Nominal (universal) algebra: equational logic with names and binding

Murdoch J. Gabbay* Aad Mathijssen†

May 27, 2009

Abstract

In informal mathematical discourse (such as the text of a paper on theoretical computer science) we often reason about equalities involving binding of object-variables. We find ourselves writing assertions involving *meta-variables* and *capture-avoidance constraints* on where object-variables can and cannot occur free. Formalising such assertions is problematic because the standard logical frameworks cannot express capture-avoidance constraints directly.

In this paper we make the case for *extending* the logic of equality with meta-variables and capture-avoidance constraints, to obtain 'nominal algebra'. We use nominal techniques that allow for a direct formalisation of meta-level assertions, while remaining close to informal practice. We investigate proof-theoretical properties, we provide a sound and complete semantics in nominal sets, and we compare and contrast our design decisions with other possibilities leading to similar systems.

*http://www.gabbay.org.uk
†A.H.J.Mathijssen@tue.nl

Contents

1 Introduction **3**

2 Syntax **7**
2.1 Terms and signatures . . . 7
2.2 Judgement forms, axioms and theories . . . 8

3 Derivations **10**
3.1 Permutation and substitution actions . . . 10
3.2 Inference rules . . . 11
3.3 Proof-theoretical results . . . 16

4 Denotations **21**
4.1 Nominal sets . . . 22
4.2 Interpretations, models and validity . . . 24
4.3 Free term models . . . 27
4.4 Completeness for equality derivations . . . 29
4.5 Completeness for freshness . . . 32

5 Design alternatives **33**
5.1 N-abs: nominal algebra without atoms-abstraction . . . 33
5.2 N+feq: nominal algebra with stronger freshness derivation rules . . . 36

6 Conclusions **39**
6.1 Related work . . . 39
6.2 Future work . . . 42

A Equivariance **46**

1 Introduction

Perhaps *equality* is the simplest possible judgement form. Informal specification of logic and computation often involves equalities with *binding*, subject to conditions about *freshness*. For example:

λ-calculus:	$\lambda x.(ex) = e$	if $x \notin fv(e)$
First-order logic:	$\forall x.(\phi \supset \psi) = \phi \supset \forall x.\psi$	if $x \notin fv(\phi)$
π-calculus:	$\nu x.(P \mid Q) = P \mid \nu x.Q$	if $x \notin fv(P)$
Process algebra with data:	$\sum x.p = p$	if $x \notin fv(p)$

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

Substitution:	$(\zeta y.e')[x \mapsto e] = \zeta y.(e'[x \mapsto e])$	if $y \notin fv(e)$

Here $fv(e)$ denotes the free variables of e. It is not hard to extend this short list with many more examples.[1]

In the equalities between expressions above there are *two* levels of variable:

- x and y range over variable symbols. These are sometimes called *object-level* variables.
- e, e', ϕ, ψ, P, Q and p range over expressions. These are sometimes called *meta-level* variables.

These equalities are subject to freshness side-conditions $x \notin fv(e)$, placing conditions between the object-level variable denoted by x, and the syntax of the expressions denoted by e. These freshness side-conditions make these equalities something other than 'just equalities'.

A straightforward way to formalise these meta-level properties is to enrich to the logic of equality with meta-variables. In a setting with binders, we have to face the following problems:

- When is an object-variable fresh for a meta-variable? We only know this when the meta-variable is instantiated to a concrete expression (not mentioning meta-variables).
- What is a suitable representation of α-conversion of object-variables in the setting with meta-variables?
- In the presence of meta-variables, substitution of expressions for object-variables becomes non-trivial: what does it mean when we try to substitute an expression for a variable in a meta-variable?

A number of different solutions have been proposed for these problems. The state of the art solution is based on the use of some kind of (typed) λ-calculus. For example a typical higher-order logic [vB01, Mei92] has a base type of 'individuals', and then uses higher types of functions to operate on individuals (or other functions). Other methods include the use of combinators [CF58, Bar84] and cylindric techniques [HMT85, LS04].

Unfortunately, none of these solutions allows for a *natural* formalisation of the kind of schematic specification of informal discourse discussed above. They feature a two-level structure of object-level and meta-level variables and freshness side-conditions.

A consequence of this is that the formalisation of schemas of theorems and proofs are not always a matter of simple refinement, but sometimes requires a fair amount of emulation. In De Bruijn's words [dB91]:

> "I think that in formalizing mathematics, and in particular in preparing mathematics for justification, it is usually elegant as well as efficient to do everything in the *natural* way."

There have been solutions to the problem of binding and meta-variables that embrace the difference between object- and meta-variables. One of these solutions uses so-called nominal

[1] Process algebras with data are discussed in [Gro97, Lut02, GMR+08].

techniques [GP01], which has two levels of variable called *atoms* and *unknowns* in [UPG04] — and a built-in notion of freshness of atoms with respect to unknowns.

Nominal techniques have been applied to unification [UPG04], term rewriting [FG07] and first-order logic [Pit03]. The application was purely to represent and reason about formal *syntax* with meta-variables.

In this paper we explore **nominal algebra**, an application of nominal techniques to represent algebraic reasoning in the presence of atoms, unknowns, and freshness side-conditions. In this way, and consistent with de Bruijn's philosophy, we hope to provide a natural and yet fully formal representation of the kind of algebraic-style reasoning seen in informal mathematical practice.

Nominal algebra derivation rules are in Figures 1 and 2 (see Section 3 for full details; the relevant definition is Definition 3.10). These rules were first presented and studied in a workshop paper [GM06b] and longer technical report [GM06c]; the material was expanded on in a conference paper [GM07] and in the second author's thesis [Mat07]. This paper is the journal version of [GM07].

It turns out that in nominal algebra, informal equivalences can be represented as axioms almost symbol-for-symbol. For example the equalities between expressions from the beginning of the Introduction are represented by:

λ-calculus:	$a\#X \vdash$	$\lambda[a](Xa) = X$
First-order logic:	$a\#X \vdash$	$\forall[a](X \supset Y) = X \supset \forall[a]Y$
π-calculus:	$a\#X \vdash$	$\nu[a](X \mid Y) = X \mid \nu[a]Y$
Process algebra with data:	$a\#X \vdash$	$\sum[a]X = X$
Substitution:	$b\#X \vdash$	$(\zeta[b]Y)[a \mapsto X] = \zeta[b](Y[a \mapsto X])$

The equalities here are between *nominal terms*, which are the formal syntax which we will use to represent the expressions above (the formal definition is in Definition 2.4, later in this paper). Here a and b are distinct *atoms* representing object-level variables; X and Y are *unknowns* representing meta-level variables; $[a]t$ is an atoms-abstraction of an atom a. Each equality is equipped with a *freshness condition* of the form $a\#X$ that guarantees that X can only be instantiated to a term for which a is fresh. The rest of this paper makes this formal.

We develop the proof theory of nominal algebra and show that it supports the following key features of meta-level reasoning:

- instantiation of meta-variables, by means of *capturing* substitution of expressions for meta-variables;
- α*-renaming* of object-variables and *capture-avoiding* substitution of expressions for object-variables in the presence of meta-variables;
- generation of *fresh* object-variables inside a derivation.

Furthermore, we provide a denotation in so-called *nominal sets*. Nominal sets were introduced by Gabbay and Pitts in [GP01].[2] They have proved to be an effective model for syntax with names and binding (see for example [Pit03]). In fact, nominal sets have inspired the design of nominal terms, which form the basis of nominal algebra. For this reason, nominal sets permit a natural semantic interpretation of atoms a, abstractions $[a]t$, and freshness $a\#t$, which are not conveniently definable on 'ordinary' sets.

Overview. We introduce the syntax of nominal algebra in Section 2. In Section 3 we provide a notion of derivation of freshness and equality with the ability to impose axioms, and we provide examples and proof-theoretical results. We provide a denotation of nominal algebra in terms of nominal sets in Section 4, and we show how derivability is complete for this denotation. In Section 5 we discuss variations on the nominal algebra theme; a simplified variant N-abs which drops atoms-abstraction without losing expressivity, and an enriched variant N+feq with two extra

[2] In [GP01], nominal sets are called FM-sets, named after the Fraenkel and Mostowski who devised a permutation model of set theory in order to prove the independence of the axiom of choice from the other axioms of Zermelo-Fraenkel set theory with atoms [Bru96].

