

Nominal Equational Logic

Ranald A. Clouston¹ Andrew M. Pitts²

Computer Laboratory, University of Cambridge, Cambridge CB3 0DF, UK

Abstract

This paper studies the notion of “freshness” that often occurs in the meta-theory of computer science languages involving various kinds of names. Nominal Equational Logic is an extension of ordinary equational logic with assertions about the freshness of names. It is shown to be both sound and complete for the support interpretation of freshness and equality provided by the Gabbay-Pitts nominal sets model of names, binding and α -conversion.

Keywords: Equational logic, nominal sets, permutation actions, universal algebra.

This paper is dedicated to Gordon Plotkin, whose enormous contributions to the mathematical foundations of computer science are an inspiration.

1 Introduction

Language constructs involving names are a major concern in computer science—much more so than in related disciplines that also use formal languages, such as mathematics and logic. For example, witness the issues surrounding substitution of expressions for identifiers, the sharing of structures through aliasing, and local scoping of definitions—all of which involve properties of names. In this paper we focus on the property of “freshness” of names and present an extension of equational logic that takes it into account. Figure 1 gives three examples of increasing subtlety to illustrate what we mean by freshness.

The first example is drawn from the π -calculus [21]. The notion of freshness here is “ $a \notin fn(Q)$ ”, meaning that the channel name a does not occur free in the process expression Q . Since Q is just a particular kind of finite tree and its set of free names $fn(Q)$ is defined by recursion over the tree’s structure, this notion of freshness is very straightforward.

¹ Email: Ranald.Clouston@cl.cam.ac.uk

² Email: Andrew.Pitts@cl.cam.ac.uk

Scope extrusion in the π -calculus [21]:

$$(\nu a P)|Q = \nu a (P|Q) \quad \text{if } a \notin \text{fn}(Q).$$

Capture-avoiding simultaneous substitution [32]:

$$(\lambda a. M)[\sigma] = \lambda a. M[\sigma] \quad \text{if } a \notin \bigcup \{\{b\} \cup \text{fv}(\sigma b) \mid \sigma b \neq b\}.$$

Normalisation-by-evaluation [3]:

$$\downarrow_{\tau \rightarrow \tau'}(f) = \lambda a. \downarrow_{\tau'}(f(\uparrow_{\tau} a)) \quad \text{if “}a \text{ is fresh for the} \\ \text{function } f \in \llbracket \tau \rrbracket \rightarrow \llbracket \tau' \rrbracket \text{.” (??)}$$

Fig. 1. Three Examples of Freshness in the Wild

The second example is a property of simultaneous substitution for λ -terms (see [32], for example). The freshness condition “ $a \notin \bigcup \{\{b\} \cup \text{fv}(\sigma b) \mid \sigma b \neq b\}$ ” (where $\text{fv}(-)$ returns the finite set of free variables of a λ -term) ensures the binder $\lambda a. (-)$ does not capture free variables in the substitution σ . Here σ is not a finite tree, but rather an infinite mathematical object—namely a function from the countably infinite set of variables $\{a, b, \dots\}$ to the set of λ -terms [2]. However, we impose a finiteness condition on substitutions, namely that $\sigma b \neq b$ only holds for finitely many variables b . Consequently $\bigcup \{\{b\} \cup \text{fv}(\sigma b) \mid \sigma b \neq b\}$ is just a finite set of variables and the notion of freshness in this example is not much more complex than in the first example.

The third example is a property of the reification (\downarrow_{τ}) and reflection (\uparrow_{τ}) functions used to compute $\beta\eta$ -long normal forms of simply typed λ -terms via a functional semantics [3]. Here the semantics $\llbracket \tau \rrbracket$ of a simple type is an infinite set of objects, defined by recursion on the structure of the type expression τ . For example the semantics of a function type $\tau \rightarrow \tau'$ is a set $\llbracket \tau \rrbracket \rightarrow \llbracket \tau' \rrbracket$ of functions from $\llbracket \tau \rrbracket$ to $\llbracket \tau' \rrbracket$. Reification produces typed λ -terms from elements of the semantics; whereas reflection maps typed λ -terms back to semantic elements. The formula for $\downarrow_{\tau \rightarrow \tau'}(f)$ given in the figure only makes sense if the variable a is chosen to be “fresh” for the mathematical function f . Since f may well involve all variables in its graph, it is not at all clear what this should mean. Several mechanisms have been proposed to explain precisely what is meant by this problematic freshness condition—see [7] and [25, Section 6].

In fact all three examples given in the figure are instances of the mathematical notion of freshness provided by the *nominal sets* model of names. This was introduced by Gabbay and Pitts [17]³ and has subsequently been developed and

³ There it was called the “FM-sets” model because the presentation was phrased in terms of the classic

applied in a number of ways: see [30,24,13,14,1,34,6,5,8,29,10,22,33] for example. The effectiveness of nominal sets rests upon two observations. First, properties of names to do with freshness, binding and α -conversion can all be expressed in terms of the primitive operation of swapping names; and secondly, this operation of name-swapping makes sense (and has very convenient properties) not only for finite syntactic objects, but also for infinite mathematical objects, like sets and functions. In this setting the fundamental notion is *support*: One says that a finite set of names supports an object x if x is invariant under swapping any pair of names not in the set. If there is such a finite set of names, then it turns out that there is a smallest such, called the *support* of x . For such x , it makes good sense to say that a is *fresh* for x if a is not in its support. It does make good sense because this language-independent relation has useful properties and coincides with ad hoc notions of freshness in particular cases, such as those in Figure 1. For a recent account of nominal sets, see [25] (section 6 of which deals with the third, normalisation-by-evaluation example in Figure 1).

Writing $a \# x$ to indicate that a name a is not in the support of a finitely supported object x , note that all three of the examples in Figure 1 take the form of equations conditioned by freshness assumptions:

$$\begin{aligned} a \# Q &\Rightarrow (\nu a P)|Q = \nu a (P|Q) \\ a \# \sigma &\Rightarrow (\lambda a. M)[\sigma] = \lambda a. M[\sigma] \\ a \# f &\Rightarrow \downarrow_{\tau \rightarrow \tau'} (f) = \lambda a. \downarrow_{\tau'} (f(\uparrow_{\tau} a)) . \end{aligned}$$

It seems that many properties of names can be axiomatised using such conditional equations: The work of Gabbay and Mathijssen gives several interesting examples [16,15]. However, as well as equations, assertions about freshness also arise naturally, sometimes with freshness conditions, such as

$$b \# Q \Rightarrow b \# \nu a Q$$

and sometimes unconditionally, such as

$$\Rightarrow a \# \nu a Q .$$

So in this paper we study the properties of “equations and freshnesses conditioned by finitely many (possibly zero) freshness assumptions”:

$$a_1 \# x_1 \wedge \cdots \wedge a_n \# x_n \Rightarrow t = t' \tag{1}$$

$$a_1 \# x_1 \wedge \cdots \wedge a_n \# x_n \Rightarrow a \# t . \tag{2}$$

We use a simple extension of the usual language of algebraic terms t that adds names and the kind of explicit name-permutations introduced in [34]. The language has a natural interpretation in nominal sets. The main contribution of this paper is to extend the usual (many-sorted) equational logic to a logic for deriving judgements

Fraenkel-Mostowski permutation model of set theory with atoms.

of the form (1) and (2). We call it *nominal equational logic* and we prove it is both sound and complete for the intended interpretation of the judgements in nominal sets, where freshness means “not in the support of”.

Contents of the paper

In Section 2 we briefly recall the facts that we need about nominal sets. Sections 3–5 describe the algebraic language we use and its interpretation in nominal sets. Sections 6 and 7 introduce the notion of a theory in nominal equational logic (NEL) and its algebras in nominal sets; we give a sound axiomatisation of satisfaction of judgements in an algebra (Theorem 7.4). Section 8 develops some consequences of our formulation of NEL to do with invariance under permuting names. Section 9 describes a term-algebra construction using ground terms (that is, ones with no variables); and this is used in Section 10 to prove that NEL is complete for its intended interpretation in nominal sets (Theorem 10.10). This completeness result is harder to establish than is the case for ordinary equational logic, because the relationship between variables and indeterminates (new constants) is more subtle for NEL. Variables in our setting stand for elements of nominal sets that may depend, via the notion of support, on names; thus the dependency of a variable x on names is *implicit*. Whereas a constant in NEL stands for a fixed element of a nominal set and has an explicitly given support. To prove the completeness theorem we show that the validity of judgements involving variables can be reduced to the validity of ones involving ground terms, via the substitution for variables of constants with suitably fresh supports. Such a reduction was sketched by Gabbay in connection with his “fresh logic” [13, Theorem 9.3]. For NEL we found that the main technical result needed for the reduction (Proposition 10.4) depends quite delicately upon the formulation of the language of terms and the NEL rules for freshness; we prove it via a non-trivial operation on terms for replacing constants by variables (see Figure 6). Finally, Section 11 discusses related work and draws some conclusions.

Acknowledgements

We gratefully acknowledge the work of Murdoch Gabbay and Aad Mathijssen on *nominal algebra* [16,15], which has influenced the results presented here; we are also grateful for the comments of members of the CANS project in Cambridge and King’s College, London. This research was supported by the Woolf Fisher Trust (Clouston) and UK EPSRC grant EP/D000459/1 (Pitts).

2 Atoms, Permutations and Nominal Sets

In the Introduction we discussed some aspects of computer science languages involving *names*. From now on, in keeping with the origins of nominal sets in models of Zermelo-Fraenkel set theory with atoms, we will use the elements of a fixed set **Atom** as our names and refer to them as *atoms*. We assume **Atom** is countably infinite and that it is partitioned into countably infinitely many different *sorts of atom*: There

is a countably infinite set AtomSort and a function $\text{sort} : \text{Atom} \rightarrow \text{AtomSort}$ with the property that for each sort of atom $\alpha \in \text{AtomSort}$, the following set is countably infinite.

$$\text{Atom}_\alpha \triangleq \{a \in \text{Atom} \mid \text{sort}(a) = \alpha\} . \quad (3)$$

The set Perm of (finite, sort-respecting) *permutations* of atoms consists of all bijections $\pi : \text{Atom} \rightarrow \text{Atom}$ such that

$$\text{dom}(\pi) \triangleq \{a \in \text{Atom} \mid \pi(a) \neq a\} \quad (4)$$

is finite and $\text{sort}(\pi(a)) = \text{sort}(a)$ for all $a \in \text{Atom}$. We give Perm the structure of a group by taking the group multiplication to be composition of bijections: If $\pi, \pi' \in \text{Perm}$, then their *composition* $\pi'\pi$, mapping $a \in \text{Atom}$ to $\pi'(\pi(a))$, is again in Perm . The group unit is given by the identity function on Atom , written ι ; and the inverse of $\pi \in \text{Perm}$ is the bijection π^{-1} mapping a to a' if $\pi(a') = a$. We take for granted the fact that Perm is generated by *transpositions* $(a \ a')$ (where a and a' are atoms of the same sort) mapping a to a' , a' to a and leaving all other atoms fixed.

As usual, an *action* of Perm on a set X is a function $(\pi, x) \mapsto \pi \cdot x$ from $\text{Perm} \times X$ to X satisfying:

$$\iota \cdot x = x \quad (5)$$

$$\pi' \cdot (\pi \cdot x) = \pi' \pi \cdot x . \quad (6)$$

Given such an action and an element $x \in X$, we say that a finite subset $\bar{a} \subseteq \text{Atom}$ *supports* x if for all atoms a, a' of the same sort

$$a, a' \notin \bar{a} \Rightarrow (a \ a') \cdot x = x . \quad (7)$$

Then a *nominal set* is simply a set X equipped with an action of Perm such that for each $x \in X$ there exists some finite subset $\bar{a} \subseteq \text{Atom}$ supporting x .

Definition 2.1 (*Freshness Relation*) Given a nominal set X , if $a \in \text{Atom}$ and $x \in X$, we write $a \# x$ and say a is *fresh for* x if there is some finite subset $\bar{a} \subseteq \text{Atom}$ supporting x with $a \notin \bar{a}$. More generally, if \bar{a} is a finite set of atoms we write

$$\bar{a} \# x \quad (8)$$

to mean that $a \# x$ holds for each $a \in \bar{a}$. In fact (8) is equivalent to saying that \bar{a} is disjoint from some single finite set of atoms supporting x . This is because support sets are closed under intersection: see [25, Section 3.1]. For this reason we have the following fundamental property of the freshness relation.

Lemma 2.2 *Suppose x is an element of a nominal set X . If a and a' are atoms (of the same sort) satisfying $a \# x$ and $a' \# x$, then $(a \ a') \cdot x = x$. \square*

We make nominal sets into a category, called \mathcal{Nom} , by taking morphisms $f : X \rightarrow Y$ to be *equivariant* functions, that is, functions $f \in Y^X$ satisfying

$$\pi \cdot (f x) = f(\pi \cdot x) \quad (9)$$

for all $\pi \in \text{Perm}$ and $x \in X$. Composition and identities in \mathcal{Nom} are as in the category of sets and functions. Properties of this category are developed in [11,17,4,25]. In the rest of this section we recall those that we need in this paper.

Definition 2.3 (*Nominal Sets of Atoms*) Each set Atom_α of atoms of a particular sort α is a nominal set once we give it the action:

$$\pi \cdot a \triangleq \pi(a) . \quad (10)$$

The freshness relation for this nominal set turns out to be inequality: $a \# a'$ iff $a \neq a'$.

Definition 2.4 (*Nominal Set of Finite Sets of Atoms*) The set $\mathcal{P}_{\text{fin}}(\text{Atom})$ of finite subsets $\bar{a} \subseteq \text{Atom}$ is a nominal set once we give it the action:

$$\pi \cdot \bar{a} \triangleq \{\pi(a) \mid a \in \bar{a}\} . \quad (11)$$

The freshness relation for this nominal set turns out to be: $a \# \bar{a}$ iff $a \notin \bar{a}$.

Definition 2.5 (*Nominal Set of Permutations*) In this paper we will need to consider two different actions of Perm on itself:

$$\text{left multiplication: } (\pi, \pi') \mapsto \pi\pi' \quad (12)$$

$$\text{conjugation: } (\pi, \pi') \mapsto \pi\pi'\pi^{-1} . \quad (13)$$

Note that for any pair of distinct atoms a and a' of the same sort it is the case that $(a \ a') \neq \iota$. Therefore $(a \ a')\pi \neq \pi$, for any $\pi \in \text{Perm}$. Consequently no permutation π has a finite support set with respect to the left multiplication action; so Perm is not a nominal set with respect to this action. However, it is a nominal set with respect to the conjugation action, since it is not hard to see that the finite set of atoms $\text{dom}(\pi)$, defined in (4), supports π with respect to this action. Indeed $\text{dom}(\pi)$ is the smallest support set for π and so in this nominal set we have $a \# \pi$ iff $\pi(a) = a$.

Lemma 2.6 (Finite Products of Nominal Sets) *The terminal object in \mathcal{Nom} is given by a one-element set, $1 = \{()\}$ say, with the unique permutation action. In this case $a \# ()$ holds for all a . The categorical product of nominal sets X and Y is given by their Cartesian product $X \times Y = \{(x, y) \mid x \in X \wedge y \in Y\}$ with permutation action:*

$$\pi \cdot (x, y) \triangleq (\pi \cdot x, \pi \cdot y) . \quad (14)$$

In this case one can calculate that $a \# (x, y)$ iff $a \# x$ and $a \# y$.

