf{b}$. Given a valuation ρ of variables of sort b to members of $[\![\mathsf{b}]\!]$ we define the values of terms by

$$[\![()]\!]\rho \triangleq *, \quad [\![(t_1, t_2)]\!]\rho \triangleq ([\![t_1]\!]\rho, [\![t_2]\!]\rho), \quad [\![op\, t]\!]\rho \triangleq [\![op]\!]([\![t]\!]\rho),$$
$$[\![a.\, t]\!]\rho \triangleq \langle a \rangle([\![t]\!]\rho), \quad [\![\pi\, x]\!]\rho \triangleq \pi \cdot (\rho(x)) \ .$$

We may now follow Sec. 6 and define a translation T from this logic to NEL. Given a nominal signature Σ we define the NEL-signature $T(\Sigma)$ as follows: set $\mathsf{Sort}_{T(\Sigma)}$ to be the sorts enumerated by (17) and $\mathsf{Op}_{T(\Sigma)}$ to be the set of operation symbols of Σ along with the new symbols

- $unit : [\,] \to 1$;
- $pair : [\mathsf{s}, \mathsf{s}'] \to \mathsf{s} \times \mathsf{s}'$ for all sorts s, s';
- $\{atm_a \mid a \in \mathbb{A}_\nu\} : [\,] \to \nu$ for all atom sorts ν;
- $\{abs_a \mid a \in \mathbb{A}_\nu\} : [\mathsf{s}] \to \nu.\,\mathsf{s}$ for all sorts s and atom sorts ν

with the evident Perm-actions. Hence given any Σ-term t we produce the $T(\Sigma)$-term $T(t)$ by the obvious substitutions, replacing each $()$ by *unit* and so forth.

Given an Σ-structure M for a nominal signature Σ we may define a $T(\Sigma)$-structure $T(M)$ by following (18) for each $T(M)[\![\mathsf{s}]\!]$ and setting

$$T(M)[\![op]\!] \triangleq M[\![op]\!], \quad T(M)[\![unit]\!] \triangleq *, \quad T(M)[\![pair]\!](x_1, x_2) \triangleq (x_1, x_2),$$
$$T(M)[\![atm_a]\!] \triangleq a, \quad T(M)[\![abs_a]\!](x) \triangleq \langle a \rangle x \ .$$

Now given a Σ-theory $\mathbb{T}$ for a nominal signature Σ we define the $T(\Sigma)$-theory $T(\mathbb{T})$ by replacing each axiom $\nabla \vdash \overline{a} \mathrel{\#} t \approx t' : \mathsf{s}$ by $\nabla \vdash \overline{a} \mathrel{\#} T(t) \approx T(t') : \mathsf{s}$ and adding the binding axioms

$$(19) \qquad x : \mathsf{s} \vdash a \mathrel{\#} abs_a\, x : \nu.\,\mathsf{s}$$

for every sort s, atom sort ν and some $a \in \mathbb{A}_\nu$. Analogues of Lems. 6.2 and 6.6 for this translation T follow.

Still following Sec. 6, we define a translation U from $T(\Sigma)$-structures M to Σ-structures by first setting $U(M)[\![\mathsf{b}]\!] \triangleq M[\![\mathsf{b}]\!]$. We then define a family of equivariant functions $f_{\mathsf{s}} : U(M)[\![\mathsf{s}]\!] \to M[\![\mathsf{s}]\!]$ as s ranges over the sorts (17) as follows:

- f_{b} is the identity on $U(M)[\![\mathsf{b}]\!] = M[\![\mathsf{b}]\!]$;
- $f_1(*) = M[\![unit]\!]$;
- $f_{\mathsf{s}_1\times\mathsf{s}_2}(x_1, x_2) = M[\![pair]\!](f_{\mathsf{s}_1}(x_1), f_{\mathsf{s}_2}(x_2))$ for all $x_i \in U(M)[\![\mathsf{s}_i]\!]$ and $i = 1, 2$;
- $f_\nu(a) = M[\![atm_a]\!]$ for all $a \in \mathbb{A}_\nu$;
- $f_{\nu.\mathsf{s}}(\langle a\rangle x) = M[\![abs_a]\!](f_{\mathsf{s}}(x))$ for all $x \in U(M)[\![\mathsf{s}]\!]$ and $a \in \mathbb{A}_\nu$.

To see that $f_{\nu.\mathsf{s}}$ is a well-defined function requires the binding axiom (19). Using all this we may now define the structure $U(M)$ on operation symbols by

$$U(M)[\![op]\!] \;\triangleq\; M[\![op]\!] \circ f_{\mathsf{s}}$$

for all $op : \mathsf{s} \to \mathsf{b}$ of Σ. We can then show by a routine induction that

$$M[\![T(t)]\!]\rho = f_{\mathsf{s}}(U(M)[\![t]\!]\rho)$$

where t has sort s according to some freshness environment and ρ is a valuation for that environment. In the case where $\mathsf{s} = \mathsf{b}$, f_{b} is the identity, so $M[\![T(t)]\!]\rho = U(M)[\![t]\!]\rho$, following Lem. 6.8. Therefore if M is a $T(\mathbb{T})$-algebra then $U(M)$ is a $\mathbb{T}$-algebra *so long as all the axioms of* $\mathbb{T}$ *have base sort*. We can then prove a version of Thm. 6.9 provided again that the sorts of the axioms and the judgement in question are of base sort b.