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

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

Figure 1: Derivation rules for freshness

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

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

$$\frac{\nabla^{\pi}\sigma}{t^{\pi}\sigma=u^{\pi}\sigma}\,(\mathbf{ax}_{\nabla\vdash\mathbf{t=u}}) \qquad \frac{a\#t \quad b\#t}{(a\ b)\cdot t=t}\,(\mathbf{perm})$$

$$\frac{\begin{array}{c}[a\#X_1,\ldots,a\#X_n] \quad \Delta\\ \vdots\\ t=u\end{array}}{t=u}\,(\mathbf{fr}) \quad (n\geq 1,\ a\notin t,u,\Delta)$$

Figure 2: Derivation rules for equality

rules for freshness derivations. In each case we justify the design decisions we made in Sections 2 and 3, using a combination of arguments on derivations and models. Finally in the Conclusions (Section 6) we discuss the development of nominal algebra so far, and related and future work.

Comments on symbols for equality. 'Equality' plays a central rôle in this paper, and (just like the Eskimos have more than one word for snow) we have more than one symbol for equality. All our usages are standard, but we take a moment to give an overview:

- The symbol $=$ is used as part of the formal syntax of nominal algebra equality judgement form '$\Delta \vdash t = u$'. This is derivable equality between terms and is defined in Subsection 2.2.
- The symbol $=$ is also used in the meta-level discourse of this paper, to express equality of elements. This is denotational equality. When we write '$\pi = \pi'$', we mean 'π and π' denote the same permutation'.
- This is a standard overloading of the symbol $=$ but it creates a problem; how to unambiguously indicate equality of formal syntax (*syntactic identity*); the notion 'denote the same syntax-tree'. We do not want to write $t = u$ for syntactic identity because it might not always be instantly obvious whether we intend 't and u represent the same syntax-tree' or 'we can derive that the denotations of the syntax-trees represented by t and u, are equal'. We therefore write $\equiv$ for syntactic identity (Definition 2.8) between terms.
- In Examples 2.6 and 2.15 an object-level equality term-former $\approx$ is mentioned. As far as this paper is concerned, $\approx$ is just a binary term-former and $t \approx u$ is just a term. It *does* intuitively represent object-level equality and Example 2.15 mentions the relevant axioms (**Esubst**) and (**Erefl**); this is studied in full detail elsewhere [GM06d, GM08c].

Comments on the word 'algebra'. The word 'algebra' in the title is used differently in different parts of the literature: there are (at least) 'algebra' in the sense of solving equations like $x^2 + bx + c = 0$; 'algebra' as the dedicated study of structures like groups rings and fields; 'process algebras' in the theory of concurrency; 'algebras and co-algebras' in category theory; and 'universal algebra' the study of the logic of equality.

It seems prudent to be clear about what is meant by algebra in *this* paper. We mean algebra in the sense of universal algebra as presented for example in [BS81]. *Algebra* for us is a logic of equality, whose basic judgement form is 't is equal to u', bells and whistles notwithstanding. The denotation is such that a derivable equality is interpreted by denotational identity — so if we can prove that t is equal to u, then the denotations of t and u must be identical in all models. A *(univeral) algebraic theory* is a collection of axioms asserting equalities between terms. A *model* of an algebraic theory is a set[3] with functions on it interpreting the signature of the theory, such that the equalities of the theory are valid identities.

From that point of view, our goal in this paper is to present a minimal extension of universal algebra whose axioms, derivations, and denotations provide built-in support for the kinds of equalities involving names, binding, and freshness side-conditions that we considered above.

We will usually write 'nominal universal algebra' as just 'nominal algebra'.

Comments on foundations. Mathematical papers are usually written using an informal set theory which, if their authors were pressed to be more formal, would turn out to be Zermelo-Fraenkel set theory (with or without the Axiom of Choice). Part of the idea of what is now called nominal techniques, which is described in [GP01], is to base our foundation on Zermelo-Fraenkel set theory with atoms (ZFA). In particular, we use a foundation in ZFA in this paper.

It is the atoms of ZFA sets that we use in our nominal terms syntax for nominal algebra, and also in the nominal sets denotation which follows. Why should we be concerned about this? Can we not model variable symbols using, say, ordinals?

[3] In this paper it will be a nominal set, but here we do not care about the difference.

Treating variable symbols as atomic, we can prove theorems about their behaviour which have specifically to do with them being atomic, and so having no internal structure. These theorems are not true of ordinals because they do have internal structure, although the theorems will still be provable, on a case-by-case basis, if the constructions involved do not use that internal structure — that is, if the constructions treat the ordinals *as if* they were atomic. We find it simpler to just take variables to *be* atomic in the first place.

One such theorem, which we will use in our proofs, is *meta-level equivariance*. Informally this states that validity and provability are invariant under permuting atoms; since atoms have no internal structure, we can permute them. The definition, discussion, and proof are in Appendix A.

2 Syntax

We now develop *nominal terms* [UPG04] as a formal syntax in which the example expressions in the Introduction may be represented.

2.1 Terms and signatures

Definition 2.1. Fix disjoint countably infinite collections of **atoms**, **unknowns**, and **term-formers**.

$a, b, c, \ldots$ will range permutatively over atoms, $X, Y, Z, \ldots$ will range permutatively over unknowns, and $\mathsf{f}, \mathsf{g}, \ldots$ will range permutatively over term-formers. Here *permutative* means that distinct meta-variables represent distinct elements, so that for example 'a and b' means 'two distinct atoms', 'X and Y and Z' means 'three distinct unknowns', and 'f and g' means 'two distinct term-formers'.

We also assume that to each term-former f is associated some unique **arity** n which is a non-negative number; we write $\mathsf{f} : n$ to indicate that f has arity n. It is convenient to assume that there are infinitely many term-formers of each arity.

For the purpose of α-conversion, we need to be able to rename atoms. We use *permutations* of atoms instead of substitutions of atoms for atoms because permutations have better mathematical properties; most notably, permutations are capture-avoiding by definition (see the Introduction of [GP01] and [Pit03] for a detailed exposition).

Definition 2.2. Let $\mathbb{A} = \{a, b, c, \ldots\}$. A **(finite) permutation** π of atoms is a total bijection $\mathbb{A} \to \mathbb{A}$ with **finite support**, meaning that for some finite set of atoms (which may be empty) $\pi(a) \neq a$, *but* for all atoms not in that set, $\pi(a) = a$.

Finite support is a mathematical notion of 'most': π is a bijection on atoms such that $\pi(a) = a$ for *most* a.

We introduce some notation for permutations that we will need later on.

Definition 2.3. Write id for the **identity permutation** such that $id(a) = a$ always. Write $\pi \circ \pi'$ for **functional composition** and write π^{-1} for **inverse**. This makes permutations into a group — write $\mathbb{P}$ for the set of all permutations. Write $(a\ b)$ for the permutation that **swaps** a and b, i.e. the permutation that maps a to b, b to a and all other c to themselves.

Using the above ingredients we can form nominal terms.

Definition 2.4. (Nominal) terms t, u, v are inductively defined by:

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

We call $[a]t$ an **atoms-abstraction**; it represents the '$x.e$' or '$x.\phi$' part of expressions such as '$\lambda x.e$' or '$\forall x.\phi$'. We call $\pi \cdot X$ a **moderated unknown**. We write $id \cdot X$ just as X, for brevity.

In Section 3 we will see that in $\pi \cdot X$ the unknown X will get substituted for a term and then π will permute the atoms in that term. This notion is grounded in semantics [GP01] and permits a succinct treatment of α-renaming atoms (see Section 3.3.3 and [UPG04]).

Definition 2.5. A **signature** Σ is a set of term-formers with their arities.

Example 2.6. Here are some example signatures:

- $\{\mathsf{lam} : 1, \mathsf{app} : 2\}$ is a signature for the λ-calculus [GM09a, GM09b].