Proof. See for example [25, Section 3.2]. □

Lemma 2.7 (Exponentials of Nominal Sets) *The category \mathcal{Nom} is Cartesian closed. Given nominal sets X and Y , the exponential $X \rightarrow_{\text{fs}} Y$ has underlying set given by the set of functions f from X to Y that are finitely supported with*

respect to the permutation action given by

$$(\pi \cdot f)(x) \triangleq \pi \cdot (f(\pi^{-1} \cdot x)) \quad (\pi \in \text{Perm}, f \in Y^X, x \in X). \quad (15)$$

The evaluation morphism $ev : (X \rightarrow_{\text{fs}} Y) \times X \rightarrow Y$ is given by function application

$$ev(f, x) = f(x) \quad (16)$$

which is indeed equivariant

$$\pi \cdot (f(x)) = (\pi \cdot f)(\pi \cdot x) \quad (17)$$

because of (15). Given a morphism $f : Z \times X \rightarrow Y$, the unique morphism $\bar{f} : Z \rightarrow (X \rightarrow_{\text{fs}} Y)$ satisfying $ev \circ (\bar{f} \times id) = f$ is given by Currying:

$$\bar{f}(z)(x) = f(z, x) \quad (z \in Z, x \in X).$$

Proof. See for example [25, Section 3.2]. □

Remark 2.8 (*Global Elements of Nominal Sets*) It is worth remarking that although \mathcal{Nom} is very rich in structure,⁴ unlike the category of sets it is not *well-pointed*. In other words, a pair of morphisms $f, g : X \rightarrow Y$ may well be unequal even though they have equal compositions with all global elements of X , that is, with all morphisms $1 \rightarrow X$. This is because morphisms $1 \rightarrow X$ in \mathcal{Nom} correspond not to arbitrary elements $x \in X$, but to ones that are supported by the empty set of atoms. To see this, first note that equivariant functions $f : 1 \rightarrow X$ correspond to elements $x = f() \in X$ satisfying $\pi \cdot x = x$ for all $\pi \in \text{Perm}$. Since Perm is generated as a group by the transpositions, this is equivalent to requiring $(a a') \cdot x = x$, for all atoms a, a' (of equal sort); and by definition of support sets, this is equivalent to saying that \emptyset supports x .

In particular, the elements of the exponential $X \rightarrow_{\text{fs}} Y$ with empty support correspond to global elements; and as for any Cartesian closed category, these in turn correspond to morphisms $X \rightarrow Y$ in \mathcal{Nom} . More concretely, this amounts to the easily verified fact that a function $f \in Y^X$ is equivariant (9) if and only if it has empty support with respect to the permutation action given by (15).

Lemma 2.9 (Finite Coproducts of Nominal Sets) *The initial object in \mathcal{Nom} is given by the empty set, \emptyset , with the unique permutation action. The coproduct of nominal sets X and Y is given by their disjoint union $X + Y = \{(0, x) \mid x \in X\} \cup \{(1, y) \mid y \in Y\}$ with permutation action:*

$$\pi \cdot (0, x) \triangleq (0, \pi \cdot x) \quad \pi \cdot (1, y) \triangleq (1, \pi \cdot y). \quad (18)$$

In this case one can calculate that $a \# (0, x)$ iff $a \# x$ in X and that $a \# (1, y)$ iff $a \# y$ in Y . □

⁴ \mathcal{Nom} is an atomic topos, being equivalent to a topos of continuous G -sets [19, III.9] for a suitable choice of topology on $G = \text{Perm}$.

3 Signatures and Structures

We are going to consider a simple generalisation of the usual notion of many-sorted algebraic signature [20, Sec. 3.1] in which the operation symbols are drawn from a nominal set rather than a set, and hence may have non-empty support. A *NEL-signature* Σ is specified by

- 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 function that assigns to each $op \in \text{Op}_\Sigma$ a type consisting of a finite (possibly empty) list \vec{s} of sorts of Σ and a sort s of Σ . As usual, the list $\vec{s} = [s_1, \dots, s_n]$ indicates the number and sort of arguments that op accepts and s indicates the sort of result it returns. We write

$$op : \vec{s} \rightarrow s \tag{19}$$

to indicate this typing information and say that op has *arity* n if \vec{s} is a list of length n . Equivariance of the typing function means that for all $\pi \in \text{Perm}$, if (19) holds, then so does $\pi \cdot op : \vec{s} \rightarrow s$. Thus for all possible types $\vec{s} \rightarrow s$, we can split Op_Σ into smaller nominal sets

$$\text{Op}_\Sigma(\vec{s}, s) \triangleq \{op \in \text{Op}_\Sigma \mid op : \vec{s} \rightarrow s\} \tag{20}$$

of operation symbols with that type.

Example 3.1 (*λ -Calculus*) Here is a NEL-signature for the untyped λ -calculus [2]. Fixing a sort of atoms $\nu \in \text{AtomSort}$ to represent names of variables, the theory's signature has a single sort tm (representing λ -terms) and nominal set of operation symbols

$$\{V_a \mid a \in \text{Atom}_\nu\} \cup \{L_a \mid a \in \text{Atom}_\nu\} \cup \{A\}$$

with Perm-action

$$\begin{aligned} \pi \cdot V_a &\triangleq V_{\pi(a)} \\ \pi \cdot L_a &\triangleq L_{\pi(a)} \\ \pi \cdot A &\triangleq A. \end{aligned}$$

The type of these operation symbols is defined to be

$$V_a : [] \rightarrow \text{tm}, \quad L_a : [\text{tm}] \rightarrow \text{tm}, \quad \text{and} \quad A : [\text{tm}, \text{tm}] \rightarrow \text{tm}.$$

In other words, the nominal set Op_Σ is isomorphic to the coproduct $\text{Atom}_\nu + \text{Atom}_\nu + 1$, where Atom_ν is the nominal set of atoms of sort ν (Definition 2.3) and 1 is the terminal nominal set (Lemma 2.6).

Remark 3.2 (*Nominal Signatures*) The reader familiar with the notion of *nominal signature* [34] should compare the above example with a nominal signature for λ -calculus, for example that given in [25, Example 2.1]. Compared with nominal

signatures, NEL-signatures avoid the use of both sorts of atoms and atom-binding sorts in arities, at the expense of having more operation symbols (typically, whole families of operation symbols parameterised by atoms) and specification of binding properties at the level of axioms rather than syntax. This is discussed more fully at the end of the paper in Section 11.

Given a NEL-signature Σ , a Σ -structure M in the category \mathcal{Nom} is specified by

- a nominal set $M[[\mathfrak{s}]]$ for each sort \mathfrak{s} of Σ ; and
- for each type $\vec{\mathfrak{s}} \rightarrow \mathfrak{s}$ of Σ , an equivariant function

$$\begin{aligned} M[[\!-\!]] : \text{Op}_\Sigma(\vec{\mathfrak{s}}, \mathfrak{s}) &\longrightarrow (M[[\vec{\mathfrak{s}}]] \rightarrow_{f_{\mathfrak{s}}} M[[\mathfrak{s}]]) \\ op &\mapsto M[[op]] \end{aligned} \tag{21}$$

where if $\vec{\mathfrak{s}} = [\mathfrak{s}_1, \dots, \mathfrak{s}_n]$, then $M[[\vec{\mathfrak{s}}]] \triangleq M[[\mathfrak{s}_1]] \times \dots \times M[[\mathfrak{s}_n]]$ is a finite product of nominal sets.

Note that because $\rightarrow_{f_{\mathfrak{s}}}$ is the exponential in the category \mathcal{Nom} (see Lemma 2.7), specifying an equivariant function as in (21) is equivalent to giving an equivariant function $\text{Op}_\Sigma(\vec{\mathfrak{s}}, \mathfrak{s}) \times M[[\vec{\mathfrak{s}}]] \longrightarrow M[[\mathfrak{s}]]$.

4 Terms and Values

The terms over a conventional algebraic signature are built up from variables by applying operation symbols. Given a structure in the category of sets for the signature and a valuation of the variables as elements of the structure, each term denotes an element of the structure. We wish to extend this to NEL-signatures and structures for them in the category \mathcal{Nom} of nominal sets. Doing so involves an extension of the usual notion of algebraic term to take account of the atom-permutation action that is part of the notion of nominal set. Since operations in a NEL-signature denote finitely supported functions (21), the action of a permutation on a compound term can distribute through the term to act on the operator and on its arguments, as in (17). Thus the only trace of the permutation action on terms that it is really necessary to incorporate into their structure is in the case that a permutation acts on a variable. So as in [34], we use *suspensions* πx consisting of a permutation π waiting to be applied once more is known about the unknown element of a nominal set represented by the variable x .⁵ Fixing a countably infinite set Var of *variables*, the grammar of terms over a NEL-signature Σ is given in Figure 2.

Notation 4.1 Note that all occurrences of variables x in terms are preceded by a suspended permutation π . However, when π is the identity permutation ι , we shall very often abbreviate the term ιx just to x .

Definition 4.2 (*Nominal Sets of Well-Sorted Terms*) A *sorting environment* over a NEL-signature Σ is a partial function Γ from a finite subset $\text{dom}(\Gamma) \subseteq \text{Var}$ of

⁵ The term “moderated variable” is also used for what we call suspensions: see [8,16].

Variables	$x \in \text{Var}$
Permutations	$\pi \in \text{Perm}$
Operation symbols	$op \in \text{Op}_\Sigma$
Terms	$t ::= \pi x \mid op t \cdots t$

Fig. 2. Terms over a NEL-signature, Σ

variables to the set Sort_Σ of sorts of the signature. The sets $\Sigma_s(\Gamma)$ of *terms of sort* $s \in \text{Sort}_\Sigma$ in a *sorting environment* Γ are inductively defined by:

- if $\pi \in \text{Perm}$, $x \in \text{dom}(\Gamma)$ and $\Gamma(x) = s$, then $\pi x \in \Sigma_s(\Gamma)$;
- if $op \in \text{Op}_\Sigma$ has type $[s_1, \dots, s_n] \rightarrow s$ and $t_i \in \Sigma_{s_i}(\Gamma)$ for $i = 1..n$, then $op t_1 \cdots t_n \in \Sigma_s(\Gamma)$. (In case $n = 0$, $op : [] \rightarrow s$ is usually called a *constant* of sort s , and we get $op \in \Sigma_s(\Gamma)$.)

We make each $\Sigma_s(\Gamma)$ into a nominal set as follows. The action $(\pi, t) \mapsto \pi \cdot t$ of atom-permutations on well-sorted terms is inherited from the given action on operators and the conjugation action on permutations (13):

$$\begin{aligned} \pi \cdot (\pi' x) &\triangleq \pi \pi' \pi^{-1} x \\ \pi \cdot (op t_1 \cdots t_n) &\triangleq (\pi \cdot op) (\pi \cdot t_1) \cdots (\pi \cdot t_n). \end{aligned} \tag{22}$$

As noted in Definition 2.5, permutations are finitely supported with respect to the conjugation action; and operators are finitely supported because they are elements of the given nominal set Op_Σ . It follows that with the above action, $\Sigma_s(\Gamma)$ is a nominal set and that its freshness relation $a \# t$ is given by:

$$\begin{aligned} a \# \pi x &\Leftrightarrow a \notin \text{dom}(\pi) \\ a \# (op t_1 \cdots t_n) &\Leftrightarrow a \# op \wedge a \# t_1 \wedge \cdots \wedge a \# t_n. \end{aligned} \tag{23}$$

Example 4.3 (*λ -Calculus*) For the NEL-signature in Example 3.1 it is not hard to see that when $\Gamma = \emptyset$ is the empty sorting environment, the nominal set $\Sigma_{\text{tm}}(\emptyset)$ is isomorphic to the usual set of abstract syntax trees for λ -terms (with variables ranging over Atom_ν) with Perm -action that applies a permutation to the atoms occurring in the leaves of a syntax tree. For example, the λ -term $\lambda a. \lambda b. a b$ corresponds to the element $L_a(L_b(A V_a V_b)) \in \Sigma_{\text{tm}}(\emptyset)$. However, for non-empty sorting environments we get generalised λ -terms, such as $L_a(L_b(A((a b)x) V_b)) \in \Sigma_{\text{tm}}([x : \text{tm}])$, with meta-level variables x standing for unknown λ -terms and suspended permutations $(a b)$ of object-level variables a and b (cf. [34]).

Next we describe the intended interpretation of terms as elements of nominal sets.

$$\begin{aligned}
(op\ t_1 \cdots t_n)\{\sigma\} &\triangleq op\ t_1\{\sigma\} \cdots t_n\{\sigma\} \\
(\pi\ x)\{\sigma\} &\triangleq \pi * \sigma(x)
\end{aligned}$$

where $(\pi, t) \mapsto \pi * t$ is defined by

$$\begin{aligned}
\pi * (\pi' x) &\triangleq \pi \pi' x \\
\pi * (op\ t_1 \cdots t_n) &\triangleq (\pi \cdot op) (\pi * t_1) \cdots (\pi * t_n) .
\end{aligned}$$

Fig. 3. Term substitution

Definition 4.4 (*Valuations*) Given a NEL-signature Σ , let M be a Σ -structure in \mathcal{Nom} as in Section 3 and let Γ be a sorting environment over Σ . The finite product in \mathcal{Nom} of the nominal sets $M[\Gamma(x)]$ as x ranges over $dom(\Gamma)$ will be written $M[\Gamma]$. We call the elements of this nominal set Γ -*valuations in M* . They are functions ρ defined on the finite set of variables $dom(\Gamma)$ and mapping each $x \in dom(\Gamma)$ to an element $\rho(x)$ of the nominal set $M[\Gamma(x)]$. Since $M[\Gamma]$ is given by a finite product of nominal sets, the action of a permutation $\pi \in \mathbf{Perm}$ on $\rho \in M[\Gamma]$ is given by:

$$(\pi \cdot \rho)(x) = \pi \cdot \rho(x) \quad (x \in dom(\Gamma)) \quad (24)$$

and (hence) $a \# \rho$ holds iff $\forall x \in dom(\rho). a \# \rho(x)$.

The *value* $M[t]\rho$ of a well-sorted term $t \in \Sigma_s(\Gamma)$ with respect to a valuation $\rho \in M[\Gamma]$ is an element of the nominal set $M[\mathbf{s}]$. Values are defined by recursion on the structure of terms:

$$\begin{aligned}
M[\pi x]\rho &\triangleq \pi \cdot \rho(x) \\
M[op\ t_1 \cdots t_n]\rho &\triangleq M[op](M[t_1]\rho, \dots, M[t_n]\rho) .
\end{aligned} \quad (25)$$

Combining (22), (24) and (25) with the fact (21) that $M[-]$ is an equivariant function, we get:

$$\pi \cdot (M[t]\rho) = M[\pi \cdot t](\pi \cdot \rho) . \quad (26)$$

So $(t, \rho) \mapsto M[t]\rho$ is an equivariant function $\Sigma_s(\Gamma) \times M[\Gamma] \longrightarrow M[\mathbf{s}]$.

5 Substitution

Given a NEL-signature Σ and sorting environments $\Gamma = [x_1 : \mathbf{s}_1, \dots, x_n : \mathbf{s}_n]$ and Γ' over Σ , the set $\Sigma(\Gamma, \Gamma')$ of *substitutions from Γ to Γ'* consists of functions σ mapping each variable x_i in $dom(\Gamma)$ to a term $\sigma(x_i) \in \Sigma_{\mathbf{s}_i}(\Gamma')$. Given a term $t \in \Sigma_s(\Gamma)$ and a substitution $\sigma \in \Sigma(\Gamma, \Gamma')$, we get a term $t\{\sigma\} \in \Sigma_s(\Gamma')$, defined as in Figure 3. The following standard properties of a notion of substitution are easily verified for the definition in the figure (by induction on the structure of terms):

$$t\{id\} = t \quad (27)$$

where $id \in \Sigma(\Gamma, \Gamma)$ is the *identity substitution*, $x \mapsto \iota x$; and

$$(t\{\sigma\})\{\sigma'\} = t\{\sigma; \sigma'\} \quad (28)$$

where $\sigma; \sigma' \in \Sigma(\Gamma, \Gamma'')$ is the *composition* of $\sigma \in \Sigma(\Gamma, \Gamma')$ and $\sigma' \in \Sigma(\Gamma', \Gamma'')$, given by

$$(\sigma; \sigma')(x) \triangleq \sigma(x)\{\sigma'\}. \quad (29)$$

The proof of (28) involves first proving:

$$(\pi * t)\{\sigma\} = \pi * (t\{\sigma\}) \quad (30)$$

by induction on the structure of t .