This is reasonable restriction on the class of judgements over a nominal signature that we wish to consider, as it is in keeping with the restrictions made in [20] on the sorts that variables may take and the result sorts of operation symbols. The intuition justifying all these restrictions is that the base sorts represent the sorts that terms may take in some formal system that we are defining with our theory, while the other sorts are used only in constructing a term. Allowing theories to include axioms that use these non-base sorts is not in keeping with this intuition, and also allows the definition of inconsistent judgements like $\vdash a \not\# a : \nu$. Therefore we may reasonably consider the translation T sound and complete.

8 Further work – Characterising atom abstractions

The previous sections have shown that NEL is expressive enough to encode the binding structure of operators. However, we might ask the related but stronger question of whether we can characterise atom abstractions with a NEL-theory, in the sense of the well known result that products may be characterised by an equational theory (see e.g. [9]). In fact this characterisation is not quite possible, and this section will discuss why this is, how close we can come to this result and how in future work we might extend NEL to have this expressivity.

Just as the equational theory for products requires operation symbols for products and projection, we will need some way to deconstruct our atom abstractions.

The appropriate such notion is *concretion* [13, Def. 5.3]:

Definition 8.1 *Concretion* is an equivariant function $[\mathbb{A}_\nu]X \otimes \mathbb{A}_\nu \to X$ whose action on (y, b) is written $y \mathbin{@} b$. In other words, by Ex. 2.6 and Lem. 3.3, $\langle a \rangle x \mathbin{@} b$ is defined if either $b = a$ or $b \# x$. Concretion is defined by

$$\langle a \rangle x \mathbin{@} b \;\triangleq\; (a\ b) \cdot x \ .$$

In particular, in the case that $b = a$, $\langle a \rangle x \mathbin{@} a = x$.

Now suppose we wished to define a NEL-theory for atom-abstraction and concretion over some sort s. We would introduce new sorts $\nu.\mathsf{s}$ for all $\nu \in \mathbb{A}\mathsf{Sort}$, along with operation symbols representing atom abstraction, $abs_a : [\mathsf{s}] \to \nu.\mathsf{s}$ for all $a \in \mathbb{A}_\nu$. To then define operation symbols for concretion likewise, $con_a : [\nu.\mathsf{s}] \to \mathsf{s}$, would involve interpreting con_a as a total function $[\![\nu.\mathsf{s}]\!] \to [\![\mathsf{s}]\!]$, where in fact it should only be partial; we should only have the term $con_a\, x$ when we can guarantee that $a \# x$. NEL, like standard equational logic, offers no support for such partiality.

However concretion can be extended to a total function for a surprisingly large range of nominal sets. For example, the nominal set $\mathbb{A}$ has such a 'total concretion' defined on it by

$$\begin{aligned} \langle b \rangle b \mathbin{@} a &\;\triangleq\; a \ ; \\ b \neq c \Rightarrow \langle b \rangle c \mathbin{@} a &\;\triangleq\; c \ . \end{aligned} \tag{20}$$

The second equation includes the case that $c = a$, despite the fact that $a \in supp(\langle b \rangle a) = \{a\}$. Similarly, concretion can be defined as a total function on any nominal set equipped with a *restriction structure* [18, Thm. 22] (this does not subsume (20) as $\mathbb{A}$ cannot have a restriction structure defined upon it).

If we restrict our attention to those nominal sets for which concretion can be defined as total we can indeed characterise atom abstraction with a NEL-theory:

$$\begin{aligned} x : \mathsf{s} &\vdash a \#\!\!\# \; abs_a\, x : \nu.\mathsf{s} \\ x : \mathsf{s} &\vdash con_a\, abs_a\, x \approx x : \mathsf{s} \\ b \#\!\!\# \; x : \mathsf{s} &\vdash con_b\, abs_a\, x \approx (a\ b)\, x : \mathsf{s} \\ a \#\!\!\# \; x : \nu.\mathsf{s} &\vdash abs_a\, con_a\, x \approx x : \nu.\mathsf{s} \end{aligned} \tag{21}$$

Note that the first axiom listed is this theory's binding axiom (Def. 6.3).

However such total concretions cannot be defined on nominal sets in general. Consider the nominal set $\mathbb{A} \otimes \mathbb{A}$, and the pair that would be defined by

$$\langle a \rangle (a, b) \mathbin{@} b \tag{22}$$

where $a \neq b$ by definition. $\langle a \rangle (a, b)$ has support $\{b\}$ by Lem. 3.3, so because concretion is equivariant, (22) must have support contained in $\{b\}$ by (5). But all elements of $\mathbb{A} \otimes \mathbb{A}$ have two atoms in their support sets.