  We show how the terms in this signature relate to 'ordinary' λ-calculus expressions. For convenience identify atoms with *variable symbols*, then the syntax of the untyped λ-calculus is inductively defined by:

  $$e \quad ::= \quad a \;\mid\; \lambda a.e \;\mid\; ee$$

  The map $\text{-}'$ from untyped λ-calculus expressions to nominal terms is inductively defined by:

  $$a' = a \qquad (\lambda a.e)' = \mathsf{lam}([a](e')) \qquad (e_1 e_2)' = \mathsf{app}(e_1', e_2')$$

  We generally sugar $\mathsf{lam}([a]t)$ to $\lambda[a]t$ and $\mathsf{app}(t, u)$ to tu.

- $\{\bot : 0, \supset : 2, \forall : 1, \approx : 2\}$ is a signature for first-order logic with equality (the symbol for equality inside the logic is $\approx$) [GM06d, GM08c].

  We sugar $\bot()$ to $\bot$, $\supset(t, u)$ to $t \supset u$, $\forall([a]t)$ to $\forall[a]t$ and $\approx(t, u)$ to $t \approx u$.

A little more on this is in Example 2.15.

Remark 2.7. Consistent with previous work on nominal rewriting [FG07] we do not impose an a priori sort system on terms. Although this allows us to write 'silly terms' like $\lambda(tu)$ and $\forall(t \approx u)$, it simplifies the presentation, and the results specialise easily to the more specific cases.

Definition 2.8. Write $t \equiv u$ for **syntactic identity** of terms.

Note that if $\pi = \pi'$ then $\pi \cdot X \equiv \pi' \cdot X$, since permutations are represented by themselves. There is no quotient by abstraction so for example $[a]a \not\equiv [b]b$.

Definition 2.9. We define $a \in t$ inductively by:

$$a \in a \qquad a \in [a]t \qquad \frac{a \in t}{a \in [b]t} \qquad \frac{(\pi(a) \neq a)}{a \in \pi \cdot X} \qquad \frac{a \in t_i \ (1 \leq i \leq n)}{a \in \mathsf{f}(t_1, \ldots, t_n)}$$

We read $a \in t$ as 'a **occurs in (the syntax of)** t'. We write $a \notin t$ when $a \in t$ does not hold, and we read this as 'a **does not occur in** t'.

We define $X \in t$ inductively by:

$$X \in \pi \cdot X \qquad \frac{X \in t}{X \in [b]t} \qquad \frac{X \in t_i \ (1 \leq i \leq n)}{X \in \mathsf{f}(t_1, \ldots, t_n)}$$

We read $X \in t$ as 'X **occurs in (the syntax of)** t'. We write $X \notin t$ when $X \in t$ does not hold, and we read this as 'X **does not occur in** t'.

2.2 Judgement forms, axioms and theories

Definition 2.10. A **freshness** is a pair $a\#t$ of an atom a and a term t. Call a freshness $a\#X$ (so $t \equiv X$) **primitive**. Write Δ and ∇ for finite sets of *primitive* freshnesses and call them **freshness contexts**.

Recall that the atom a corresponds with a variable symbol x, and the unknown X corresponds with a meta-variable e or ϕ. Intuitively, $a\#X$ corresponds with 'x is not a free variable symbol in g/ϕ'.

Definition 2.11. We may drop set brackets in sets of freshnesses, e.g. writing $a\#t, b\#u$ for $\{a\#t, b\#u\}$. Also, we may write $a\#t, u$ for $a\#t, a\#u$. Furthermore, for any set of freshnesses S write $a \in S$ when a occurs anywhere in S, and $X \in S$ when X occurs anywhere in S.

Definition 2.12. An **equality** is a pair $t = u$ where t and u are terms.

Equalities will be used to state that two terms are *provably equal*.

Definition 2.13. Nominal algebra has two judgement forms:

- A **freshness judgement form** $\Delta \vdash a\#t$ is a pair of a freshness context Δ and a freshness $a\#t$.
- An **equality judgement form** $\Delta \vdash t = u$ is a pair of a freshness context Δ and an equality $t = u$.

We may write $\emptyset \vdash a\#t$ as $\vdash a\#t$, and $\emptyset \vdash t = u$ as $\vdash t = u$.

Definition 2.14. A **theory** $\mathsf{T} = (\Sigma, Ax)$ is a pair of a signature Σ and a possibly infinite set of *equality* judgement forms Ax in that signature; we call them the **axioms**.

We do not allow freshness judgements as axioms, but see Subsection 5.2.

Example 2.15. Here are some nominal algebra theories — we make these axioms 'do' something in Section 3 when we discuss derivations, which can use axioms; so for the moment these axioms have just a suggestive status illustrating use of the nominal terms' syntax:

- CORE is a family of theories with no axioms; there is one such theory for each signature Σ. It has built-in α-equivalence, so for example $\lambda[a]a$ is equal to $\lambda[b]b$.[4]

  Theory CORE is discussed in Subsection 3.3.3.

- SUB gives substitution term-former sub the correct behaviour in theories LAM and FOL. It is a family of theories, one for each signature Σ that includes sub, with axioms

$$\begin{array}{rlll}
(\mathbf{var}{\mapsto}) & & \vdash & a[a \mapsto X] = X \\
(\#{\mapsto}) & a\#Y & \vdash & Y[a \mapsto X] = Y \\
(\mathsf{f}{\mapsto}) & & \vdash & \mathsf{f}(Y_1,\ldots,Y_n)[a \mapsto X] = \mathsf{f}(Y_1[a \mapsto X],\ldots,Y_n[a \mapsto X]) \\
(\mathbf{abs}{\mapsto}) & b\#X & \vdash & ([b]Y)[a \mapsto X] = [b](Y[a \mapsto X]) \\
(\mathbf{id}{\mapsto}) & & \vdash & Y[b \mapsto b] = Y \\
(\eta{\mapsto}) & b\#X & \vdash & [a]\mathsf{sub}(X,a) = X
\end{array}$$

  For each term-former f (including sub), there is one axiom $(\mathsf{f}{\mapsto})$. Note the use of freshness side-conditions to manage the relationship between atoms and unknowns.

  We study this in [GM06a, GM08a].

- LAM has signature $\{\mathsf{lam} : 1, \mathsf{app} : 2, \mathsf{sub} : 2\}$, the axioms of SUB for this signature, and two axioms

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

  where we sugar $\mathsf{sub}([a]t, u)$ to $t[a \mapsto u]$.

  We study this in [GM09a, GM08b, GM09b].

- FOL has signature $\{\bot : 0, \supset : 2, \forall : 1, \approx : 2, \mathsf{sub} : 2\}$, the axioms of SUB for this signature, and seven axioms

$$\begin{array}{lll}
(\mathbf{MP}) & \vdash \top \supset X & = X \\
(\mathbf{Mer}) & \vdash ((((X \supset Y) \supset (\neg Z \supset \neg W)) \supset Z) \supset V) & \\
 & \qquad \supset ((V \supset X) \supset (W \supset X)) & = \top \\
(\mathbf{Qinst}) & \vdash \forall[a]X \supset X[a \mapsto Y] & = \top \\
(\mathbf{Qdist}) & \vdash \forall[a](X \wedge Y) \Leftrightarrow \forall[a]X \wedge \forall[a]Y & = \top \\
(\mathbf{Qextr}) \quad a\#X & \vdash \forall[a](X \supset Y) \Leftrightarrow X \supset \forall[a]Y & = \top \\
(\mathbf{Esubst}) & \vdash Z \approx Y \wedge X[a \mapsto Y] \supset X[a \mapsto Z] & = \top \\
(\mathbf{Erefl}) & \vdash X \approx X & = \top
\end{array}$$

[4] α-equivalence is expressed as a derivation rule: the (**perm**) rule from Figure 2. The (**perm**) rule is discussed in detail in Subsection 3.2.2.

Here we use standard classical logic sugar for $\top$, $\neg$, $\wedge$ and $\Leftrightarrow$.

Axioms (**MP**) and (**Mer**) characterise propositional logic; axioms (**Qinst**), (**Qdist**) and (**Qextr**) characterise quantification; and axioms (**Esubst**) and (**Erefl**) characterise object-level equality.

We study this in [GM06d, GM08c].

Similar developments for other systems with binding, such as process algebra with data [Gro97, Lut02, GMR+08] and the π-calculus [Par01] from the Introduction should also be possible.

3 Derivations

In this section we define notions of derivation which represent freshness assumptions on meta-variables (Figure 1), and permit axioms involving abstraction that are conditioned by freshness assumptions (Figure 2), just like we do in informal practice.