In the case $t = \pi x$ is a suspension, $t\{\sigma\}$ is the term $\pi * \sigma(x)$ obtained by distributing π through the structure of the term $\sigma(x)$ as in the second part of the Figure 3 (cf. [34, Fig. 1]). In forming $\pi * t$ from π and t , when π meets a sub-term of t that is another suspension, $\pi' x'$ say, the left multiplication action (12) is used and $\pi\pi' x'$ is formed. This, rather than the conjugation action (13), is needed here in order to ensure that the function $(t, \rho) \mapsto M[[t]]\rho$ is compositional, in the following sense.

Lemma 5.1 *Given a NEL-signature Σ , sorting environments Γ, Γ' and a Σ -structure M , then for all $t \in \Sigma_s(\Gamma)$, $\sigma \in \Sigma(\Gamma, \Gamma')$ and $\rho \in M[[\Gamma']]$*

$$M[[t\{\sigma\}]]\rho = M[[t]](M[[\sigma]]\rho) \quad (31)$$

where by definition $M[[\sigma]]\rho \in M[[\Gamma]]$ is the valuation mapping each $x \in \text{dom}(\Gamma)$ to $M[[\sigma(x)]]\rho$.

Proof. Using the definition of $\pi * (-)$ from the second part of Figure 3, along with (24), (25) and the fact that $op \mapsto M[[op]]$ is equivariant, it follows by induction on the structure of t that

$$M[[\pi * t]]\rho = \pi \cdot (M[[t]]\rho) \quad (32)$$

and from this we get (31), again by induction on the structure of t . \square

Property (32) shows that the $(\pi, t) \mapsto \pi * t$ action of permutations on terms denotes in nominal equational logic the built-in **Perm**-action of the nominal sets that interpret the sorts. Gabbay and Mathijssen [16] call this the “object-level” action of π on t . By contrast, the “meta-level” action $(\pi, t) \mapsto \pi \cdot t$, defined in (22), is the one appropriate to terms as *functions* of their variables via substitution. Recalling from (15) the action of atom-permutations on functions, we have the following result expressing the $(\pi, t) \mapsto \pi \cdot t$ action in terms of the $(\pi, t) \mapsto \pi * t$ action (cf. Gabbay and Mathijssen [16, Lemma 2.3]).

Lemma 5.2 *Given a NEL-signature Σ , a substitution $\sigma \in \Sigma(\Gamma, \Gamma')$ and a term $t \in \Sigma_s(\Gamma)$, for any $\pi \in \text{Perm}$*

$$(\pi \cdot t)\{\sigma\} = (\pi * t)\{\pi^{-1} * \sigma\}.$$

where by definition, $\pi^{-1} * \sigma \in \Sigma(\Gamma, \Gamma')$ is the substitution mapping each $x \in \text{dom}(\Gamma)$ to $\pi^{-1} * \sigma(x)$.

Proof. This can be proved by induction on the structure of t . In the base case that $t = \pi' x$ is a suspension, we have $(\pi \cdot t)\{\sigma\} = (\pi \cdot (\pi' x))\{\sigma\} \triangleq (\pi\pi' \pi^{-1} x)\{\sigma\} \triangleq \pi\pi' \pi^{-1} * \sigma(x)$ and also

$$\begin{aligned} (\pi * t)\{\pi^{-1} * \sigma\} &= (\pi * (\pi' x))\{\pi^{-1} * \sigma\} \triangleq (\pi\pi' x)\{\pi^{-1} * \sigma\} \\ &\triangleq \pi\pi' * (\pi^{-1} * \sigma)(x) \\ &\triangleq \pi\pi' * (\pi^{-1} * \sigma(x)) \\ &= \pi\pi' \pi^{-1} * \sigma(x) \end{aligned}$$

where in the last step we use the easily verified fact that $*$ is a Perm-action on terms. \square

As a corollary of this we have that $(t, \sigma) \mapsto t\{\sigma\}$ is equivariant:

Corollary 5.3 *Given a NEL-signature Σ , a substitution $\sigma \in \Sigma(\Gamma, \Gamma')$ and a term $t \in \Sigma_s(\Gamma)$, for any $\pi \in \text{Perm}$*

$$\pi \cdot (t\{\sigma\}) = (\pi \cdot t)\{\pi \cdot \sigma\}$$

where by definition, $\pi \cdot \sigma \in \Sigma(\Gamma, \Gamma')$ is the substitution mapping each $x \in \text{dom}(\Gamma)$ to $\pi \cdot \sigma(x)$.

Proof. By induction on the structure of t , using the special case of Lemma 5.2 when $\sigma = id$ in the base case that t is a suspension, along with (27) and Figure 3. \square

Note that under the action $(\pi, \sigma) \mapsto \pi \cdot \sigma$, each set of substitutions $\Sigma(\Gamma, \Gamma')$ is a nominal set: σ is supported by any finite set of atoms that supports all of the finitely many terms $\sigma(x)$ as x ranges over $\text{dom}(\Gamma)$.

6 Theories and Algebras

Ordinary equational logic formalises reasoning about equations between algebraic terms. As explained in the Introduction, we wish to formalise reasoning both about equality and about the freshness relation of Definition 2.1. In the formal system we will use the symbols “ \approx ” and “ $\#$ ” for the equality and freshness relations, and continue to use “ $=$ ” and “ $\#$ ” for their interpretation in nominal sets as the actual equality and “not-in-the-support-of” relations.

As also discussed in the Introduction, it is natural to allow assertions about equality and freshness to be conditioned by assumptions about which atoms are

fresh for particular elements. 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-theory* \mathbb{T} to consist of a NEL-signature Σ together with a collection of *axioms* of the form

$$\nabla \vdash \bar{a} \# t \approx t' : \mathfrak{s} \quad (33)$$

where

- ∇ is a *freshness environment*, which by definition is partial function defined on a finite subset $\text{dom}(\nabla) \subseteq \text{Var}$ of variables and mapping each $x_i \in \text{dom}(\nabla)$ to a pair $\nabla(x_i) = (\mathfrak{s}_i, \bar{a}_i) \in \text{Sort}_\Sigma \times \mathcal{P}_{\text{fin}}(\text{Atom})$ of a sort and a finite set of atoms;
- $\bar{a} \in \mathcal{P}_{\text{fin}}(\text{Atom})$; and
- $t, t' \in \Sigma_{\mathfrak{s}}(\nabla)$ are terms of the same sort $\mathfrak{s} \in \text{Sort}_\Sigma$ in the sorting environment ∇ obtained from ∇ by composing with first projection.

If $\text{dom}(\nabla)$ consists of the distinct variables x_1, \dots, x_n and $\nabla(x_i) = (\mathfrak{s}_i, \bar{a}_i)$ for $i = 1..n$, then we write ∇ as

$$\nabla = [\bar{a}_1 \# x_1 : \mathfrak{s}_1, \dots, \bar{a}_n \# x_n : \mathfrak{s}_n] \quad (34)$$

in which case the associated sorting environment is

$$\nabla \cdot = [x_1 : \mathfrak{s}_1, \dots, x_n : \mathfrak{s}_n]. \quad (35)$$

We let Perm act on freshness environments (34) as follows, using the action of permutations on finite sets of atoms from Definition 2.4:

$$\pi \cdot \nabla \triangleq [\pi \cdot \bar{a}_1 \# x_1 : \mathfrak{s}_1, \dots, \pi \cdot \bar{a}_n \# x_n : \mathfrak{s}_n]. \quad (36)$$

This action makes the collection of all freshness environments into a nominal set for which the freshness relation is:

$$a \# \nabla \Leftrightarrow a \notin \bar{a}_1 \cup \dots \cup \bar{a}_n. \quad (37)$$

Notation 6.1 Although the single form of judgement (33) combining equality and freshness is useful for stating the general rules of nominal equational logic, in particular cases it is clearer to use the following abbreviations.

- $t \approx t' : \mathfrak{s}$ means $\emptyset \# t \approx t' : \mathfrak{s}$; similarly, $x : \mathfrak{s}$ in a freshness environment means $\emptyset \# x : \mathfrak{s}$.
- $\bar{a} \# t : \mathfrak{s}$ means $\bar{a} \# t \approx t : \mathfrak{s}$.
- $a \# t \approx t' : \mathfrak{s}$ means $\{a\} \# t \approx t' : \mathfrak{s}$; similarly, $a \# x : \mathfrak{s}$ in a freshness environment means $\{a\} \# x : \mathfrak{s}$.

Example 6.2 (*λ -Terms Modulo $\alpha\beta\eta$ -Equivalence*) Figure 4 gives a NEL-theory over the signature from Example 3.1 for $\alpha\beta\eta$ -equivalence of untyped λ -terms [2]. The theory has seven axioms, making use of variables $x, x', x_1, x_2 \in \text{Var}$ and atoms

$$\begin{aligned}
& x : \mathbf{tm} \vdash a \# L_a x : \mathbf{tm} & (\alpha) \\
& a \# x : \mathbf{tm}, x' : \mathbf{tm} \vdash A(L_a x) x' \approx x : \mathbf{tm} & (\beta-1) \\
& x' : \mathbf{tm} \vdash A(L_a V_a) x' \approx x' : \mathbf{tm} & (\beta-2) \\
& x : \mathbf{tm}, a' \# x' : \mathbf{tm} \vdash A(L_a(L_{a'} x)) x' \approx L_{a'}(A(L_a x) x') : \mathbf{tm} & (\beta-3) \\
& x_1 : \mathbf{tm}, x_2 : \mathbf{tm}, x' : \mathbf{tm} \vdash A(L_a(A x_1 x_2)) x' \approx & \\
& \quad A(A(L_a x_1) x') (A(L_a x_2) x') : \mathbf{tm} & (\beta-4) \\
& a' \# x : \mathbf{t} \vdash A(L_a x) V_{a'} \approx (a a') x : \mathbf{tm} & (\beta-5) \\
& a \# x : \mathbf{tm} \vdash x \approx L_a(A x V_a) : \mathbf{tm} & (\eta)
\end{aligned}$$

Fig. 4. A NEL-theory for $\alpha\beta\eta$ -equivalence

$a, a' \in \mathbf{Atom}_\nu$.⁶ Although (α) is an axiom about freshness, we will see below (Example 7.5) that it gives the effect of α -equivalence modulo the rules of nominal equational logic. For β -equivalence we adapt the Gabbay-Mathijssen nominal algebra for capture-avoiding substitution [16, Fig. 4]. Axioms $(\beta-1)$ – $(\beta-4)$ unwind the capture-avoiding substitution in a conventional β -conversion, according to the structure of t in a β -redex $A(L_a t) t'$. The axiom $(\text{ren} \mapsto)$ in [16, Fig. 4] connecting capture-avoiding substitution with name-permutation becomes $(\beta-5)$. Finally, for η -equivalence we use the axiom (η) . The relationship between this NEL-theory and the classical notion of $\alpha\beta\eta$ -equivalence of syntax trees for λ -terms (and the associated freshness relation “not a free variable of”) will be explored in Example 9.5.

Turning to the interpretation of NEL-theories in \mathcal{Nom} , first note that the intended meaning of the freshness environment (34) is to assert not only that each variable x_i has sort \mathbf{s}_i , but also that it stands for an element of the corresponding nominal set whose support is disjoint from \bar{a}_i . Accordingly, we take the meaning of ∇ in a Σ -structure M to be the subset $M[\nabla] \subseteq M[\nabla^\cdot]$ of the nominal set of valuations (Definition 4.4) given by

$$M[\nabla] \triangleq \{\rho \in M[\nabla^\cdot] \mid \bar{a}_1 \# \rho(x_1) \wedge \cdots \wedge \bar{a}_n \# \rho(x_n)\} \quad (38)$$

where $\#$ is the freshness relation (Definition 2.1) for each nominal set $M[\mathbf{s}_i]$ and ∇^\cdot is the sorting environment associated with ∇ as in (35).

Definition 6.3 (*Satisfaction*) Let Σ be a NEL-signature. A Σ -structure M satisfies a judgement $\nabla \vdash \bar{a} \# t \approx t' : \mathbf{s}$ if for all $\rho \in M[\nabla]$ it is the case both that $M[t]\rho$ and $M[t']\rho$ are equal elements of the nominal set $M[\mathbf{s}]$ and that the freshness relation $\bar{a} \# M[t]\rho$ holds in $M[\mathbf{s}]$.

Given a NEL-theory \mathbb{T} , a \mathbb{T} -algebra in \mathcal{Nom} is a structure for the signature of \mathbb{T} that satisfies all its axioms. Given a judgement $\nabla \vdash \bar{a} \# t \approx t' : \mathbf{s}$, the *semantic*

⁶ We make use of the abbreviations from Notation 4.1 and 6.1 to state the axioms; for example, (α) written out in full is: $\emptyset \# x : \mathbf{tm} \vdash \{a\} \# L_a(\iota x) \approx L_a(\iota x) : \mathbf{tm}$.

$$\begin{array}{c}
\text{(REFL)} \frac{}{\nabla \vdash t \approx t : s} \quad t \in \Sigma_s(\nabla^\cdot) \qquad \text{(SYMM)} \frac{\nabla \vdash \bar{a} \# t \approx t' : s}{\nabla \vdash \bar{a} \# t' \approx t : s} \\
\\
\text{(TRANS)} \frac{\nabla \vdash \bar{a}_1 \# t \approx t' : s \quad \nabla \vdash \bar{a}_2 \# t' \approx t'' : s}{\nabla \vdash (\bar{a}_1 \cup \bar{a}_2) \# t \approx t'' : s} \\
\\
\text{(SUBST)} \frac{\nabla' \vdash \sigma \approx \sigma' : \nabla \quad \nabla \vdash \bar{a} \# t \approx t' : s}{\nabla' \vdash \bar{a} \# t\{\sigma\} \approx t'\{\sigma'\} : s} \quad \sigma, \sigma' \in \Sigma(\nabla^\cdot, (\nabla')^\cdot) \\
\\
\text{(WEAK)} \frac{\nabla \vdash \bar{a} \# t \approx t' : s}{\nabla' \vdash \bar{a} \# t \approx t' : s} \quad \nabla \leq \nabla' \qquad \text{(ATM-INTRO)} \frac{\nabla \vdash \bar{a} \# t \approx t' : s}{\nabla \#^a \vdash \bar{a} \cup \{a\} \# t \approx t' : s} \quad a \# (\bar{a}, t, t') \\
\\
\text{(ATM-ELIM)} \frac{\nabla \#^a \vdash \bar{a} \# t \approx t' : s}{\nabla \vdash \bar{a} \# t \approx t' : s} \quad a \# (\nabla, \bar{a}, t, t') \qquad \text{(\#-EQUIVAR)} \frac{}{\bar{a} \# x : s \vdash \pi \cdot \bar{a} \# \pi x : s} \\
\\
\text{(SUSP)} \frac{}{\{a \mid \pi(a) \neq \pi'(a)\} \# x : s \vdash \pi x \approx \pi' x : s}
\end{array}$$

Fig. 5. The Rules of Nominal Equational Logic

consequence relation

$$\nabla \vDash_{\mathbb{T}} \bar{a} \# t \approx t' : s \quad (39)$$

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

7 Nominal Equational Logic

Figure 5 gives a collection of rule schemes for inductively generating judgements of the form $\nabla \vdash \bar{a} \# t \approx t' : s$. The rules preserve the well-formedness condition we placed on judgements at the beginning of Section 6, namely that the equated terms both have the given sort in the sorting environment associated with the given freshness environment.