A number of ways present themselves to extend NEL so that (21) may characterise atom abstraction over any nominal set. For example, we could introduce partiality directly [3], or we could carry around freshness information in the sort structure and only allow concretion to be defined where our sort guarantees the correct freshness. The latter approach would require the extension to NEL outlined in [6], where the collection of sorts may form a nominal set, along with an ordering on the sorts [14] so that if we have finite sets of atoms $\overline{a}$, $\overline{a}'$ such that $\overline{a} \subseteq \overline{a}'$ then a sort with $\overline{a}$ guaranteed to be fresh for it will be a subsort of that sort with $\overline{a}'$ guaranteed fresh.

References

[1] Barendregt, H. P., "The Lambda Calculus: Its Syntax and Semantics," North-Holland, Amsterdam, 1984.

[2] Berghofer, S. and C. Urban, *A head-to-head comparison of de Bruijn indices and names*, Electronic Notes in Theoretical Computer Science **174** (2007), pp. 53–67.

[3] Burmeister, P., *Partial algebras – an introductory survey*, in: I. Rosenberg and G. Sabidussi, editors, *Algebras and Orders (Proc. of the Summer School at Montreal)*, Kluver Publ. Co, Dordrecht, 1992 pp. 1–70.

[4] Cheney, J., *Logic column 14; nominal logic and abstract syntax*, SIGACT News **36** (2005), pp. 47–69.

[5] Cheney, J., *Towards a general theory of names, binding and scope*, in: *MERLIN* (2005), pp. 33–40.

[6] Clouston, R., "Equational logic for names and binders," Ph.D. thesis, University of Cambridge (2009).

[7] Clouston, R. A. and A. M. Pitts, *Nominal equational logic*, Electronic Notes in Theoretical Computer Science **172** (2007), pp. 223–257.

[8] de Bruijn, N. G., *Lambda calculus notation with nameless dummies, a tool for automatic formula manipulation, with application to the Church-Rosser theorem*, Indagationes Mathematicae **34** (1972), pp. 381–392.

[9] de Vrijer, R. C., *Extending the lambda calculus with surjective pairing is conservative*, in: *LICS*, 1989, pp. 204–215.

[10] Fiore, M. P., G. D. Plotkin and D. Turi, *Abstract syntax and variable binding*, in: *14th Annual Symposium on Logic in Computer Science* (1999), pp. 193–202.

[11] Fiore, M. P. and S.Staton, *A congruence rule format for name-passing process calculi from mathematical structural operational semantics*, in: *Proceedings of the 21st Annual IEEE Symposium on Logic in Computer Science (LICS'06)* (2006), pp. 49–58.

[12] Gabbay, M. J. and A. Mathijssen, *Nominal (universal) algebra: equational logic with names and binding*, Journal of Logic and Computation (2009), in press.

[13] Gabbay, M. J. and A. M. Pitts, *A new approach to abstract syntax with variable binding*, Formal Aspects of Computing **13** (2002), pp. 341–363.

[14] Goguen, J. and R. Diaconescu, *An Oxford survey of order sorted algebra*, in: *Mathematical Structures in Computer Science*, 1994, pp. 363–392.

[15] Gunter, C. A., "Semantics of Programming Languages: Structures and Techniques," Foundations of Computing, MIT Press, Cambridge, MA, 1992.

[16] Pfenning, F. and C. Elliot, *Higher-order abstract syntax*, in: *Proceedings of ACM-SIGPLAN Conference on Programming Language Design and Implementation* (1988), pp. 199–208.

[17] Pitts, A. M., *Nominal logic: A first order theory of names and binding*, in: N. Kobayashi and B. C. Pierce, editors, *Theoretical Aspects of Computer Software, 4th International Symposium, TACS 2001, Sendai, Japan, October 29-31, 2001. Proceedings*, Lecture Notes in Computer Science **2215** (2001), pp. 219–242.

[18] Pitts, A. M., *Nominal System T*, in: *Proceedings of the 37th ACM SIGACT-SIGPLAN Symposium on Principles of Programming Languages (POPL 2010), Madrid, Spain* (2010), pp. 159–170.

[19] Pottier, F., *An overview of Cαml*, in: *ACM SIGPLAN Workshop on ML (ML 2005), Tallinn, Estonia*, Electronic Notes in Theoretical Computer Science (2005), pp. 27–52.

[20] Urban, C., A. M. Pitts and M. J. Gabbay, *Nominal unification*, in: M. Baaz, editor, *Computer Science Logic and 8th Kurt Gödel Colloquium (CSL'03 & KGC), Vienna, Austria. Proceedings*, Lecture Notes in Computer Science **2803** (2003), pp. 513–527.