3.1 Permutation and substitution actions

Before we introduce our calculus, we elaborate on two important prequisites for the instantiation of axioms; we need to be able to *permute atoms in terms*, and *substitute terms for unknowns* in a capturing way.

Definition 3.1. The **(object-level) permutation action** $\pi \cdot t$ on terms is inductively defined by:

$$\pi \cdot a \equiv \pi(a) \qquad \pi \cdot (\pi' \cdot X) \equiv (\pi \circ \pi') \cdot X \qquad \pi \cdot [a]t \equiv [\pi(a)](\pi \cdot t)$$
$$\pi \cdot \mathsf{f}(t_1, \ldots, t_n) \equiv \mathsf{f}(\pi \cdot t_1, \ldots, \pi \cdot t_n)$$

Intuitively, π propagates through the structure of t until it reaches an atom or a moderated unknown.

Composition and identity of permutations extend to terms:

Lemma 3.2. $(\pi \circ \pi') \cdot t \equiv \pi \cdot (\pi' \cdot t)$ *and* $id \cdot t \equiv t$.

Proof. By induction on the structure of t, using Definition 3.1. □

Substitution is the mechanism by which unknowns become terms, and this is necessary in algebra so that we can define instances of axioms:

Definition 3.3. A **substitution (on unknowns)** σ is a function from unknowns to terms.

Definition 3.4. The **(meta-level) substitution action** $t\sigma$ on terms is inductively defined by:

$$a\sigma \equiv a \qquad (\pi \cdot X)\sigma \equiv \pi \cdot \sigma(X) \qquad ([a]t)\sigma \equiv [a](t\sigma)$$
$$\mathsf{f}(t_1, \ldots, t_n)\sigma \equiv \mathsf{f}(t_1\sigma, \ldots, t_n\sigma)$$

Intuitively, σ propagates through the structure of t until it reaches an atom or a moderated unknown. σ acts on the X in $\pi \cdot X$; then π acts on $\sigma(X)$. We suggest reading $\pi \cdot X$ as 'permute as π in whatever X becomes'. For example suppose $\sigma(X) \equiv a$; then

$$((a\ b) \cdot X)\sigma \equiv (a\ b) \cdot b \equiv a.$$

Substitution does not avoid capture. For example,

$$([a]X)\sigma \equiv [a]a.$$

This corresponds with what happens when we write 'instantiate - to x in $\lambda x.$-'; we obtain $\lambda x.x$.

Lemma 3.5. $\pi \cdot t\sigma \equiv (\pi \cdot t)\sigma$.

Proof. By induction on the structure of t, using Definitions 3.1 and 3.4. The case of $t \equiv \pi' \cdot X$ uses Lemma 3.2. □

Another permutation action is useful.

Definition 3.6. The **meta-level permutation action** t^π on terms t is inductively defined by:

$$a^\pi \equiv \pi(a) \qquad (\pi' \cdot X)^\pi \equiv \pi \circ \pi' \circ \pi^{-1} \cdot X \qquad ([a]t)^\pi \equiv [\pi(a)]t^\pi$$
$$\mathsf{f}(t_1, \ldots, t_n)^\pi \equiv \mathsf{f}({t_1}^\pi, \ldots, {t_n}^\pi)$$

Also for this permutation action, composition and identity of permutations extend to terms.

Lemma 3.7. $t^{\pi \circ \pi'} \equiv t^{\pi' \pi}$ *and* $t^{id} \equiv t$.

Proof. By induction on the structure of t, using Definition 3.6. □

In the presence of substitution, the two permutation actions $\pi \cdot t$ and t^π are interdefinable; however, sometimes one is more natural than the other, we shall point out how later (Remark 3.14).

Lemma 3.8. *Given a term t, let σ be a substitution that maps each $X \in t$ to $\pi \cdot X$, and let σ' be a substitution that maps each $X \in t$ to $\pi^{-1} \cdot X$.*
Then $\pi \cdot t \equiv t^\pi\sigma$ *and* $t^\pi \equiv (\pi \cdot t)\sigma'$.

Proof. By induction on the structure of t, using Definitions 3.1, 3.6 and 3.4 of $\pi \cdot t$, t^π and $t\sigma$. The only interesting case is when $t \equiv \pi' \cdot X$. Then we need to show $\pi \cdot (\pi' \cdot X) \equiv (\pi' \cdot X)^\pi\sigma$. Since $(\pi' \cdot X)^\pi\sigma \equiv (\pi \circ \pi' \circ \pi^{-1}) \cdot (\pi \cdot X)$ by Definitions 3.6 and 3.4, it suffices to show

$$\pi \cdot (\pi' \cdot X) \equiv (\pi \circ \pi' \circ \pi^{-1}) \cdot (\pi \cdot X).$$

This follows using Definition 3.1 and the fact that $\pi \circ \pi' = \pi \circ \pi' \circ \pi^{-1} \circ \pi$.
The proof of $(\pi' \cdot X)^\pi \equiv (\pi \cdot (\pi' \cdot X))\sigma'$ follows similar lines. □

Definition 3.9. We extend notation for t^π, $\pi \cdot t$ and $t\sigma$ to freshness contexts Δ as follows:

$$\begin{array}{rcl} \Delta^\pi & \text{is} & \{\pi(a)\#X \mid a\#X \in \Delta\} \\ \pi \cdot \Delta & \text{is} & \{\pi(a)\#\pi \cdot X \mid a\#X \in \Delta\} \\ \Delta\sigma & \text{is} & \{a\#\sigma(X) \mid a\#X \in \Delta\} \end{array}$$

Note that Δ^π is a freshness context, but $\pi \cdot \Delta$ and $\Delta\sigma$ need not be.

3.2 Inference rules

For the reader's convenience we recall some conventions:

- $a, b, c, \ldots$ range permutatively over atoms.
- $X, Y, Z, \ldots$ range permutatively over unknowns.
- $\mathsf{f}, \mathsf{g}, \ldots$ range permutatively over term-formers.
- π, π' range over permutations (not permutatively; it may be that $\pi = \pi'$).
- $t, t_1, \ldots, t_n, u, \ldots$ range over terms (not permutatively).

Definition 3.10. Define a notion of **derivability** by the natural deduction rules in Figures 1 and 2.

We will use the following notation:

- We write $\Delta \vdash a\#t$ when a derivation of $a\#t$ exists using the elements of Δ as assumptions. We say '$\Delta \vdash a\#t$ is **derivable**', or just '$\Delta \vdash a\#t$'.

  We write $\Delta \nvdash a\#t$ when $\Delta \vdash a\#t$ is not derivable.

  Note that the rules for freshness are syntax-directed, so if t is in a signature Σ then every term in the derivation of $\Delta \vdash a\#t$ must also be in Σ.

  When S is a set of freshnesses we write $\Delta \vdash S$ to mean '$\Delta \vdash a\#t$ for each $a\#t \in S$' as a convenient shorthand.

- Suppose that $\mathsf{T} = (\Sigma, Ax)$ is a theory. We write $\Delta \vdash_{\mathsf{T}} t = u$ when $t = u$ may be derived such that:
  - The derivation uses (at most) assumptions from Δ.
  - For each instance of $(\mathbf{ax}_{\nabla \vdash \mathbf{t=u}})$ used in the derivation, $(\nabla \vdash t = u) \in Ax$.
  - The derivation mentions only terms in the signature Σ.

  We say '$\Delta \vdash_{\mathsf{T}} t = u$ is **derivable (in T)**', or just '$\Delta \vdash_{\mathsf{T}} t = u$'.

  We write $\Delta \nvdash_{\mathsf{T}} t = u$ when $\Delta \vdash_{\mathsf{T}} t = u$ is not derivable.

In Figure 2, the rules (**refl**), (**symm**) and (**tran**) ensure that equality is an equivalence relation, and the rules (**cong**[]) and (**congf**) ensure that it is a congruence. The $(\mathbf{ax}_{\nabla \vdash \mathbf{t=u}})$ rule instantiates axioms in derivations and is discussed in Subsection 3.2.1. The (**perm**) rule expresses α-conversion, and is discussed in Subsection 3.2.2. Finally, (**fr**) (read bottom-up) introduces 'a fresh atom a' into a derivation. Here, the square brackets denote *discharge* in the sense of natural deduction (as in implication introduction) [Pra65] of these extra assumptions $a\#X_1, \ldots, a\#X_n$. (**fr**) is discussed in Subsection 3.2.3.