Notation 7.1 Figure 5 makes use of the following notation.

- Rules (REFL), (#-EQUIVAR) and (SUSP) make use of the abbreviations for judgements introduced in Notation 6.1.
- In rule (REFL), ∇^\cdot denotes the sorting environment associated with a freshness environment ∇ as in (35).
- In rule (SUBST), $\sigma, \sigma' \in \Sigma(\nabla^\cdot, (\nabla')^\cdot)$ are substitutions (Section 5) and

$$\nabla' \vdash \sigma \approx \sigma' : \nabla \quad (40)$$

stands for the finite number of hypotheses $\nabla' \vdash \bar{a}_i \# \sigma(x_i) \approx \sigma'(x_i) : s_i$ for

$i = 1..n$, assuming $\nabla = [\bar{a}_1 \# x_1 : s_1, \dots, \bar{a}_n \# x_n : s_n]$. The operation of substitution, $t\{\sigma\}$, used in the rule was defined in Figure 3.

- The relation

$$\nabla \leq \nabla' \quad (41)$$

of *weakening* between freshness environments used as a side-condition in rule (WEAK) is defined to hold if $\text{dom}(\nabla) \subseteq \text{dom}(\nabla')$ and for all $x \in \text{dom}(\nabla)$, if $\nabla(x) = (s, \bar{a})$, then $\nabla'(x) = (s, \bar{a}')$ for some $\bar{a}' \supseteq \bar{a}$.

- In rules (ATM-INTRO) and (ATM-ELIM), if $\nabla = [\bar{a}_1 \# x_1 : s_1, \dots, \bar{a}_n \# x_n : s_n]$, then

$$\nabla^{\#a} \triangleq [\bar{a}_1 \cup \{a\} \# x_1 : s_1, \dots, \bar{a}_n \cup \{a\} \# x_n : s_n]. \quad (42)$$

- In rule (ATM-INTRO) the side-condition “ $a \# (\bar{a}, t, t')$ ” refers to the semantic freshness relation (Definition 2.1) in the product nominal set $\mathcal{P}_{\text{fin}}(\text{Atom}) \times \Sigma_s(\nabla) \times \Sigma_s(\nabla)$. In other words the condition is that $a \notin \bar{a}$ holds and that the relations $a \# t$ and $a \# t'$ hold as defined in (23). Similarly the side-condition “ $a \# (\nabla, \bar{a}, t, t')$ ” to rule (ATM-ELIM) means that these properties hold, together with $a \# \nabla$, as in (37).

Definition 7.2 (*Logical Consequence*) The set of *theorems* of a NEL-theory \mathbb{T} is the least set of judgements containing the axioms of \mathbb{T} and closed under the rules in Figure 5. We write

$$\nabla \vdash_{\mathbb{T}} \bar{a} \# t \approx t' : s \quad (43)$$

to indicate that the judgement is a theorem of \mathbb{T} and call (43) the *logical consequence relation*.

We are going to show that the rules in Figure 5 are both sound and complete for the interpretation of judgements in \mathcal{Nom} . In other words, we will show that the logical consequence relation coincides with the semantic consequence relation of Definition 6.3. Completeness will eventually be proved in Section 10. For the moment we concentrate on the simpler property of soundness.

The rules of nominal equational logic combine the usual properties of equality (that it is an equivalence relation and is preserved under substituting equal terms) with some properties of the nominal sets notion of freshness (Definition 2.1) that have been identified in the literature [24,14,34,13,16] and which are listed in the following lemma.

Lemma 7.3 *Let x be an element of a nominal set X .*

- (i) *For each sort of atoms $\alpha \in \text{AtomSort}$, there is some $a \in \text{Atom}_\alpha$ with $a \# x$.*
- (ii) *If $f : X \rightarrow Y$ is a morphism in \mathcal{Nom} and $a \# x$, then $a \# f(x)$.*
- (iii) *If $\pi \in \text{Perm}$ and $a \# x$, then $\pi(a) \# \pi \cdot x$.*
- (iv) *If $\pi, \pi' \in \text{Perm}$ and $\{a \in \text{Atom} \mid \pi(a) \neq \pi'(a)\} \# x$, then $\pi \cdot x = \pi' \cdot x$.*

Proof. Part (i) holds because support sets are finite, whereas the set Atom_α is infinite. For part (ii), just note that since f is equivariant, if $\bar{a} \in \mathcal{P}_{\text{fin}}(\text{Atom})$ supports x in X , then \bar{a} supports $f(x)$ in Y . For the proof of part (iii), see [25, Lemma 3.7].

For part (iv), since $\{a \mid \pi(a) \neq \pi'(a)\} = \{a \mid \pi^{-1}\pi'(a) \neq a\} = \text{dom}(\pi^{-1}\pi')$, it suffices to prove a more general version of Lemma 2.2:

$$\forall \pi \in \text{Perm. } \text{dom}(\pi) \# x \Rightarrow \pi \cdot x = x. \quad (44)$$

This can be done by induction on the size of the finite set $\text{dom}(\pi)$ (for all π simultaneously). In the base case $\text{dom}(\pi) = \emptyset$, $\pi(a) = a$ for all a , so $\pi = \text{id}$ and thus $\pi \cdot x = \text{id} \cdot x = x$ by definition of action. For the induction step, suppose $\text{dom}(\pi)$ is non-empty and $\text{dom}(\pi) \# x$, that is, $a \# x$ holds for all a with $\pi(a) \neq a$. Picking some $a \in \text{dom}(\pi)$, we first show that $\text{dom}((\pi(a) a)\pi) \subseteq \text{dom}(\pi) - \{a\}$. Take some $b \in \text{dom}((\pi(a) a)\pi)$, that is $((\pi(a) a)\pi)(b) \neq b$. If $\pi(b) = \pi(a)$ then $a = ((\pi(a) a)\pi)(b) \neq b$; but this contradicts the bijectivity of π , so $\pi(b) \neq \pi(a)$, so $b \neq a$. Then

$$\text{either } \pi(b) = a$$

$$\text{or } \pi(b) \neq a, \text{ so } \pi(b) = ((\pi(a) a)\pi)(b) \neq b.$$

In each case $\pi(b) \neq b \neq a$, that is, $b \in \text{dom}(\pi) - \{a\}$.

Since $\text{dom}(\pi) - \{a\}$ has strictly fewer elements than $\text{dom}(\pi)$, so does $\text{dom}((\pi(a) a)\pi)$ and so by the induction hypothesis $(\pi(a) a)\pi \cdot x = x$. So $\pi \cdot x = (a \pi(a)) \cdot x$ and we just have to see that $(a \pi(a)) \cdot x = x$. Since $a \in \text{dom}(\pi)$ it is also the case that $\pi(a) \in \text{dom}(\pi)$; but $\text{dom}(\pi) \# x$ and thus $a \# x$ and $\pi(a) \# x$; therefore by Lemma 2.2, $(a \pi(a)) \cdot x = x$. \square

Theorem 7.4 (Soundness) *If a judgement $\nabla \vdash \bar{a} \# t \approx t' : s$ is a theorem of a NEL-theory \mathbb{T} , then it is satisfied by any \mathbb{T} -algebra in Nom :*

$$\nabla \vdash_{\mathbb{T}} \bar{a} \# t \approx t' : s \Rightarrow \nabla \vDash_{\mathbb{T}} \bar{a} \# t \approx t' : s.$$

Proof. Let M be a \mathbb{T} -algebra. We have to show that the collection of judgements satisfied by M (Definition 6.3) is closed under each of the rules in Figure 5.

Closure of satisfaction under rules (REFL), (SYMM) and (TRANS) is immediate from Definition 6.3. Closure under rule (SUBST) follows from the compositionality property (31) of the function $(t, \rho) \mapsto M[[t]]\rho$. It is easy to see from the definition of that function in (25) that the value $M[[t]]\rho$ of any term t only depends on the values of ρ at variables that actually occur in the term; closure of satisfaction under rule (WEAK) follows easily from this observation. Closure under rules ($\#$ -EQUIVAR) and (SUSP) follows directly from the corresponding properties (iii) and (iv) of freshness in Lemma 7.3. The only two remaining cases are for rules (ATM-INTRO) and (ATM-ELIM), and they are worth giving in detail.

For rule (ATM-INTRO), if $\rho \in M[[\nabla \#^a]]$, then $\forall x \in \text{dom}(\rho). a \# \rho(x)$ and hence as noted in Definition 4.4, $a \# \rho$. If we also have $a \# (t, t')$, then by Lemma 7.3(ii) applied to the function $(t, \rho) \mapsto M[[t]]\rho$ (which we noted in (26) is equivariant), we have $a \# M[[t]]\rho$. Hence if M satisfies $\nabla \vdash \bar{a} \# t \approx t' : s$, it also satisfies $\nabla \#^a \vdash \bar{a} \cup \{a\} \# t \approx t' : s$ for any a with $a \# (t, t')$.⁷

⁷ Rule (ATM-INTRO) also includes the inessential side-condition $a \notin \bar{a}$, since without it the rule becomes an

For the rule (ATM-ELIM) , suppose $a \# (\nabla, \bar{a}, t, t')$. If $\rho \in M[\nabla]$, then we can use Lemma 7.3(i) to find an atom a' (of the same sort as a) with $a' \# (\rho, \nabla, \bar{a}, t, t')$. Note that since $a \# \nabla$ and $a' \# \nabla$, we have $(a \ a') \cdot \rho \in M[\nabla]$. In fact $(a \ a') \cdot \rho \in M[\nabla \#^a]$ since $a \# (a \ a') \cdot \rho$ (by Lemma 7.3(iii) applied to $a' \# \rho$). So if M satisfies $\nabla \#^a \vdash \bar{a} \cup \{a\} \not\# t \approx t' : s$, then

$$\bar{a} \# M[t]((a \ a') \cdot \rho) = M[t']((a \ a') \cdot \rho) \in M[s]. \quad (45)$$

Since the function $(t, \rho) \mapsto M[t]\rho$ is equivariant, we can apply $(a \ a') \cdot (-)$ to (45) and use Lemma 7.3(iii) to get

$$(a \ a') \cdot \bar{a} \# M[(a \ a') \cdot t]((a \ a')(a \ a') \cdot \rho) = M[(a \ a') \cdot t']((a \ a')(a \ a') \cdot \rho) \in M[s].$$

But $(a \ a')(a \ a') = \iota$ and since $\{a, a'\} \# (\bar{a}, t, t')$, by Lemma 2.2 we also have $(a \ a') \cdot \bar{a} = \bar{a}$, $(a \ a') \cdot t = t$ and $(a \ a') \cdot t' = t'$; so $\bar{a} \# M[t]\rho = M[t']\rho \in M[s]$. Therefore M also satisfies $\nabla \vdash \bar{a} \not\# t \approx t' : s$. \square

We end this section with an example of nominal equational reasoning.

Example 7.5 (α -Equivalence) Let \mathbb{T}_α be the NEL-theory with signature as in Example 3.1 and whose single axiom is the judgement (α) from Figure 4. To illustrate nominal equational reasoning, we show that α -equivalent λ -abstractions are provably equal, in the sense that if $a \neq a'$ are unequal elements of Atom_ν , then

$$a' \not\# x : \text{tm} \vdash L_a x \approx L_{a'}(a \ a')x : \text{tm} \quad (46)$$

is a theorem of \mathbb{T}_α .⁸ To see this, first note that by (REFL) and (ATM-INTRO) we have

$$a' \not\# x : \text{tm} \vdash_{\mathbb{T}_\alpha} a' \not\# L_a x : \text{tm} \quad (47)$$

and by (WEAK) applied to (α) we also have

$$a' \not\# x : \text{tm} \vdash_{\mathbb{T}_\alpha} a \not\# L_a x : \text{tm}. \quad (48)$$

Applying (TRANS) to (47) and (48) yields

$$a' \not\# x : \text{tm} \vdash_{\mathbb{T}_\alpha} \{a, a'\} \not\# L_a x : \text{tm}. \quad (49)$$

Thus taking $\nabla = [\{a, a'\} \not\# x : \text{tm}]$, $\nabla' = [a' \not\# x : \text{tm}]$ and $\sigma \in \Sigma(\nabla; (\nabla'))$ to be the substitution $x \mapsto L_a x$, (49) gives us

$$\nabla' \vdash_{\mathbb{T}_\alpha} \sigma \approx \sigma : \nabla. \quad (50)$$

An instance of (SUSP) with $\pi = \iota$ (the identity permutation) and $\pi' = (a \ a')$ is

$$\{a, a'\} \not\# x : \text{tm} \vdash_{\mathbb{T}_\alpha} \iota x \approx (a \ a')x : \text{tm}. \quad (51)$$

instance of (WEAK) in the case that $a \in \bar{a}$.

⁸ Here we are using the formulation of α -equivalence in terms of swapping with a fresh name: cf. [17, Proposition 2.2].

Since $\iota x\{\sigma\} = \iota * \sigma(x) = \iota * (L_a x) = L_a x$ and $((a \ a')x)\{\sigma\} = (a \ a') * \sigma(x) = (a \ a') * (L_a x) = L_{a'}((a \ a')x)$, we can apply (SUBST) to (50) and (51) to get (46) as a theorem of \mathbb{T}_α , as required.

8 Equivariance

In ordinary equational logic we are used to the idea that a single axiom involving variables stands for a whole family of facts, obtained by substituting particular terms for the variables. In nominal equational logic, axioms involve not just variables, but also names, represented by atoms. For example the axiom

$$x : \mathbf{tm} \vdash a \not\# L a x : \mathbf{tm} \quad (\alpha)$$

from the NEL-theory in Figure 4 involves a particular atom $a \in \mathbf{Atom}_\nu$ as well as the variable $x \in \mathbf{Var}$. Just as for ordinary equational logic, we can use rule (SUBST) from Figure 5 to replace x by particular terms. But what about replacing a by a different atom a' ? If $a' \neq a$, then the judgement $x : \mathbf{tm} \vdash a' \not\# L a' x : \mathbf{tm}$ is not an axiom of the theory in Figure 4, by definition. Nevertheless it is a theorem of that theory. This is because the logical consequence relation (43) for any NEL-theory \mathbb{T} turns out to be invariant under permuting atoms, even though we make no assumption that the set of axioms of \mathbb{T} is closed under the permutation action.

Theorem 8.1 (Equivariance of Logical Consequence) *For any NEL-theory \mathbb{T} , if $\nabla \vdash_{\mathbb{T}} \bar{a} \not\# t \approx t' : \mathbf{s}$, then for all $\pi \in \mathbf{Perm}$, $\pi \cdot \nabla \vdash_{\mathbb{T}} \pi \cdot \bar{a} \not\# \pi \cdot t \approx \pi \cdot t' : \mathbf{s}$.*

The theorem is a corollary of Lemma 5.2 and the following result.