Remark 3.11 (Natural deduction vs. sequent derivation presentation)**.** We define our notion of entailment in natural deduction style; '$t = u$' has no meaning on its own but it can form part of the syntax of a natural deduction derivation proving that the sequent $\Delta \vdash t = u$ is derivable.

The reader who prefers to specify their notions of entailment based entirely on sequents, can rephrase the rules in Figures 1 and 2 in sequent form (a sequent version of (**fr**), the only even slightly non-trivial rule to translate, is given in Subsection 3.2.3).

We prefer the natural deduction presentation for its simplicity and compactness (we do not have to write '$\Delta \vdash$' everywhere); example derivations follow below. This is just a matter of presentation. However, we should draw the reader's attention to one subtlety of natural deduction derivation; $a\#X \vdash a\#X$ is derivable, and the natural deduction derivation that proves that this sequent is derivable is the (sublimely concise) tree

$$a\#X.$$

This derivation has one assumption, and one conclusion, and they are the same. Thus, in natural deduction there is no need for an explicit 'identity' rule, as is needed in a sequent presentation.

Note finally that $a\#X$ is actually shorthand for $a\#id \cdot X$, though we will elide the id (for example, in $(\#\mathbf{X})$).

Remark 3.12. Not very deeply hidden in the rules in Figure 1 is the standard definition of 'not in the free variables of': $(\#\mathbf{ab})$ corresponds with '$x \notin fv(y)$'; $(\#[]\mathbf{a})$ corresponds with '$x \notin fv(\lambda x.t)$' (or '$x \notin fv(\forall x.\phi)$', and so on); we leave the interpretation of $(\#[]\mathbf{b})$ and $(\#\mathbf{f})$ to the reader.

Freshness $\#$ cannot only formalise a known definition, it must also generalise to account for unknowns X; these are intended to represent 'unknown terms', so $(\#\mathbf{X})$ expresses that $\pi(a)$ is fresh for $\pi \cdot X$ provided that we have *assumed* that a is fresh for X (perhaps this is one point where the use of *permutations*, rather than atom-for-atom substitutions, is key). $(\#\mathbf{X})$ excludes the identity permutation id to guarantee a nice computational property, that the algorithm naturally obtained by reading the freshness derivation rules bottom-up, must terminate.

Note that because freshness models the informal judgement '$x \notin fv(t)$', its derivability does not depend on the theory T; in other words, the judgement-form for freshness is $\Delta \vdash a\#t$ and not $\Delta \vdash_{\mathsf{T}} a\#t$. More on this in Subsection 5.2.

Example 3.13 (Freshness derivations)**.** In the signature of theory LAM (Example 2.15) we can derive:

$$\dfrac{\dfrac{\dfrac{}{a\#b}\,(\#\mathbf{ab})}{a\#[b]b}\,(\#[]\mathbf{b})}{a\#\lambda[b]b}\,(\#\mathsf{f}) \qquad \dfrac{a\#X \quad \dfrac{\dfrac{}{a\#[a]Y}\,(\#[]\mathbf{a})}{a\#\lambda[a]Y}\,(\#\mathsf{f})}{a\#X(\lambda[a]Y)}\,(\#\mathsf{f})$$

The following are *non*-derivable freshnesses in this signature:

$$\nvdash a\#a \qquad \nvdash a\#X(\lambda[a]Y) \qquad \nvdash a\#(\lambda[a]b)a$$

In the signature of theory FOL (Example 2.15), derivable freshnesses are:

$$\vdash a\#\forall[a]P \qquad a\#T \vdash a\#(a \approx a)[a \mapsto T] \qquad a\#X \vdash b\#(b\ a)\cdot X.$$

Non-derivable freshnesses in this signature are:

$$\nvdash a\#\forall[b]P \qquad \nvdash a\#(a \approx a)[a \mapsto T] \qquad a\#X \nvdash a\#(b\ a)\cdot X.$$

Examples of derivability of equality can be found in the rest of this section.

3.2.1 The $(\mathbf{ax}_{\nabla\vdash\mathbf{t}=\mathbf{u}})$ rule: instantiating axioms

$(\mathbf{ax}_{\nabla\vdash\mathbf{t}=\mathbf{u}})$ allows us to permutatively rename atoms and to instantiate unknowns. This gives the effect that atoms in axioms can be understood to range over *any* (distinct) atoms, and unknowns can be understood to range over *any* terms (this idea goes back to the use of nominal rewrite rules in nominal rewriting [FG07]).

Consider a simple axiom: $\vdash X = Y$. Here, we intend X and Y to be instantiated; so we do not also need the axiom $\vdash X = Z$ or $\vdash X = \mathsf{2}$ (suppose a term-former $\mathsf{2}$), because these can all be obtained from $\vdash X = Y$ by instantiation. Thus, in the case of an axiom a single piece of syntax with variables (unknowns) represents the infinite collection of all of its instantiation instances. This is standard.

Now consider an axiom $\vdash a = X$. Here, we intend X to be instantiated — and we intend a to be permuted. In this way it is not necessary to consider a scheme of axioms $\vdash a = X$, $\vdash b = X$, $\vdash c = X$ (one for every possible atom); the permutation π in the axiom rule gives us this power from a single axiom mentioning, say, a.

In this sense, unknowns in axioms are variables, and atoms in axioms are *also* variables; whereas unknowns intuitively range by *instantiation* over all terms, atoms intuitively range *permutatively* over all atoms (it is possible to endow a substitution structure on atoms but it is not necessary to 'hard-wire' this; we use axioms, e.g. SUB from Example 2.15).

We will now consider some examples of the use of axioms.

$$\dfrac{}{(\lambda[b]a)b = a[b \mapsto b]}\,(\mathbf{ax}_\beta) \qquad \dfrac{}{(\lambda[b]b)a = b[b \mapsto a]}\,(\mathbf{ax}_\beta) \qquad \dfrac{\dfrac{}{b[a \mapsto a] = b}\,(\mathbf{id}{\mapsto})}{(\lambda[a]b)a = b}\,(\beta)$$

are valid derivations in theory LAM (Example 2.15). Note that substitution of terms for unknowns does not avoid capture, reflecting the intuition that they represent meta-variables.

The use of the $(\mathbf{ax}_{\nabla\vdash\mathbf{t}=\mathbf{u}})$ rule can introduce freshness proof obligations:

$$\dfrac{\dfrac{}{a\#b}\,(\#\mathbf{ab})}{\lambda[a](ba) = b}\,(\mathbf{ax}_\eta) \qquad \dfrac{a\#a}{\lambda[a](aa) = a}\,(\mathbf{ax}_\eta)$$

The left derivation is valid but the right one is not, because $a\#a$ is not derivable.

Note that instantiation of axioms (**abs↦**) and (**id↦**) from theory SUB (Example 2.15), which both mention distinct atoms a and b, can never identify these atoms. For example,

$$\cancel{\frac{c\#X}{([c]Y)[c\mapsto X]=[c](Y[c\mapsto X])}\,(\mathbf{ax_{abs\mapsto}})}$$

is not a valid instance of (**abs↦**) since permutations are bijective: there is no π such that both $\pi(a)=c$ and $\pi(b)=c$.

In informal practice, derivations are often presented in a calculational style, e.g. the sequence of equalities

$$\lambda x.(((\lambda x.y)x)x) =_\beta \lambda x.(yx) =_\eta y$$

represents that two expressions can be related by (reading from left to right) first applying β-conversion followed by η-conversion. This is fully formally represented by the following derivation:

$$\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{}{(\lambda[a]b)a=b[a\mapsto a]}\,(\mathbf{ax}_\beta) \quad \dfrac{\dfrac{}{a\#b}\,(\#\mathbf{ab})}{b[a\mapsto a]=b}\,(\mathbf{ax}_{\#\mapsto})}{(\lambda[a]b)a=b}\,(\mathbf{tran})}{((\lambda[a]b)a)a=ba}\,(\mathbf{congf})}{[a](((\lambda[a]b)a)a)=[a](ba)}\,(\mathbf{cong[]})}{\lambda[a](((\lambda[a]b)a)a)=\lambda[a](ba)}\,(\mathbf{congf}) \quad \dfrac{\dfrac{}{a\#b}\,(\#\mathbf{ab})}{\lambda[a](ba)=b}\,(\mathbf{ax}_\eta)}{\lambda[a](((\lambda[a]b)a)a)=b}\,(\mathbf{tran})$$

Reading the derivation bottom-up, the instance of (**tran**) in the conclusion introduces two equalities that correspond directly to the two equations in the above sequence of equalities. The derivation of the right equality takes care of η-equality: it instantiates the (η) axiom, and shows that the freshness side-condition is satisfied. The derivation of the left equality takes care of the β-equality: it instantiates the (β) axiom, which introduces a substitution, and we show how this substitution is applied.

Remark 3.14. Another version of the ($\mathbf{ax}_{\nabla\vdash\mathbf{t=u}}$) rule is possible, which uses the object-level action $\pi\cdot t$ instead of the meta-level action t^π:

$$\frac{\pi\cdot\nabla\sigma}{\pi\cdot t\sigma=\pi\cdot u\sigma}\,(\mathbf{ax}'_{\nabla\vdash\mathbf{t=u}}).$$

However in this case, atoms in substitution σ are renamed according to π. For example, $\vdash [b]a=[a]a$ is derivable using ($\mathbf{ax}_{\mathbf{[a]X=[b]X}}$) where we choose $\pi=(b\ a)$ and $\sigma(X)\equiv a$ and $\sigma(Y)\equiv id\cdot Y$ for all other Y. It is also derivable using ($\mathbf{ax}'_{\mathbf{[a]X=[b]X}}$), but we must choose $\pi=(b\ a)$ and $\sigma(X)\equiv b$ and $\sigma(Y)\equiv id\cdot Y$ for all other Y. We find ($\mathbf{ax}'_{\nabla\vdash\mathbf{t=u}}$) slightly less natural than ($\mathbf{ax}_{\nabla\vdash\mathbf{t=u}}$).

Note that we do not allow *freshness* axioms ($\mathbf{ax}_{\nabla\vdash\mathbf{a\#t}}$). The effect of freshness axioms can be obtained by an *equality* axiom $\Delta, b\#X_1,\ldots,b\#X_n\vdash(b\ a)\cdot t=t$ where b is fresh and $X_1,\ldots,X_n$ are the unknowns mentioned in Δ and t. More on this in Subsections 4.5 and 5.2.

3.2.2 The (perm) rule: α-equivalence

The (**perm**) rule concisely expresses α-equivalence. To illustrate this, the following derivations are valid in CORE (the theory with no axioms):

$$\dfrac{\dfrac{\dfrac{}{a\#b}\,(\#\mathbf{ab})}{a\#[b]b}\,(\#[]\mathbf{b}) \quad \dfrac{}{b\#[b]b}\,(\#[]\mathbf{a})}{[a]a=[b]b}\,(\mathbf{perm}) \qquad \dfrac{\dfrac{a\#X}{a\#[b]X}\,(\#[]\mathbf{b}) \quad \dfrac{}{b\#[b]X}\,(\#[]\mathbf{a})}{[a](b\ a)\cdot X=[b]X}\,(\mathbf{perm})$$

So $\vdash_{\mathsf{CORE}} [a]a = [b]b$ and $a\#X \vdash_{\mathsf{CORE}} [a](b\ a)\cdot X = [b]X$. To see that the instances of (**perm**) are valid, we note that $[a]a \equiv (b\ a)\cdot[b]b$ and $[a](b\ a)\cdot X \equiv (b\ a)\cdot[b]X$.

As another example, we show how we use the (**perm**) rule to rename a bound variable in a (representation of a) λ-calculus expression. Consider the following derivation in the λ-calculus:

$$(\lambda x.xx)(\lambda x.\lambda y.xy) =_\beta (\lambda x.\lambda y.xy)(\lambda x.\lambda y.xy) =_\beta \lambda y.(\lambda x.\lambda y.xy)y =_\beta \lambda y.\lambda z.yz.$$

In the last step the bound variable x is implicitly renamed during β-reduction to avoid capture. Nominal algebra makes this explicit. We present the nominal algebra derivation of this last step as a calculation:

$$\begin{array}{rcll}
\lambda[b]\underline{(\lambda[a]\lambda[b]ab)b} & = & \lambda[b]\underline{(\lambda[b]ab)[a\mapsto b]} & (\beta)\\
& = & \lambda[b]\lambda\underline{([b]ab)[a\mapsto b]} & (\mathsf{f}{\mapsto})\\
& = & \lambda[b]\lambda\underline{([c]ac)[a\mapsto b]} & (\mathbf{perm})\ (\vdash b\#[c]ac,\ \vdash c\#[c]ac)\\
& = & \lambda[b]\lambda[c]\underline{(ac)[a\mapsto b]} & (\mathbf{abs}{\mapsto})\ (\vdash c\#b)\\
& = & \lambda[b]\lambda[c]\underline{(a[a\mapsto b])}(c[a\mapsto b]) & (\mathsf{f}{\mapsto})\\
& = & \lambda[b]\lambda[c]b\underline{(c[a\mapsto b])} & (\mathbf{var}{\mapsto})\\
& = & \lambda[b]\lambda[c]bc & (\#{\mapsto})\ (\vdash a\#c)
\end{array}$$

In each step of the calculation, we have indicated in the hint which derivation rule is applied and which freshness constraints it had to satisfy (if any), and we have underlined the subterm on which the axiom is applied. It is not hard to reconstruct the derivation tree using (**cong**[]), (**congf**) and (**tran**).

As a final example we show that we can rename an atom which is substituted for, using the explicit substitution term-former sub from theory SUB (Example 2.15):

Lemma 3.15. $b\#X \vdash_{\mathsf{CORE}} X[a\mapsto T] = ((b\ a)\cdot X)[b\mapsto T]$

Proof. De-sugaring, we derive $\mathsf{sub}([a]X,T) = \mathsf{sub}([b](b\ a)\cdot X,T)$ from $b\#X$:

$$\dfrac{\dfrac{\dfrac{}{a\#[a]X}\,(\#[]\mathbf{a}) \qquad \dfrac{b\#X}{b\#[a]X}\,(\#[]\mathbf{b})}{[b](b\ a)\cdot X = [a]X}\,(\mathbf{perm})}{[a]X = [b](b\ a)\cdot X}\,(\mathbf{symm})}{\mathsf{sub}([a]X,T) = \mathsf{sub}([b](b\ a)\cdot X,T)}\,(\mathbf{congf})$$

□

In Subsection 3.3.3 we will show that derivability in CORE precisely corresponds to α-equivalence on nominal terms, in the sense of nominal unification [UPG04, Figure 2] and nominal rewriting [FG07, p.13].

Remark 3.16. (**perm**) admits a representation as an axiom:

$$a\#X, b\#X \vdash (a\ b)\cdot X = X.$$

It is not hard to see that with this axiom, (**perm**) becomes an instance of (**ax**). Our intended semantics is in nominal sets so (**perm**) is mandatory for soundness (Theorem 4.24) to hold, and accordingly we have built it into the derivation system.

3.2.3 The (fr) rule: introducing fresh atoms

(**fr**) allows us to introduce a fresh atom into the derivation. We may wish to do this, for example, in order that we can α-convert an abstracted atom to be 'fresh'. In the absence of unknowns X we can 'just rename'. In the presence of unknowns X, there only exists an a fresh for X if we assume $a\#X$; within the proof-theory of nominal algebra, (**fr**) is designed to guarantee we have an infinite supply.

In a sequent style presentation of nominal algebra, (**fr**) would be

$$\frac{\Delta, a\#X_1, \ldots, a\#X_n \vdash t = u}{\Delta \vdash t = u} \quad (n \geq 1, a \notin t, u, \Delta).$$

To prove that (**fr**) gives us extra deductive power we consider a theory C with one axiom $a\#X \vdash X = a$.

Lemma 3.17. *We can derive $\vdash_{\mathsf{C}} X = Y$ with (**fr**), and not without it.*

Proof. First, we derive $\vdash_{\mathsf{C}} X = Y$ with (**fr**):

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

In the derivation above the superscript ^{1} is an annotation which associates the instance of the rule (**fr**) with the assumptions it discharges in the derivation, as is standard notation in natural deduction.