Lemma 8.2 *For any NEL-theory \mathbb{T} , if*

$$\nabla \vdash_{\mathbb{T}} \bar{a} \not\# t \approx t' : \mathbf{s} \quad (52)$$

then for all $\pi \in \mathbf{Perm}$

$$\nabla \vdash_{\mathbb{T}} \pi \cdot \bar{a} \not\# \pi * t \approx \pi * t' : \mathbf{s} . \quad (53)$$

Proof. If (52) holds, then we have $\nabla \vdash \sigma \approx \sigma' : [\bar{a} \not\# x : \mathbf{s}]$, where σ and σ' are the substitutions mapping x to t and t' respectively. Applying (SUBST) to this and ($\not\#$ -EQUIVAR) gives (53). \square

Proof of Theorem 8.1. Suppose that $\nabla = [\bar{a}_1 \not\# x_1 : \mathbf{s}_1, \dots, \bar{a}_n \not\# x_n : \mathbf{s}_n]$. Given $\pi \in \mathbf{Perm}$, consider the substitution $\pi^{-1} * id$ defined as in Lemma 5.2. It maps each x_i to $\pi^{-1}x_i$ ($i = 1..n$) and is an element of $\Sigma((\pi \cdot \nabla)^\cdot, \nabla^\cdot)$. By (REFL) (for $t = x_i$), (ATM-INTRO) (applied repeatedly for each of the atoms in $\pi \cdot \bar{a}_i$), (WEAK) (with respect to $[\pi \cdot \bar{a}_i \not\# x_i : \mathbf{s}_i] \leq \pi \cdot \nabla$), and ($\not\#$ -EQUIVAR) (for the permutation π^{-1}), we have

$$\pi \cdot \nabla \vdash_{\mathbb{T}} \bar{a}_i \not\# \pi^{-1}x_i \approx \pi^{-1}x_i : \mathbf{s}_i \quad (i = 1..n)$$

and hence

$$\pi \cdot \nabla \vdash_{\mathbb{T}} \pi^{-1} * id \approx \pi^{-1} * id : \nabla . \quad (54)$$

So if (52) holds, then by Lemma 8.2 so does (53) and we can apply (SUBST) to this and (54) to deduce $\pi \cdot \nabla \vdash_{\mathbb{T}} \pi \cdot \bar{a} \# (\pi * t) \{ \pi^{-1} * id \} \approx (\pi * t') \{ \pi^{-1} * id \} : s$. Lemma 5.2 gives us $(\pi * t) \{ \pi^{-1} * id \} = (\pi \cdot t) \{ id \}$; and the latter is $\pi \cdot t$, by (27). Similarly, $(\pi * t') \{ \pi^{-1} * id \} = \pi \cdot t'$. Therefore we have $\pi \cdot \nabla \vdash_{\mathbb{T}} \pi \cdot \bar{a} \# \pi \cdot t \approx \pi \cdot t' : s$, as required. \square

Remark 8.3 (*Theorems of \mathbb{T} form a Nominal Set*) Note that the set of judgements of the form (33) over a NEL-signature Σ , once equipped with the atom-permutation action

$$\pi \cdot (\nabla \vdash \bar{a} \# t \approx t' : s) \triangleq (\pi \cdot \nabla \vdash \pi \cdot \bar{a} \# \pi \cdot t \approx \pi \cdot t' : s) \quad (55)$$

forms a nominal set. The freshness relation in this nominal set is

$$a \# (\nabla \vdash \bar{a} \# t \approx t' : s) \Leftrightarrow a \# \nabla \wedge a \notin \bar{a} \wedge a \# t \wedge a \# t' \quad (56)$$

using the freshness relation for terms (23) and for freshness environments (37).

Given a NEL-theory \mathbb{T} over Σ , Theorem 8.1 says that its set of theorems is closed under the permutation action (55). Therefore it too is a nominal set, with freshness relation as in (56).

9 Ground Term Algebras

In this section we show how to form a \mathbb{T} -algebra in \mathcal{Nom} from the terms of a NEL-theory \mathbb{T} that do not involve any variables. The construction provides a stepping stone towards the completeness result of the next section.

Definition 9.1 (*Ground Terms*) Let Σ be a NEL-signature. The set of *ground terms* of sort $s \in \text{Sort}_{\Sigma}$ over Σ is defined to be $\Sigma_s(\emptyset)$, that is, the set of terms that are well-sorted of sort s in the empty sorting environment, \emptyset . Note from Definition 4.2 that if t is a ground term it cannot involve any sub-terms that are suspensions, πx .

Now let \mathbb{T} be a NEL-theory with signature Σ . By virtue of the rules (REFL), (SYMM) and (TRANS) in Figure 5, the logical consequence relation of Definition 7.2 gives rise to an equivalence relation on $\Sigma_s(\emptyset)$ that relates t and t' if $\emptyset \vdash_{\mathbb{T}} t \approx t' : s$. Let $M_{\mathbb{T}}[[s]]$ denote the quotient of $\Sigma_s(\emptyset)$ by this equivalence relation. We write the equivalence class of t as $[t]$.

Recall from Section 4 that each set of terms $\Sigma_s(\Gamma)$ is a nominal set once we endow it with the Perm-action $(\pi, t) \mapsto \pi \cdot t$ of (22). In the case of ground terms, when $\Gamma = \emptyset$, Lemma 5.2 implies that this action coincides with the one associated with substitution in Figure 3:

$$\forall t \in \Sigma_s(\emptyset). \pi \cdot t = \pi * t. \quad (57)$$

Note that by Theorem 8.1, this Perm-action on $\Sigma_s(\emptyset)$ preserves the equivalence relation $\emptyset \vdash_{\mathbb{T}} t \approx t' : s$. Hence we get a well-defined action on the quotient set $M_{\mathbb{T}}[[s]]$, defined by

$$\pi \cdot [t] \triangleq [\pi \cdot t] = [\pi * t]. \quad (58)$$

It is a fact about quotients in $\mathcal{N}om$ in general that with this action $M_{\mathbb{T}}[\mathbf{s}]$ is a nominal set. For if a finite set $\bar{a} \in \mathcal{P}_{\text{fin}}(\text{Atom})$ supports t in $\Sigma_{\mathbf{s}}(\emptyset)$, then it also supports $[t]$ in $M_{\mathbb{T}}[\mathbf{s}]$, because for any $a, a' \notin \bar{a}$ (of the same sort) $(a \ a') \cdot [t] = [(a \ a') \cdot t] = [t]$. Thus

$$a \# t \Rightarrow a \# [t]. \quad (59)$$

However, we can be more precise about the freshness relation for the nominal set $M_{\mathbb{T}}[\mathbf{s}]$. As the following lemma shows, the semantic notion of freshness (Definition 2.1) coincides with the logical one determined by the rules in Figure 5 when one restricts to ground terms.

Lemma 9.2 (Semantic Freshness = Ground Logical Freshness) *For all $t \in \Sigma_{\mathbf{s}}(\emptyset)$ and $a \in \text{Atom}$*

$$a \# [t] \in M_{\mathbb{T}}[\mathbf{s}] \Leftrightarrow \emptyset \vdash_{\mathbb{T}} a \not\# t : \mathbf{s}. \quad (60)$$

Proof. Given $t \in \Sigma_{\mathbf{s}}(\emptyset)$ and an atom a , of sort α say, by Lemma 7.3(i) applied to the nominal set $\text{Atom}_{\alpha} \times \Sigma_{\mathbf{s}}(\emptyset)$, there is some $a' \in \text{Atom}_{\alpha}$ with $a' \# (a, t)$. By (REFL) and (ATM-INTRO) we have

$$\emptyset \vdash_{\mathbb{T}} a' \not\# t : \mathbf{s} \quad (61)$$

and hence by ($\not\#$ -EQUIVAR) and (SUBST)

$$\emptyset \vdash_{\mathbb{T}} a \not\# (a \ a') * t : \mathbf{s}. \quad (62)$$

Note that since $a' \# t$, by (59) we also have

$$a' \# [t]. \quad (63)$$

Suppose $a \# [t]$ holds. Then by Lemma 2.2, $(a \ a') \cdot [t] = [t]$. Therefore by (57), $[(a \ a') * t] = [(a \ a') \cdot t] = (a \ a') \cdot [t] = [t]$ and hence $\emptyset \vdash_{\mathbb{T}} (a \ a') * t \approx t : \mathbf{s}$. Applying (TRANS) and (SYMM) to this and (62) yields $\emptyset \vdash_{\mathbb{T}} a \not\# t : \mathbf{s}$.

Conversely, if $\emptyset \vdash_{\mathbb{T}} a \not\# t : \mathbf{s}$ holds, then by (61) and (TRANS) we have $\emptyset \vdash_{\mathbb{T}} \{a, a'\} \not\# t : \mathbf{s}$; and hence by (SUSP) and (SUBST), $\emptyset \vdash_{\mathbb{T}} (a \ a') * t \approx t : \mathbf{s}$. In other words $[(a \ a') * t] = [t]$ and thus as above, $(a \ a') \cdot [t] = [t]$. Then by Lemma 7.3(iii) on (63) we get $a \# (a \ a') \cdot [t] = [t]$. \square

To make $M_{\mathbb{T}}$ into a structure for the signature Σ underlying \mathbb{T} , we have to give for each type $\vec{\mathbf{s}} \rightarrow \mathbf{s}$ of Σ an equivariant function

$$M_{\mathbb{T}}[\] : \text{Op}_{\Sigma}(\vec{\mathbf{s}}, \mathbf{s}) \longrightarrow (M_{\mathbb{T}}[\vec{\mathbf{s}}] \rightarrow_{f_{\mathbf{s}}} M_{\mathbb{T}}[\mathbf{s}]). \quad (64)$$

Let this be the function mapping each $op \in \text{Op}_{\Sigma}(\vec{\mathbf{s}}, \mathbf{s})$ to

$$M_{\mathbb{T}}[op] \triangleq ([t_1], \dots, [t_n]) \mapsto [op \ t_1 \cdots t_n]. \quad (65)$$

The fact that the function in (65) is well-defined (that is, $[op \ t_1 \cdots t_n]$ only depends upon the equivalence classes of t_1, \dots, t_n) is an application of rule (SUBST); and by

virtue of (22), the function is supported by any finite set of atoms that supports op in Op_Σ . Finally, the same property (22) entails that the function $M_{\mathbb{T}}[\]$ in (64) is equivariant.

Lemma 9.3 *Given a term $t \in \Sigma_s(\Gamma)$ and a valuation $\rho \in M_{\mathbb{T}}[\Gamma]$, let $\sigma \in \Sigma(\Gamma, \emptyset)$ be a substitution that represents ρ in the sense that $\rho(x) = [\sigma(x)]$, for all $x \in \text{dom}(\Gamma)$. Then*

$$M_{\mathbb{T}}[t]\rho = [t\{\sigma\}] . \quad (66)$$

Proof. By induction on the structure of t . In the base case, when $t = \pi x$ is a suspension, using (57) we have $M_{\mathbb{T}}[t]\rho = M_{\mathbb{T}}[\pi x]\rho \triangleq \pi \cdot \rho(x) = \pi \cdot [\sigma(x)] = [\pi \cdot \sigma(x)] = [\pi * \sigma(x)] \triangleq [(\pi x)\{\sigma\}] = [t\{\sigma\}]$. The induction step, when t is of the form $op t_1 \cdots t_n$ follows from (65). \square

Theorem 9.4 (Ground Completeness) *$M_{\mathbb{T}}$ is a \mathbb{T} -algebra, that is, it satisfies all the axioms of \mathbb{T} (and hence by the Soundness Theorem 7.4, all the theorems of \mathbb{T}). Furthermore, for ground terms, a judgement $\emptyset \vdash \bar{a} \# t \approx t' : s$ is satisfied by $M_{\mathbb{T}}$ only if it is a theorem of \mathbb{T} .*

Proof. Suppose $\nabla \vdash \bar{a} \# t \approx t' : s$ is an axiom of \mathbb{T} with $\nabla = [\bar{a}_1 \# x_1 : s_1, \dots, \bar{a}_n \# x_n : s_n]$ say. Given any valuation $\rho \in M_{\mathbb{T}}[\nabla]$, for each $i = 1..n$ we have $\bar{a}_i \# \rho(x_i) \in M_{\mathbb{T}}[s_i]$. Choosing a representative term t_i for each equivalence class $\rho(x_i)$, by Lemma 9.2 we have $\emptyset \vdash_{\mathbb{T}} \bar{a}_i \# t_i : s_i$. Therefore the function σ mapping each x_i to t_i ($i = 1..n$) is a substitution in $\Sigma(\emptyset, \nabla)$ that satisfies $\emptyset \vdash_{\mathbb{T}} \sigma \approx \rho : \nabla$. Applying (SUBST) to this and $\nabla \vdash_{\mathbb{T}} \bar{a} \# t \approx t' : s$ gives $\emptyset \vdash_{\mathbb{T}} \bar{a} \# t\{\sigma\} \approx t'\{\sigma\} : s$ and hence $\bar{a} \# [t\{\sigma\}] = [t'\{\sigma\}] \in M_{\mathbb{T}}[s]$ by Lemma 9.2 again. Lemma 9.3 and the definition of σ gives $\bar{a} \# M_{\mathbb{T}}[t]\rho = M_{\mathbb{T}}[t'\rho]$. Since this holds for any valuation $\rho \in M_{\mathbb{T}}[\nabla]$, we have that $M_{\mathbb{T}}$ satisfies $\nabla \vdash \bar{a} \# t \approx t' : s$.

So $M_{\mathbb{T}}$ is a \mathbb{T} -algebra and it just remains to check that it satisfies a ground judgement $\emptyset \vdash \bar{a} \# t \approx t' : s$ only if that judgement is a theorem of \mathbb{T} . If it satisfies the judgement $\emptyset \vdash \bar{a} \# t \approx t' : s$, then $\bar{a} \# M_{\mathbb{T}}[t]\rho = M_{\mathbb{T}}[t']\rho$ holds for the unique valuation ρ in $M_{\mathbb{T}}[\emptyset]$. By Lemma 9.3 this means $\bar{a} \# [t\{\sigma\}] = [t'\{\sigma\}] \in M_{\mathbb{T}}[s]$ for σ the unique substitution in $\Sigma(\emptyset, \emptyset)$. Since this is necessarily the identity substitution for the empty sorting environment, from (27) we get $\bar{a} \# [t] = [t'] \in M_{\mathbb{T}}[s]$. Thus by Lemma 9.2, $\emptyset \vdash_{\mathbb{T}} \bar{a} \# t \approx t' : s$ holds. \square

Example 9.5 (λ -Terms Modulo $\alpha\beta\eta$ -Equivalence) If \mathbb{T} is the NEL-theory of Example 6.2, then $M_{\mathbb{T}}[\mathbf{tm}]$ is the usual nominal set of untyped λ -terms modulo $\alpha\beta\eta$ -equivalence, for which the freshness relation coincides with the “not a free variable of” relation. To see this, first recall from Example 4.3 that $\Sigma_{\mathbf{tm}}(\emptyset)$ is the set of syntax trees for λ -terms with variables V_a corresponding to atoms $a \in \text{Atom}_\nu$, with λ -abstraction terms written $L_a t$ and with application terms written $A t t'$. By virtue of the rules in Figure 5, the equivalence relation $\emptyset \vdash_{\mathbb{T}} t \approx t' : \mathbf{tm}$, by which $\Sigma_{\mathbf{tm}}(\emptyset)$ is quotiented to get $M_{\mathbb{T}}[\mathbf{tm}]$, is a congruence for λ -abstraction and application. It contains the relation of α -equivalence because (46) is a theorem of \mathbb{T} ; and it contains

η -equivalence because of axiom (η) in Figure 4. It also contains β -equivalence

$$[A(L_a t) t'] = [t]([t']/a) \in M_{\mathbb{T}}[\mathbf{tm}] \quad (67)$$

where $([t], [t']) \mapsto [t]([t']/a)$ is the usual notion of capture-avoiding substitution for λ -terms. Property (67) follows from axioms $(\beta-1)$ – $(\beta-4)$ ⁹ by examining the structure of t ; this is most easily proved as an application of the α -structural induction principle given in [25, Sect. 5.1], using the α -structurally recursive definition of capture-avoiding substitution given there. Altogether we have that if t and t' are $\alpha\beta\eta$ -equivalent syntax trees, then $[t] = [t'] \in M_{\mathbb{T}}[\mathbf{tm}]$. Furthermore, one can show by induction on the structure of t that if a is not free in it, then $a \# [t]$ in $M_{\mathbb{T}}[\mathbf{tm}]$ —the key point being that by virtue of axiom (α) and Lemma 9.2, we have $a \# [L_a t]$.