To show that it is impossible to derive $\vdash_{\mathsf{C}} X = Y$ without (**fr**) we show the more general property that for all t, if $t \not\equiv X$ then $X = t$ and $t = X$ are not derivable without the use of (**fr**).

We proceed by contradiction. Let Π be a smallest derivation tree of $X = t$ or $t = X$ where $t \not\equiv X$. By the structure of the rules, Π cannot conclude with (**refl**), (**cong**[]) or (**congf**). Also by the structure of the rules, Π cannot conclude with ($\mathbf{ax_{a\#X\vdash X=a}}$) or (**perm**), since they require a freshness condition on X.

We now consider (**symm**) and (**tran**). We only consider the case that Π is a derivation of $X = t$ case; the case that Π is a derivation of $t = X$ is similar. Suppose Π concludes in:

- (**symm**). Then $X = t$ is derived from $t = X$, and we have a smaller derivation tree of $X = t$ or $t = X$, which is a contradiction.
- (**tran**). Then $X = t$ is derived from $X = u$ and $u = t$. There are two cases depending on whether $u \equiv X$ or $u \not\equiv X$, but in either case we obtain a smaller derivation tree and a contradiction.

The result follows. □

Remark 3.18. The (**fr**) rule can be simulated by a number of steps using the following more compact rule:

$$\dfrac{\begin{array}{c}[a\#X] \quad \Delta \\ \vdots \\ t = u\end{array}}{t = u}\,(\mathbf{fr'}) \quad (a \notin t, u)$$

We use (**fr**) because it allows us to express 'let a be a fresh atom' in a single, atomic reasoning step. However, whether to prefer (**fr′**) and (**fr**) seems mostly a matter of taste.

3.3 Proof-theoretical results

We provide a number of proof-theoretical results for freshness and equality that will be used throughout this paper.

3.3.1 Equivariance

Definition 3.19. We naturally extend notation for t^π and Δ^π to theories: given a theory $\mathsf{T} = (\Sigma, Ax)$, write T^π for (Σ, Ax^π) such that $\nabla^\pi \vdash t^\pi = u^\pi \in Ax^\pi$ if and only if $\nabla \vdash t = u \in Ax$.

Lemma 3.20. *If* $\Delta \vdash_{\mathsf{T}} t = u$ *then* $\Delta \vdash_{\mathsf{T}^\pi} t = u$.

Proof. By induction on derivations. The only non-trivial case is $(\mathbf{ax}_{\nabla \vdash \mathbf{t}=\mathbf{u}})$, where after applying the inductive hypothesis we need to show that

$$\Delta \vdash_{\mathsf{T}^\pi} \nabla^{\pi'}\sigma \quad \text{implies} \quad \Delta \vdash_{\mathsf{T}^\pi} t^{\pi'}\sigma = u^{\pi'}\sigma.$$

By Lemma 3.7, it is equivalent to show that

$$\Delta \vdash_{\mathsf{T}^\pi} \nabla^{\pi^{\pi'} \circ \pi^{-1}}\sigma \quad \text{implies} \quad \Delta \vdash_{\mathsf{T}^\pi} t^{\pi^{\pi'} \circ \pi^{-1}}\sigma = u^{\pi^{\pi'} \circ \pi^{-1}}\sigma.$$

This follows by $(\mathbf{ax}_{\nabla^\pi \vdash \mathbf{t}^\pi = \mathbf{u}^\pi})$ taking permutation $\pi' \circ \pi^{-1}$ and substitution σ. □

Theorem 3.21 states that we may permute atoms in freshnesses and equality at the *meta-level*. This property is one way in which nominal algebra reflects internally the properties of atoms, as expressed by Theorem A.4, which give 'nominal techniques' much of their character:

Theorem 3.21 (Meta-level equivariance)**.** *For any* π*:*

1. *if* $\Delta \vdash a\#t$ *then* $\Delta^\pi \vdash \pi(a)\#t^\pi$*;*
2. *if* $\Delta \vdash_{\mathsf{T}} t = u$ *then* $\Delta^\pi \vdash_{\mathsf{T}} t^\pi = u^\pi$.

Proof. For the second case suppose

$$\Delta \vdash_{\mathsf{T}} t = u.$$

By equivariance (Theorem A.4) also

$$\Delta^\pi \vdash_{\mathsf{T}^\pi} t^\pi = u^\pi.$$

By Lemma 3.20 we obtain

$$\Delta^\pi \vdash_{\mathsf{T}^{\pi^{\pi^{-1}}}} t^\pi = u^\pi.$$

Using Lemma 3.7 we easily show that $\mathsf{T}^{\pi^{\pi^{-1}}}$ is syntactically identical to T, so $\Delta^\pi \vdash_{\mathsf{T}} t^\pi = u^\pi$ as required.

The proof of the first part is direct from equivariance (Theorem A.4). □

We can permute atoms in freshnesses and equations at the *object-level* (without changing the freshness context):

Theorem 3.22 (Object-level equivariance)**.** *For any* π*:*

1. *if* $\Delta \vdash a\#t$ *then* $\Delta \vdash \pi(a)\#\pi \cdot t$*;*
2. *if* $\Delta \vdash_{\mathsf{T}} t = u$ *then* $\Delta \vdash_{\mathsf{T}} \pi \cdot t = \pi \cdot u$.

Proof. By induction on the structure of derivations. We consider the most interesting cases only. Suppose the derivation concludes in...

- $(\#\mathbf{X})$. Then $a\#\pi' \cdot X$ is derived from $\pi'^{-1}(a)\#X$, for some $\pi' \neq id$, and we need to show $\pi(a)\#\pi \cdot (\pi' \cdot X)$. By Lemma 3.2, this is equivalent to $\pi(a)\#(\pi \circ \pi') \cdot X$. We continue by case distinction:
  - If $\pi \circ \pi' = id$ then $\pi(a)\#(\pi \circ \pi') \cdot X$ is equivalent to the assumption $\pi'^{-1}(a)\#X$, since $\pi = \pi'^{-1}$ by basic permutation group theory.

– If $\pi \circ \pi' \neq id$ then by $(\#\mathbf{X})$ (which may now be applied),

$$\pi(a)\#(\pi \circ \pi') \cdot X \quad \text{follows from} \quad (\pi \circ \pi')^{-1}(\pi(a))\#X.$$

This is equivalent to the assumption $\pi'^{-1}(a)\#X$, since $(\pi \circ \pi')^{-1}(\pi(a)) = \pi'^{-1}(a)$.

- $(\mathbf{ax}_{\nabla \vdash \mathbf{t=u}})$. Then $t^{\pi'}\sigma = u^{\pi'}\sigma$ is derived and $\Delta \vdash_{\mathsf{T}} \nabla^{\pi'}\sigma$. We need to derive $\pi \cdot t^{\pi'}\sigma = \pi \cdot u^{\pi'}\sigma$ from Δ. By Lemma 3.5, $\pi \cdot t^{\pi'}\sigma = \pi \cdot u^{\pi'}\sigma$ is equivalent to $(\pi \cdot t^{\pi'})\sigma = (\pi \cdot u^{\pi'})\sigma$. Now let σ' map each $X \in \nabla, t, u$ to $\pi \cdot X$, then by Lemma 3.8 it suffices to derive

$$t^{\pi' \pi}(\sigma' \circ \sigma) = u^{\pi' \pi}(\sigma' \circ \sigma).$$

By Lemma 3.7, this is equivalent to $t^{\pi \circ \pi'}(\sigma' \circ \sigma) = u^{\pi \circ \pi'}(\sigma' \circ \sigma)$. Now this follows from $\nabla^{\pi \circ \pi'}(\sigma' \circ \sigma)$ by $(\mathbf{ax}_{\nabla \vdash \mathbf{t=u}})$ with permutation $\pi \circ \pi'$ and substitution $\sigma' \circ \sigma$. By Lemmas 3.5, 3.8 and 3.7 this is equivalent to $\pi \cdot \nabla^{\pi'}\sigma$. We are done since this follows from Δ by the inductive hypothesis.