Conversely, one can show by induction on the derivation of a theorem $\nabla \vdash_{\mathbb{T}} \bar{a} \not\approx t \approx t' : s$ from the rules in Figure 5 that for any ground substitution $\emptyset \vdash_{\mathbb{T}} \sigma \approx \sigma : \nabla$ it is the case that the ground terms $t\{\sigma\}$ and $t'\{\sigma\}$ are $\alpha\beta\eta$ -equivalent syntax trees not containing \bar{a} in their set of free variables. The proof relies upon the fact that all ground instances of the axioms in Figure 4 have this property; we omit the details. In particular, if $\bar{a} \# [t] = [t'] \in M_{\mathbb{T}}[\mathbf{tm}]$, then (taking σ to be the identity) we get that t and t' are indeed $\alpha\beta\eta$ -equivalent syntax trees whose free variables are disjoint from \bar{a} .

10 Completeness

In this section we prove the main result of the paper, namely that for any NEL-theory the logical consequence relation (Definition 7.2) and the semantic consequence relation (Definition 6.3) coincide. For conventional algebra, completeness of equational logic for the usual interpretation of terms in algebras in the category of sets is a simple result: Given an equational theory, the collection of terms is quotiented by provable equality to get an algebra for which satisfaction coincides with theorem-hood. The role of variables in this term-algebra construction is to act as *indeterminates*—constants that do not occur in the signature of the original theory. Indeed, instead of working with all terms, it comes to the same thing if one extends the signature with countably many new constants and forms the term-algebra from ground-terms, as in the previous section. This interchangeability of variables and fresh constants in conventional equational logic is not so straightforward for nominal equational logic. In the interpretation of our language of terms in $\mathcal{N}om$, variables stand for indeterminate elements of nominal sets that therefore have indeterminate finite support; whereas constants (which, as usual, we identify with operation symbols of arity 0) have fixed finite supports. To prove the completeness theorem, we have to show that provability of a judgement involving variables can be recovered from provability of ground instantiations of the judgement, where the variables are

⁹ Axiom $(\beta-5)$ is not needed here since its ground instances are derivable from the other axioms.

replaced by constants with suitably fresh supports. To do so, first we introduce some notation for tuples of distinct atoms and their transpositions.

Definition 10.1 Given a tuple $\vec{\alpha} = \alpha_1, \dots, \alpha_m$ of sorts of atoms, define:

$$\text{Atom}_{\vec{\alpha}}^{(*)} \triangleq \left\{ (a_1, \dots, a_m) \in \text{Atom}_{\alpha_1} \times \dots \times \text{Atom}_{\alpha_m} \mid \bigwedge_{1 \leq i < j \leq m} a_i \neq a_j \right\}. \quad (68)$$

Permutations act on this set as for products of nominal sets of atoms (since permutation preserves distinctness of atoms):

$$(a_1, \dots, a_m) \in \text{Atom}_{\vec{\alpha}}^{(*)} \Rightarrow \pi \cdot (a_1, \dots, a_m) \triangleq (\pi(a_1), \dots, \pi(a_m)) \in \text{Atom}_{\vec{\alpha}}^{(*)}.$$

With this Perm-action $\text{Atom}_{\vec{\alpha}}^{(*)}$ is a nominal set, since clearly each $(a_1, \dots, a_m) \in \text{Atom}_{\vec{\alpha}}^{(*)}$ is supported by $\{a_1, \dots, a_m\}$; indeed this is the smallest such set of atoms, so that

$$\vec{a} \# (a_1, \dots, a_m) \Leftrightarrow a_1 \notin \vec{a} \wedge \dots \wedge a_m \notin \vec{a}. \quad (69)$$

Lemma 10.2 (Generalised Transposition) *For each tuple $\vec{\alpha} = \alpha_1, \dots, \alpha_m$ of sorts of atoms there is a morphism in \mathcal{Nom}*

$$\begin{aligned} \tau : \text{Atom}_{\vec{\alpha}}^{(*)} \times \text{Atom}_{\vec{\alpha}}^{(*)} &\longrightarrow \text{Perm} \\ (\vec{a}, \vec{a}') &\mapsto \tau_{\vec{a}, \vec{a}'} \end{aligned} \quad (70)$$

satisfying for each $\vec{a} = [a_1, \dots, a_m]$ and $\vec{a}' = [a'_1, \dots, a'_m]$ in $\text{Atom}_{\vec{\alpha}}^{(*)}$

- (i) $\tau_{\vec{a}, \vec{a}'}(a_i) = a'_i$ for $i = 1..m$;
- (ii) $\tau_{\vec{a}, \vec{a}'}(a) = a$, if $a \notin \{a_1, \dots, a_m, a'_1, \dots, a'_m\}$.
- (iii) If \vec{a} and \vec{a}' are disjoint lists, then $\tau_{\vec{a}', \vec{a}} = \tau_{\vec{a}, \vec{a}'}^{-1}$.

Proof. Given that we want (i) and (ii) to hold, to define $\tau_{\vec{a}, \vec{a}'}$ we just have to say how it acts on atoms in \vec{a}' that are not in \vec{a} , ensuring that we get a permutation. For each sort of atoms α , there is a sub-list b_1, \dots, b_k of members of \vec{a} of sort α not in \vec{a}' and a sub-list $b'_1, \dots, b'_{k'}$ of members of \vec{a}' of the same sort that are not in \vec{a} . Since $\vec{a}, \vec{a}' \in \text{Atom}_{\vec{\alpha}}^{(*)}$ it follows that $k = k'$. (Both equal the number of occurrences of α in \vec{a} minus the number of common members of \vec{a} and \vec{a}' of sort α .) So we can define $\tau_{\vec{a}, \vec{a}'}$ to map each b'_i to the corresponding b_i . In this way we get an element $\tau_{\vec{a}, \vec{a}'} \in \text{Perm}$ satisfying (i)–(iii); and it is not hard to see that the assignment $(\vec{a}, \vec{a}') \mapsto \tau_{\vec{a}, \vec{a}'}$ satisfies

$$\pi \tau_{\vec{a}, \vec{a}'} \pi^{-1} = \tau_{\pi \cdot \vec{a}, \pi \cdot \vec{a}'} \quad (\pi \in \text{Perm}, \vec{a}, \vec{a}' \in \text{Atom}_{\vec{\alpha}}^{(*)}). \quad (71)$$

Recall from Definition 2.5 that Perm regarded as a nominal set has Perm-action given by conjugation (13). Thus property (71) says that the function $(\vec{a}, \vec{a}') \mapsto \tau_{\vec{a}, \vec{a}'}$ is equivariant and hence is a morphism in \mathcal{Nom} . \square

Definition 10.3 (*Atom-Parameterised Constants*) Given a tuple $\vec{\alpha} = \alpha_1, \dots, \alpha_m$ of sorts of atoms and a sort $s \in \text{Sort}_\Sigma$ of a NEL-signature Σ , let $\Sigma[c_{\vec{\alpha}} : s]$ denote the NEL-signature obtained from Σ by adding new operation symbols $c_{\vec{\alpha}} : [] \rightarrow s$ as \vec{a} ranges over $\text{Atom}_{\vec{\alpha}}^{(*)}$. Thus $\Sigma[c_{\vec{\alpha}} : s]$ has the same set of sorts as Σ and has nominal set of operators given by the coproduct (Definition 2.9) $\text{Op}_\Sigma + \text{Atom}_{\vec{\alpha}}^{(*)}$, represented concretely as a union $\text{Op}_\Sigma \cup \{c_{\vec{a}} \mid \vec{a} \in \text{Atom}_{\vec{\alpha}}^{(*)}\}$, where we assume each operation symbol $c_{\vec{a}}$ is not already an element of Op_Σ . So the Perm-action on the new operation symbols satisfies $\pi \cdot c_{\vec{a}} = c_{\pi \cdot \vec{a}}$; and the type of each $c_{\vec{a}}$ is $[] \rightarrow s$.

If \mathbb{T} is a NEL-theory with underlying signature Σ , then $\mathbb{T}[c_{\vec{\alpha}} : s]$ denotes the theory with signature $\Sigma[c_{\vec{\alpha}} : s]$ and the same axioms as \mathbb{T} .¹⁰

We will use atom-parameterised constants c_{a_1, \dots, a_m} as indeterminates in the proof of the completeness theorem given below. Of course c_{a_1, \dots, a_m} is not as indeterminate as is a variable x : The former represents an element of a nominal set for which a support set is known, namely $\{a_1, \dots, a_m\}$; whereas the latter represents an element whose support only has to avoid at most finitely many atoms \bar{a} , supposing an assumption $\bar{a} \not\# x : s$ occurs in the current freshness context. Nevertheless, as the following proposition shows, one can recover a \mathbb{T} -theorem involving a variable from an instance of it obtained by substituting a new atom-parameterised constant for the variable. The proposition makes use of *single term substitution*: The term $t\{t'/x'\}$ is defined by recursion on the structure of t by:

$$\begin{aligned} (\text{op } t_1 \cdots t_n)\{t'/x'\} &\triangleq \text{op } t_1\{t'/x'\} \cdots t_n\{t'/x'\} \\ (\pi x)\{t'/x'\} &\triangleq \begin{cases} \pi x & \text{if } x \neq x' \\ \pi * t' & \text{if } x = x' \end{cases} \end{aligned} \quad (72)$$

where $\pi * t'$ is as in Figure 3. This is a special case of the kind of simultaneous substitution $t \mapsto t\{\sigma\}$ considered in Section 5, in the sense that if $t \in \Sigma_s(\Gamma, x' : s')$ (with $x' \notin \text{dom}(\Gamma)$) and $t' \in \Sigma_{s'}(\Gamma)$, then $t\{t'/x'\} = t\{\sigma\} \in \Sigma_s(\Gamma)$ where $\sigma \in \Sigma((\Gamma, x' : s'), \Gamma)$ is the substitution mapping x' to t' and mapping each $x \in \text{dom}(\Gamma)$ to itself.

Proposition 10.4 *Suppose $\nabla, \bar{a}_1 \not\# x_1 : s_1 \vdash \bar{a} \not\# t \approx t' : s$ is a well-formed judgement (with $x_1 \notin \text{dom}(\nabla)$) over the signature Σ of a NEL-theory \mathbb{T} . Given any finite set of atoms \bar{a}' supporting the judgement, that is, supporting $(\nabla, \bar{a}_1, \bar{a}, t, t')$, let $\bar{a}' = a'_1, \dots, a'_m$ be a list of the distinct atoms in $\bar{a}' - \bar{a}_1$ and suppose a'_i has sort α_i for $i = 1..m$. Let $\mathbb{T}[c_{\vec{\alpha}} : s_1]$ be the NEL-theory obtained from \mathbb{T} as in Definition 10.3. Then*

$$\nabla \vdash_{\mathbb{T}[c_{\vec{\alpha}} : s_1]} \bar{a} \not\# t\{c_{\vec{a}'}/x_1\} \approx t'\{c_{\vec{a}'}/x_1\} : s \Rightarrow \nabla, \bar{a}_1 \not\# x_1 : s_1 \vdash_{\mathbb{T}} \bar{a} \not\# t \approx t' : s. \quad (73)$$

To prove this proposition, we use an operation on terms that replaces atom-parametrised constants by variables: Given $t \in \Sigma[c_{\vec{\alpha}} : s_1]_s(\nabla)$, Figure 6 defines a

¹⁰Strictly speaking, we are relying upon the easily verified fact that $\Sigma_{s'}(\Gamma) \subseteq \Sigma[c_{\vec{\alpha}} : s]_{s'}(\Gamma)$ —so that well-formed judgements over Σ are also well-formed judgements over $\Sigma[c_{\vec{\alpha}} : s]$.

$$\begin{aligned}
(\pi x)\{c_{\vec{a}'} := x_1\} &\triangleq \pi x \\
(op t_1 \cdots t_n)\{c_{\vec{a}'} := x_1\} &\triangleq \begin{cases} \tau_{\vec{a}', \vec{a}} x_1 & \text{if } op = c_{\vec{a}} \\ op(t_1\{c_{\vec{a}'} := x_1\}) \cdots (t_n\{c_{\vec{a}'} := x_1\}) & \text{otherwise} \end{cases}
\end{aligned}$$

Fig. 6. Replacing Atom-Parameterised Constants by Variables

term $t\{c_{\vec{a}'} := x_1\} \in \Sigma_s(\nabla, x_1 : \mathbf{s}_1)$, obtained by replacing each $c_{\vec{a}}$ by the suspension $\tau_{\vec{a}', \vec{a}} x_1$, where $\tau_{\vec{a}', \vec{a}}$ is the generalised transposition from Lemma 10.2. The following lemmas give the properties of the operation in Figure 6 that we need.

- Lemma 10.5** (i) *If t does not contain any of the operation symbols in $\{c_{\vec{a}} \mid \vec{a} \in \text{Atom}_{\vec{\alpha}}^{(*)}\}$, then $t\{c_{\vec{a}'} := x_1\} = t$.*
- (ii) *If x_1 and x'_1 do not occur in t and if σ is the substitution that swaps x_1 and x'_1 , then $t\{c_{\vec{a}'} := x_1\}\{\sigma\} = t\{c_{\vec{a}'} := x'_1\}$.*
- (iii) $\pi \cdot (t\{c_{\vec{a}'} := x_1\}) = (\pi \cdot t)\{c_{\pi \cdot \vec{a}'} := x_1\}$.
- (iv) *If $a \# (t, \vec{a}')$, then $a \# t\{c_{\vec{a}'} := x_1\}$.*

Proof. Parts (i) and (ii) follow easily by induction on the structure of t . The same is true for part (iii), using (71). Part (iv) follows from part (iii) by Lemma 7.3(ii). \square

Lemma 10.6 *Suppose \mathbb{T} is a NEL-theory, with underlying signature Σ say, and that $\mathbb{T}[c_{\vec{\alpha}} : \mathbf{s}_1]$ is the theory obtained from it as in Definition 10.3. If ∇ is a freshness environment, \vec{a} is a finite set of atoms, $t, t' \in \Sigma[c_{\vec{\alpha}} : \mathbf{s}_1]_s(\nabla)$ and \vec{a}_1 supports (∇, \vec{a}, t, t') , then*

$$\begin{aligned}
\forall x_1 \in \text{Var}. \forall \vec{a}' \in \text{Atom}_{\vec{\alpha}}^{(*)}. x_1 \notin \text{dom}(\nabla) \wedge \vec{a}_1 \# \vec{a}' \Rightarrow \\
\nabla, \vec{a}_1 \not\# x_1 : \mathbf{s} \vdash_{\mathbb{T}} \vec{a} \not\# t\{c_{\vec{a}'} := x_1\} \approx t'\{c_{\vec{a}'} := x_1\} : \mathbf{s}
\end{aligned} \tag{74}$$

iff

$$\begin{aligned}
\exists x_1 \in \text{Var}. \exists \vec{a}' \in \text{Atom}_{\vec{\alpha}}^{(*)}. x_1 \notin \text{dom}(\nabla) \wedge \vec{a}_1 \# \vec{a}' \wedge \\
\nabla, \vec{a}_1 \not\# x_1 : \mathbf{s} \vdash_{\mathbb{T}} \vec{a} \not\# t\{c_{\vec{a}'} := x_1\} \approx t'\{c_{\vec{a}'} := x_1\} : \mathbf{s}.
\end{aligned} \tag{75}$$

Proof. This is an example of a “some/any” property of freshness in nominal sets—see [25, Lemma A.4(2)]. It is clear that (74) implies (75). For the converse, first note that the variable x_1 can be changed to any other $x'_1 \notin \text{dom}(\nabla)$ using (SUBST) and Lemma 10.5(ii). Similarly, $\vec{a}' \in \text{Atom}_{\vec{\alpha}}^{(*)}$ can be changed to any other $\vec{a}'' \in \text{Atom}_{\vec{\alpha}}^{(*)}$ by using part (iii) of that lemma together with the equivariance of logical consequence (Theorem 8.1) applied to the permutation $\tau_{\vec{a}', \vec{a}''}$ from Lemma 10.2. \square

Lemma 10.7 *Let $\mathbb{T}[c_{\vec{\alpha}} : \mathbf{s}_1]$ be as in the previous lemma.*

- (i) *Given a freshness environment ∇ , if $x_1 \notin \text{dom}(\nabla)$, $\pi \in \text{Perm}$, $t \in \Sigma[c_{\vec{\alpha}} :$*

$s_1]_s(\nabla^\cdot, x_1 : s_1)$, \bar{a}_1 supports (∇, π, t) and $\bar{a}' \in \text{Atom}_{\bar{\alpha}}^{(*)}$ with $\bar{a}_1 \# \bar{a}'$, then

$$\nabla, \bar{a}_1 \not\# x_1 : s_1 \vdash_{\mathbb{T}} (\pi * t)\{c_{\bar{a}'} := x_1\} \approx \pi * (t\{c_{\bar{a}'} := x_1\}) : s.$$

(ii) Given freshness environments ∇_1, ∇_2 , if $\sigma \in \Sigma[c_{\bar{\alpha}} : s_1][(\nabla_1)^\cdot, (\nabla_2)^\cdot]$, $x_1 \notin \text{dom}(\nabla_1) \cup \text{dom}(\nabla_2)$, $t \in \Sigma[c_{\bar{\alpha}} : s_1]_s((\nabla_1)^\cdot, x_1 : s_1)$, \bar{a}_1 supports $(\nabla_1, \nabla_2, \sigma, t)$ and $\bar{a}' \in \text{Atom}_{\bar{\alpha}}^{(*)}$ with $\bar{a}_1 \# \bar{a}'$, then

$$\nabla_2, \bar{a}_1 \not\# x_1 : s_1 \vdash_{\mathbb{T}} t\{\sigma\}\{c_{\bar{a}'} := x_1\} \approx t\{c_{\bar{a}'} := x_1\}\{\sigma\{c_{\bar{a}'} := x_1\}\} : s$$

where $\sigma\{c_{\bar{a}'} := x_1\} \in \Sigma(((\nabla_1)^\cdot, x_1 : s_1), ((\nabla_2)^\cdot, x_1 : s_1))$ is the substitution mapping each $x \in \text{dom}(\nabla_1)$ to $\sigma(x)\{c_{\bar{a}'} := x_1\}$ and mapping x_1 to itself.

Proof. Both parts are proved by induction on the structure of t . For (i) when $t = c_{\bar{a}}$, one uses $\nabla, \bar{a}_1 \not\# x_1 : s_1 \vdash_{\mathbb{T}} \tau_{\bar{a}', \pi \cdot \bar{a}} x_1 \approx \pi \tau_{\bar{a}', \bar{a}} x_1 : s_1$. This holds by (SUSP) and (WEAK) using

$$\begin{aligned} \bar{a}_1 &\supseteq \{a \mid \pi^{-1}(a) \neq a\} && \text{(since } \text{dom}(\pi^{-1}) = \text{dom}(\pi) \subseteq \bar{a}_1\text{)} \\ &= \{a \mid \pi \tau_{\bar{a}', \bar{a}} \pi^{-1}(a) \neq \pi \tau_{\bar{a}', \bar{a}}(a)\} \\ &= \{a \mid \tau_{\pi \cdot \bar{a}', \pi \cdot \bar{a}}(a) \neq \pi \tau_{\bar{a}', \bar{a}}(a)\} && \text{(by (71))} \\ &= \{a \mid \tau_{\bar{a}', \pi \cdot \bar{a}}(a) \neq \pi \tau_{\bar{a}', \bar{a}}(a)\} && \text{(since } \text{dom}(\pi) \subseteq \bar{a}_1 \# \bar{a}'\text{)}. \end{aligned}$$

For (ii) when t is a suspension, one uses part (i). \square

Lemma 10.8 Let $\mathbb{T}[c_{\bar{\alpha}} : s_1]$ be as in Lemma 10.6. If $\nabla \vdash_{\mathbb{T}[c_{\bar{\alpha}} : s_1]} \bar{a} \not\# t \approx t' : s$ and \bar{a}_1 supports (∇, \bar{a}, t, t') , then for any $x_1 \notin \text{dom}(\nabla)$ and any $\bar{a}' \in \text{Atom}_{\bar{\alpha}}^{(*)}$ with $\bar{a}_1 \# \bar{a}'$, it is the case that $\nabla, \bar{a}_1 \not\# x_1 : s \vdash_{\mathbb{T}} \bar{a} \not\# t\{c_{\bar{a}'} := x_1\} \approx t'\{c_{\bar{a}'} := x_1\} : s$.

Proof. This is proved by induction on the derivation of $\nabla \vdash_{\mathbb{T}[c_{\bar{\alpha}} : s_1]} \bar{a} \not\# t \approx t' : s$. Let IH be the set of well-formed judgements

$$\nabla \vdash \bar{a} \not\# t \approx t' : s \tag{76}$$

over the signature of $\mathbb{T}[c_{\bar{\alpha}} : s_1]$ such that for all \bar{a}_1 supporting (∇, \bar{a}, t, t') , (74) holds. Note that if (76) is an axiom of $\mathbb{T}[c_{\bar{\alpha}} : s_1]$, it is by definition an axiom of \mathbb{T} and hence does not contain any occurrences of the operation symbols $c_{\bar{a}}$. So by Lemma 10.5(i), $\nabla, \bar{a}_1 \not\# x_1 : s \vdash_{\mathbb{T}} \bar{a} \not\# t\{c_{\bar{a}'} := x_1\} \approx t'\{c_{\bar{a}'} := x_1\} : s$ is $\nabla, \bar{a}_1 \not\# x_1 : s \vdash_{\mathbb{T}} \bar{a} \not\# t \approx t' : s$, which is a theorem of \mathbb{T} by applying (WEAK) to the axiom (76). So IH contains the axioms of $\mathbb{T}[c_{\bar{\alpha}} : s_1]$ and to prove the lemma we just have to show that it is closed under each of the rules in Figure 5. Closure under rules (REFL), (SYMM), (WEAK), (ATM-INTRO), ($\not\#$ -EQUIVAR) and (SUSP) is straightforward. Closure under (TRANS), (SUBST) and (ATM-ELIM), where the support of the hypotheses of each rule is possibly bigger than that of its conclusion, requires some work. We give the argument for (TRANS). The proof for the other two rules is similar, using Lemma 10.7(ii) for the case of (SUBST).

To prove closure under (TRANS) , suppose

$$(\nabla \vdash \bar{a} \# t \approx t' : \mathbf{s}), (\nabla \vdash \bar{a}' \# t' \approx t'' : \mathbf{s}) \in IH . \quad (77)$$

To see that $(\nabla \vdash \bar{a} \cup \bar{a}' \# t \approx t'' : \mathbf{s}) \in IH$, for any \bar{a}_1 supporting $(\nabla, \bar{a} \cup \bar{a}', t, t'')$, by Lemma 10.6 it suffices to find some $x_1 \notin \text{dom}(\nabla)$ and some $\bar{a}' \in \text{Atom}_{\bar{\alpha}}^{(*)}$ with $\bar{a}_1 \# \bar{a}'$ and

$$\nabla, \bar{a}_1 \# x_1 : \mathbf{s} \vdash_{\mathbb{T}} \bar{a} \cup \bar{a}' \# t\{c_{\bar{a}'} := x_1\} \approx t''\{c_{\bar{a}'} := x_1\} : \mathbf{s} . \quad (78)$$

Pick some $x_1 \notin \text{dom}(\nabla)$, some $\bar{a}_2 \in \mathcal{P}_{\text{fin}}(\text{Atom})$ supporting (\bar{a}_1, t') and some $\bar{a}' \in \text{Atom}_{\bar{\alpha}}^{(*)}$ disjoint from \bar{a}_2 and hence also satisfying $\bar{a}_1 \# \bar{a}'$. From (77) we get $\nabla, \bar{a}_2 \# x_1 : \mathbf{s}_1 \vdash_{\mathbb{T}} \bar{a} \# t\{c_{\bar{a}'} := x_1\} \approx t'\{c_{\bar{a}'} := x_1\} : \mathbf{s}$ and $\nabla, \bar{a}_2 \# x_1 : \mathbf{s}_1 \vdash_{\mathbb{T}} \bar{a} \# t\{c_{\bar{a}'} := x_1\} \approx t''\{c_{\bar{a}'} := x_1\} : \mathbf{s}$. Noting that $(\nabla, \bar{a}_2 \# x_1 : \mathbf{s}_1) \leq (\nabla, \bar{a}_1 \# x_1 : \mathbf{s}_1)^{\#(\bar{a}_2 - \bar{a}_1)}$, by (TRANS) and (WEAK)

$$(\nabla, \bar{a}_1 \# x_1 : \mathbf{s}_1)^{\#(\bar{a}_2 - \bar{a}_1)} \vdash_{\mathbb{T}} \bar{a} \cup \bar{a}' \# t\{c_{\bar{a}'} := x_1\} \approx t''\{c_{\bar{a}'} := x_1\} : \mathbf{s} . \quad (79)$$

But $\bar{a} \cup \bar{a}' \subseteq \bar{a}_1$, so $(\bar{a}_2 - \bar{a}_1) \# \bar{a} \cup \bar{a}'$; and \bar{a}_1 supports ∇ , so $(\bar{a}_2 - \bar{a}_1) \# \nabla$. Also, since $(\bar{a}_2 - \bar{a}_1) \# (\bar{a}', t, t'')$, by Lemma 10.5(iv) we have $(\bar{a}_2 - \bar{a}_1) \# (t\{c_{\bar{a}'} := x_1\}, t''\{c_{\bar{a}'} := x_1\})$. Therefore we can apply (ATM-ELIM) to (79), obtaining (78), as required. \square

Proof of Proposition 10.4. First note that since substitution is equivariant (Corollary 5.3), by Lemma 7.3(ii) we have that \bar{a}' supports $t\{c_{\bar{a}'}/x_1\}$ and $t'\{c_{\bar{a}'}/x_1\}$. Therefore, picking any $x'_1 \notin \text{dom}(\nabla) \cup \{x_1\}$ and $\bar{a}'' \in \text{Atom}_{\bar{\alpha}}^{(*)}$ with $\bar{a}' \# \bar{a}''$, by Lemma 10.8 if $\nabla \vdash_{\mathbb{T}[c_{\bar{\alpha}}:\mathbf{s}_1]} \bar{a} \# t\{c_{\bar{a}'}/x_1\} \approx t'\{c_{\bar{a}'}/x_1\} : \mathbf{s}$ holds, then so does

$$\nabla, \bar{a}' \# x'_1 : \mathbf{s}_1 \vdash_{\mathbb{T}} \bar{a} \# t\{c_{\bar{a}'}/x_1\}\{c_{\bar{a}''} := x'_1\} \approx t'\{c_{\bar{a}'}/x_1\}\{c_{\bar{a}''} := x'_1\} : \mathbf{s} .$$

From this, using Lemma 10.7(ii), Lemma 10.5(i) and the definition in Figure 6, we get

$$\nabla, \bar{a}' \# x'_1 : \mathbf{s}_1 \vdash_{\mathbb{T}} \bar{a} \# t\{\tau_{\bar{a}'', \bar{a}'} x'_1/x_1\} \approx t'\{\tau_{\bar{a}'', \bar{a}'} x'_1/x_1\} : \mathbf{s} . \quad (80)$$

Note that from Lemma 10.2, $\tau_{\bar{a}'', \bar{a}'}$ fixes the atoms in \bar{a}_1 and maps \bar{a}'' onto \bar{a}' . Therefore, writing \bar{a}'' for the union of \bar{a}_1 with the atoms in \bar{a}'' , by $(\#-\text{EQUIVAR})$ and (WEAK) we have

$$\nabla, \bar{a}'' \# x_1 : \mathbf{s}_1 \vdash_{\mathbb{T}} \bar{a}' \# \tau_{\bar{a}'', \bar{a}'} x_1 : \mathbf{s}_1 . \quad (81)$$

Since \bar{a}' and \bar{a}'' are disjoint, by Lemma 10.2(iii) we have $\tau_{\bar{a}'', \bar{a}'} \tau_{\bar{a}', \bar{a}''} = \iota$. So by (SUBST) , (80) and (81) we get

$$\nabla, \bar{a}'' \# x'_1 : \mathbf{s}_1 \vdash_{\mathbb{T}} \bar{a} \# t \approx t' : \mathbf{s} .$$

Since $(\nabla, \bar{a}'' \# x'_1 : \mathbf{s}_1) \leq (\nabla, \bar{a}_1 \# x_1 : \mathbf{s}_1)^{\#(\bar{a}'' - \bar{a}_1)}$, we can apply (WEAK) to this to get

$$(\nabla, \bar{a}_1 \# x_1 : \mathbf{s}_1)^{\#(\bar{a}'' - \bar{a}_1)} \vdash_{\mathbb{T}} \bar{a} \# t \approx t' : \mathbf{s} . \quad (82)$$

By definition of \bar{a}'' , the set $\bar{a}'' - \bar{a}_1$ consists of the atoms in \bar{a}'' ; and we chose these to be disjoint from the set \bar{a}' supporting $(\nabla, \bar{a}_1, \bar{a}, t, t')$. So we can apply (ATM-ELIM) to (82) to get $\nabla, \bar{a}_1 \# x_1 : s_1 \vdash_{\mathbb{T}} \bar{a} \# t \approx t' : s$, as required. \square

Proposition 10.9 *With the same assumptions as in the statement of Proposition 10.4, it is the case that*

$$\nabla, \bar{a}_1 \# x_1 : s_1 \vdash_{\mathbb{T}} \bar{a} \# t \approx t' : s \Rightarrow \nabla \vdash_{\mathbb{T}[c_{\bar{a}}:s_1]} \bar{a} \# t\{c_{\bar{a}'}/x_1\} \approx t'\{c_{\bar{a}'}/x_1\} : s. \quad (83)$$

Proof. First note that for any two NEL-theories \mathbb{T} and \mathbb{T}' , if the signature and axioms of \mathbb{T} are contained in those of \mathbb{T}' , then the judgements over the signature of \mathbb{T} that are theorems of \mathbb{T}' contain the axioms of \mathbb{T} and are closed under the rules in Figure 5. Therefore

$$\nabla \vdash_{\mathbb{T}} \bar{a} \# t \approx t' : s \Rightarrow \nabla \vdash_{\mathbb{T}'} \bar{a} \# t \approx t' : s. \quad (84)$$

In particular, every $\mathbb{T}[c_{\bar{a}}:s_1]$ -algebra M is a \mathbb{T} -algebra when we forget the finitely supported functions assigned by the structure M to the operation symbols $c_{\bar{a}}$.

Now suppose

$$\nabla, \bar{a}_1 \# x_1 : s_1 \vdash_{\mathbb{T}} \bar{a} \# t \approx t' : s \quad (85)$$

holds. To prove (83) we have to show for each $\mathbb{T}[c_{\bar{a}}:s_1]$ -algebra M and valuation $\rho \in M[\nabla]$ that

$$\bar{a} \# M[t\{c_{\bar{a}'}/x_1\}]\rho = M[t'\{c_{\bar{a}'}/x_1\}]\rho \in M[s] \quad (86)$$

holds. As noted in (69), we have $\bar{a}_1 \# \bar{a}'$, since \bar{a}' is disjoint from \bar{a}_1 . Hence by Lemma 7.3(ii) applied to the equivariant function $M[-]$, we have $\bar{a}_1 \# M[c_{\bar{a}'}] \in M[s_1]$. Therefore the extended valuation $\rho[x_1 \mapsto M[c_{\bar{a}'}]]$ lies in $M[\nabla, \bar{a}_1 \# x_1 : s_1]$. So regarding M as a \mathbb{T} -algebra as above, from (85) we get

$$\bar{a}_1 \# M[t](\rho[x_1 \mapsto M[c_{\bar{a}'}]]) = M[t'](\rho[x_1 \mapsto M[c_{\bar{a}'}]]) \in M[s].$$

Now we can apply the compositionality property (31) of substitutions to obtain (86), as required for property (83). \square

Using Propositions 10.4 and 10.9, we can now prove the desired completeness result.

Theorem 10.10 (Completeness) *A judgement $\nabla \vdash \bar{a} \# t \approx t' : s$ is a theorem of a NEL-theory \mathbb{T} if it is satisfied by any \mathbb{T} -algebra in Nom :*

$$\nabla \vdash_{\mathbb{T}} \bar{a} \# t \approx t' : s \Rightarrow \nabla \vdash_{\mathbb{T}} \bar{a} \# t \approx t' : s. \quad (87)$$

Proof. We prove (87) by induction on the length of the freshness environment ∇ , for all \mathbb{T} , \bar{a} , t , t' and s simultaneously. The base case when the length is zero is a consequence of the Ground Completeness Theorem 9.4. For the induction step, if $\nabla, \bar{a}_1 \# x_1 : s_1 \vdash_{\mathbb{T}} \bar{a} \# t \approx t' : s$ (with $x_1 \notin \text{dom}(\nabla)$), then by (83) we get

$$\nabla \vdash_{\mathbb{T}[c_{\bar{a}}:s_1]} \bar{a} \# t\{c_{\bar{a}'}/x_1\} \approx t'\{c_{\bar{a}'}/x_1\} : s$$

and hence by induction hypothesis

$$\nabla \vdash_{\mathbb{T}[c_{\bar{\alpha}}:s_1]} \bar{a} \not\# t\{c_{\bar{a}'}/x_1\} \approx t'\{c_{\bar{a}'}/x_1\} : s.$$

Now we can apply (73) to deduce $\nabla, \bar{a}_1 \not\# x_1 : s_1 \vdash_{\mathbb{T}} \bar{a} \not\# t \approx t' : s$, as required. \square

11 Related Work

The first logical analyses of the permutative treatment of names, binding and α -equivalence were in terms of set theory [17] and higher-order logic [12]. They concentrated upon the logical properties of “support” (see Section 2). The complimentary “not-in-the-support-of” relation that we call *freshness*, turns out to be more relevant for the intended applications of the theory—such as structurally recursive/inductive properties of syntactical data modulo α -equivalence [25]. Indeed a large part of the theory depends on some simple, first order properties of freshness that were identified in [24]. The “nominal logic” introduced in that paper treats atoms like any other sort of data in many-sorted first-order logic; and thus statements about atoms are phrased in terms of variables ranging over a sort of atoms. An alternative is to include concrete atoms in the syntax of the logic. Since atoms are indeed atomic, that is, there are no compound expressions of atom sort, there may seem to be little difference between a concrete atom and a variable of sort atom. But there is an important difference: Two different variables x and y may, upon substitution, get replaced by the same entity; whereas two different atoms a and b , upon permutation, always remain distinct relative to each other. The perspective of the presheaf models of names and binding introduced by Fiore, Plotkin and Turi [9] is helpful here: Variables obey a renaming discipline involving all functions between finite sets, whereas atoms obey one involving only injective functions. In any case, the use of concrete atoms enabled Cheney and Gabbay to develop a Gentzen-style version of nominal logic with much better proof-theoretic properties than the original version—see [13,14,5]. A careful use of concrete atoms is also an important feature of the work on nominal unification [34] and logic programming [4,6]. For example, restricting “swapping terms” $(t\ t') \cdot t''$ to just be of the form $(a\ a') \cdot t''$ where a and a' are concrete atoms, allows one to restrict the syntactic occurrence of swapping to just the variables-with-suspended-permutations, πx , that were first introduced in the work on nominal unification [34] and that play an important role here.

The work most closely related to the results presented here is that on *nominal algebra* by Gabbay and Mathijssen [16,15]. This takes the notion of nominal signature and the language of nominal terms from [34,8] and develops an extension of equational logic closely related to the one presented here. The main difference is that their theory of nominal algebra treats freshness as a subsidiary notion: Their nominal algebraic theories can contain equational axioms (conditioned by freshness assumptions about variables), but not freshness axioms. Thus in the system of [15] freshness respects only definitional equality, but not logical equality. In other words,

the rule

$$\frac{\nabla \vdash a \# t : s \quad \nabla \vdash t \approx t' : s}{\nabla \vdash a \# t' : s}$$

is not valid for Gabbay and Mathijssen’s nominal algebra, whereas it is derivable in our nominal equational logic (from rule (TRANS) in Figure 5, bearing in mind that we regard “ $a \# t : s$ ” as an abbreviation for “ $\{a\} \# t \approx t : s$ ”, and “ $t \approx t' : s$ ” as an abbreviation for “ $\emptyset \# t \approx t' : s$ ”). So, unlike here, nominal algebra does not provide a complete axiomatisation of the semantic notion of freshness within nominal sets. On the other hand, in the Gabbay-Mathijssen approach, their intentional notion of freshness remains a simple, decidable property that is used as a side condition on equations. This seems quite natural from the point of view of term-rewriting; and Gabbay and Mathijssen give several interesting examples to support the claim that, from the point of view of logical theories, giving equality priority over freshness in this way is sufficiently expressive. Partly this expressivity is due to the fact that they make use of the atom-abstraction arities ($[\alpha]s$) and terms ($[a]t$) that are part of the notion of *nominal signature* [34, Definitions 2.1 and 2.3].

One consequence of our willingness to consider logical rather than just definitional freshness is that we are able to avoid the need for special syntax for atom-abstractions and can use arities and terms that are remarkably close to those of ordinary equational logic. For example, consider the NEL-theory for λ -calculus from Example 6.2, where the fact that λ -abstraction is a binder is axiomatised with a family of operation symbols L_a of type $\mathbf{tm} \rightarrow \mathbf{tm}$ (with a ranging over the set Atom_ν of atoms of sort ν) together with the freshness axiom (α). Whereas in nominal algebra, this is accomplished with a single operation symbol L of type $[\nu]\mathbf{tm} \rightarrow \mathbf{tm}$ and no axiom is needed (because of the logical properties of the atom-abstraction arity $[\nu]\mathbf{tm}$). Here, as well as wanting to completely axiomatise semantic freshness, we have striven for simplicity: The sorts, terms and rules of Nominal Equational Logic are as close to those of ordinary equational logic as we can make them. We believe this is a reasonable starting point for the study of “nominal universal algebra”. However, atom-abstraction arities could be added to NEL and probably should be, since making binding information part of a signature rather than part of a theory’s axioms is a good idea. Indeed, it would be interesting to try to add more complicated notions of binding specification, such as the one used by Pottier in his Caml system [26].

A second way in which the NEL-signatures introduced in this paper differ from the nominal signatures of [34,8] is that we have avoided the use of sorts of atom in arities. So for example in a nominal signature for λ -calculus, object-level variables are introduced via a single operation symbol $V : \nu \rightarrow \mathbf{tm}$ that is applied to nominal terms of atom sort ν (which may be atoms or variable suspensions); whereas in the corresponding NEL-signature (Example 3.1) one has a family of operation symbols $V_a : [] \rightarrow \mathbf{tm}$ indexed by atoms $a \in \text{Atom}_\nu$. Not regarding sorts of atom to be “data” sorts simplifies the notion of arity at the expense of increasing the number of operation symbols. Thus the notion of *finitely presented* NEL-signature is more complicated than the corresponding notion for nominal signatures; we do

not study this notion here. NEL’s simple notion of arity also means that we avoid the use of variables ranging over sorts of atoms; it remains to be seen whether this simplification really restricts expressive power.

Nominal equational logic is intended to capture the algebraic properties of freshness of names in general. From this perspective the running example of λ -calculus that we have used in this paper is rather special, since it concerns the use of named object-level variables and axioms for properties of substitution of terms for variables. This particular use of names is much studied in the literature, including from an algebraic perspective. For example Pigozzi and Salibra have studied a notion of *lambda abstraction algebra* [23,28] that should be compared with the NEL-theory of Example 6.2.

This paper is a first step towards developing a “nominal” version of universal algebra. We believe that the results presented here, and the work of Gabbay and Mathijssen mentioned above, show that the equational properties of freshness are a potentially useful tool for studying computer science languages involving names. However, much remains to be understood. In particular, the category of nominal sets has a “freshness tensor product”

$$X \otimes Y \triangleq \{(x, y) \in X \times Y \mid x \text{ and } y \text{ have disjoint support}\}$$

that is clearly relevant to the algebraic properties of freshness (cf. [29]). The form of generalised algebra that uses finitary enriched monads [18,27] has already been applied to presheaf categories [31] and nominal sets [10] to study algebraic properties of binding, name-restriction and renaming. The connection between our quite syntactic approach and this form of generalised algebra, and categorical algebra in general, needs to be addressed.

References

- [1] Abramsky, S., D. R. Ghica, A. S. Murovski, C.-H. L. Ong and I. D. B. Stark, *Nominal games and full abstraction for the nu-calculus*, in: *19th Annual Symposium on Logic in Computer Science* (2004), pp. 150–159.
- [2] Barendregt, H. P., “The Lambda Calculus: Its Syntax and Semantics,” North-Holland, 1984, revised edition.
- [3] Berger, U. and H. Schwichtenberg, *An inverse of the evaluation functional for typed λ -calculus*, in: *6th Annual Symposium on Logic in Computer Science* (1991), pp. 203–211.
- [4] Cheney, J., “Nominal Logic Programming,” Ph.D. thesis, Cornell University (2004).
- [5] Cheney, J., *Completeness and Herbrand theorems for nominal logic*, *Journal of Symbolic Logic* **71** (2006), pp. 299–320.
- [6] Cheney, J. and C. Urban, *α Prolog: A logic programming language with names, binding and α -equivalence*, in: B. B. Demoen and V. Lifschitz, editors, *Logic Programming, 20th International Conference, ICLP 2004, Saint-Malo, France, September 2004, Proceedings*, *Lecture Notes in Computer Science* **3132** (2004), pp. 269–283.
- [7] Dybjer, P. and A. Filinski, *Normalization and partial evaluation*, in: G. Barthe, P. Dybjer and J. Saraiva, editors, *Applied Semantics, Advanced Lectures*, *Lecture Notes in Computer Science*, Tutorial **2395**, Springer-Verlag, 2002 pp. 137–192, *International Summer School, APPSEM 2000, Caminha, Portugal, September 9–15, 2000*.

- [8] Fernandez, M., M. Gabbay and I. Mackie, *Nominal rewriting systems*, in: *Proc. 6th ACM-SIGPLAN Symposium on Principles and Practice of Declarative Programming (PPDP'04)* (2004), pp. 108–119.
- [9] 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.
- [10] Fiore, M. P. and S. Staton, *A congruence rule format for name-passing process calculi from mathematical operational semantics*, in: *21st Annual Symposium on Logic in Computer Science, Seattle, USA* (2006), pp. 49–58.
- [11] Gabbay, M. J., “A Theory of Inductive Definitions with α -Equivalence: Semantics, Implementation, Programming Language,” Ph.D. thesis, University of Cambridge (2000).
- [12] Gabbay, M. J., *FM-HOL, a higher-order theory of names*, in: F. Kamareddine, editor, *Workshop on Thirty Five years of Automath, Informal Proceedings*, Heriot-Watt University, Edinburgh, Scotland, 2002.
- [13] Gabbay, M. J., *Fresh logic*, *Journal of Logic and Computation* (200X), to appear.
- [14] Gabbay, M. J. and J. Cheney, *A sequent calculus for nominal logic*, in: *19th IEEE Symposium on Logic in Computer Science (LICS'04)* (2004), pp. 139–148.
- [15] Gabbay, M. J. and A. Mathijssen, *Nominal algebra* (2006), preprint.
- [16] Gabbay, M. J. and A. Mathijssen, *One-and-a-halfth-order logic*, in: *8th ACM-SIGPLAN International Symposium on Principles and Practice of Declarative Programming (PPDP'06), Venice, Italy* (2006), pp. 189–200.
- [17] 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.
- [18] Kelly, G. M. and A. J. Power, *Adjunctions whose counits are coequalizers, and presentations of finitary enriched monads*, *Journal of Pure and Applied Algebra* **89** (1993), pp. 163–179.
- [19] MacLane, S. and I. Moerdijk, “Sheaves in Geometry and Logic. A First Introduction to Topos Theory,” Springer-Verlag, New York, 1992.
- [20] Meinke, K. and J. V. Tucker, *Universal algebra*, in: S. Abramsky, D. M. Gabbay and T. S. E. Maibaum, editors, *Handbook of Logic in Computer Science, Volume 1*, Oxford University Press, 1992 pp. 189–411.
- [21] Milner, R., J. Parrow and D. Walker, *A calculus of mobile processes (parts I and II)*, *Information and Computation* **100** (1992), pp. 1–77.
- [22] Norrish, M., *Recursive function definition for types with binders*, in: *Theorem Proving in Higher Order Logics, 17th International Conference*, *Lecture Notes in Computer Science* **3223** (2004), pp. 241–256.
- [23] Pigozzi, D. and A. Salibra, *Lambda abstraction algebras: Representation theorems*, *Theoretical Computer Science* **140** (1995), pp. 5–52.
- [24] Pitts, A. M., *Nominal logic, a first order theory of names and binding*, *Information and Computation* **186** (2003), pp. 165–193.
- [25] Pitts, A. M., *Alpha-structural recursion and induction*, *Journal of the ACM* **53** (2006), pp. 459–506.
- [26] Pottier, F., *An overview of Cxaml*, in: *ACM SIGPLAN Workshop on ML (ML 2005), Tallinn, Estonia*, *Electronic Notes in Theoretical Computer Science* (2005), pp. 27–52.
- [27] Robinson, E. P., *Variations on algebra: monadicity and generalisations of equational theories*, *Formal Aspects of Computing* **13** (2002), pp. 308–326.
- [28] Salibra, A., *On the algebraic models of lambda calculus*, *Theoretical Computer Science* **249** (2000), pp. 197–240.
- [29] Schöpp, U. and I. D. B. Stark, *A dependent type theory with names and binding*, in: *Computer Science Logic, CSL04, Karpacz, Poland*, *Lecture notes in Computer Science* **3210** (2004), pp. 235–249.
- [30] Shinwell, M. R., A. M. Pitts and M. J. Gabbay, *FreshML: Programming with binders made simple*, in: *Eighth ACM SIGPLAN International Conference on Functional Programming (ICFP 2003), Uppsala, Sweden* (2003), pp. 263–274.
- [31] Stark, I. D. B., *Free-algebra models for the π -calculus*, in: *Foundations of Software Science and Computation Structures: Proceedings of FOSSACS 2005*, *Lecture Notes in Computer Science* **3441** (2005), pp. 155–169.

- [32] Stoughton, A., *Substitution revisited*, Theoretical Computer Science **59** (1988), pp. 317–325.
- [33] Urban, C. and S. Berghofer, *A recursion combinator for nominal datatypes implemented in Isabelle/HOL*, in: *3rd International Joint Conference on Automated Reasoning (IJCAR 2006), Seattle, USA*, Lecture Notes in Computer Science **4130** (2006), pp. 498–512.
- [34] Urban, C., A. M. Pitts and M. J. Gabbay, *Nominal unification*, Theoretical Computer Science **323** (2004), pp. 473–497.