- $(\mathbf{fr})$. Then $\Delta, a\#X_1, \ldots, a\#X_n \vdash_{\mathsf{T}} t = u$ for some $a \notin \Delta, t, u$ and we assume the inductive hypothesis of this derivation. If $\pi(a) = a$ there is no problem since then $a \notin \Delta, \pi \cdot t, \pi \cdot u$ and we may extend the derivation with $(\mathbf{fr})$.
  However, suppose $\pi(a) \neq a$ and so (possibly) $a \in \pi \cdot t, \pi \cdot u$. We observe that the predicate

  > "if the labelled tree Π is a valid derivation of $\Delta \vdash_{\mathsf{T}} t = u$, then for all permutations π' there are derivations of $\Delta \vdash_{\mathsf{T}} \pi' \cdot t = \pi' \cdot u$"

  has free variables Π, Δ, T, t and u.
  By equivariance (Theorem A.4), the predicate above holds of $\Pi^{(a'\,a)}$, $\Delta^{(a'\,a)}$, $\mathsf{T}^{(\mathsf{a}'\,\mathsf{a})}$, $t^{(a'\,a)}$ and $u^{(a'\,a)}$ (the informal notation $\Pi^{(a'\,a)}$ denotes Π in which all atoms are permuted according to $(a'\,a)$).[5] Now using Lemma 3.20 we deduce the inductive hypothesis of

$$\Delta, a'\#X_1, \ldots, a'\#X_n \vdash_{\mathsf{T}} t = u \quad \text{for any } a' \text{ such that } a' \notin \Delta, t, u \text{ and } \pi(a') = a'.$$

Then

$$\Delta, a'\#X_1, \ldots, a'\#X_n \vdash_{\mathsf{T}} \pi \cdot t = \pi \cdot u$$

and we extend the derivation with $(\mathbf{fr})$ to deduce $\Delta \vdash_{\mathsf{T}} \pi \cdot t = \pi \cdot u$ as required.

□

3.3.2 Substitution, weakening and strengthening

We can substitute terms for unknowns provided those terms violate no freshness assumptions made on the unknowns:

Theorem 3.23 (Meta-level substitution)**.** *Suppose Δ', Δ, σ are such that $\Delta' \vdash a\#(t\sigma)$ for every $a\#t \in \Delta$.*

1. *If $\Delta \vdash a\#t$ then $\Delta' \vdash a\#t\sigma$.*
2. *If $\Delta \vdash_{\mathsf{T}} t = u$ then $\Delta' \vdash_{\mathsf{T}} t\sigma = u\sigma$.*

Proof. Natural deduction derivations are such that the conclusion of one derivation may be 'plugged in' to an assumption in another derivation. For $(\#\mathbf{X})$ we use object-level equivariance (Theorem 3.22). For $(\mathbf{fr})$ we use equivariance (Theorem A.4) to rename the freshly chosen atom a if it is mentioned by Δ', $t\sigma$ or $u\sigma$. □

Corollary 3.24. *Suppose that t and u do not mention unknowns. If $\vdash_{\mathsf{T}} t = u$ then it has a derivation that does not mention any unknowns, instances of $(\#\mathbf{X})$, or instances of $(\mathbf{fr})$.*

[5] Instead of using equivariance we can work by induction on a notion of the *depth* of derivations. This would ignore that atoms are *atomic*, and have no internal structure, and so can be permuted. Equivariance gives more compact, more readable proofs.

Proof. Let Π be a derivation of $\vdash_{\mathsf{T}} t = u$. Take c to be a fresh atom (so c does not occur in Π). Let Π' be Π in which:

- each unknown X is mapped to c;
- each instance of $(\#\mathbf{X})$ is replaced by $(\#\mathbf{ab})$; that is, each instance of $(\#\mathbf{X})$ is of the form

$$\frac{[\pi^{-1}(a)\#X]}{a\#\pi \cdot X}\,(\#\mathbf{X}),$$

  where square brackets denote discharge of the freshness assumption. This is replaced by

$$\frac{}{a\#c}\,(\#\mathbf{ab}).$$

  We write c, not $\pi(c)$, because we chose c fresh so that $\pi(c) = c$.

By Theorem 3.23 it follows that Π' is a derivation of $t = u$, and it does not mention unknowns. The instances of $(\mathbf{fr})$ do not discharge any assumptions (they have been removed in the previous step), and these instances can now be removed. The result follows. □

The $(\mathbf{fr})$ rule is a form of explicit strengthening rule of freshness contexts, for derivable equality. A similar property is admissible for derivable freshness:

Lemma 3.25 (Strengthening)**.**

1. *If* $\Delta, a\#X_1, \ldots, a\#X_n \vdash b\#t$*, where* $n \geq 1$ *and* $a \notin \Delta, t$*, then* $\Delta \vdash b\#t$*.*
2. *If* $\Delta, a\#X_1, \ldots, a\#X_n \vdash_{\mathsf{T}} t = u$*, where* $n \geq 1$ *and* $a \notin \Delta, t, u$*, then* $\Delta \vdash_{\mathsf{T}} t = u$*.*

Proof. The equational case is precisely $(\mathbf{fr})$. For the freshness case, we inductively transform a derivation of $\Delta, a\#X_1, \ldots, a\#X_n \vdash b\#t$ to a derivation of $\Delta \vdash b\#t$:

- If $\Delta, a\#X_1, \ldots, a\#X_n \vdash b\#X$ by assumption, then $b\#X \in \Delta$, so $\Delta \vdash b\#X$ by assumption.
- $(\#\mathbf{ab})$ and $(\#[]\mathbf{a})$ carry over directly.
- $(\#\mathbf{X})$, $(\#[]\mathbf{b})$ and $(\#\mathsf{f})$ follow using the inductive hypothesis and the following properties: if $a \notin \pi \cdot X$ then $a \notin X$, if $a \notin [c]t$ then $a \notin t$, and if $a \notin \mathsf{f}(t_1, \ldots, t_n)$ then $a \notin t_i$ for all i.

□

We can leverage the existing results to prove a proof-theoretic version of a characteristic semantic property of atoms described in Lemma 4.17:

Corollary 3.26. *If* $\Delta \vdash_{\mathsf{T}} a = b$ *is derivable, then* $\vdash_{\mathsf{T}} c = d$ *for all* c *and* d*.*

Proof. Suppose $\Delta \vdash_{\mathsf{T}} a = b$. By Lemma 3.25 $\vdash_{\mathsf{T}} a = b$. The result follows by object-level equivariance (Theorem 3.22). □

3.3.3 A more computational presentation of CORE

In Example 2.15 we defined CORE as a family of nominal algebra theories with no axioms (one for each signature). In Subsection 3.2 we have given some examples to show that the $(\mathbf{perm})$ rule expresses α-equivalence with meta-variables.

We will now prove that theory CORE corresponds to the existing syntax-directed notion of α-equivalence on nominal terms from [UPG04, FG07]. So, the core notion of equality of nominal algebra *is* α-equivalence in the sense of nominal terms.

We will use this correspondence to show that equality in CORE is decidable and that theory CORE is *consistent* (does not equate *all* terms). Definition 3.29 was introduced in [UPG04, Figure 2]; the proofs follow the method presented in [FG07, p.13].

Definition 3.27. We write $\mathrm{ds}(\pi, \pi')$ for the **difference set** $\{a \mid \pi(a) \neq \pi'(a)\}$ of π and π'. We write $\mathrm{ds}(\pi, \pi')\#t$ for the set of freshnesses $\{a\#t \mid \pi(a) \neq \pi'(a)\}$.

$$\frac{}{a \approx_\Delta a}\ (\mathbf{Ax}) \qquad \frac{\Delta \vdash \mathrm{ds}(\pi', \pi)\#X}{\pi \cdot X \approx_\Delta \pi' \cdot X}\ (\mathbf{Ds})$$

$$\frac{t \approx_\Delta u}{[a]t \approx_\Delta [a]u}\ (\mathbf{Absaa}) \qquad \frac{(b\ a) \cdot t \approx_\Delta u \quad \Delta \vdash b\#t}{[a]t \approx_\Delta [b]u}\ (\mathbf{Absab})$$

$$\frac{t_1 \approx_\Delta u_1 \quad \cdots \quad t_n \approx_\Delta u_n}{\mathsf{f}(t_1, \ldots, t_n) \approx_\Delta \mathsf{f}(u_1, \ldots, u_n)}\ (\mathbf{F})$$

Figure 3: Syntax-directed rules for CORE

Lemma 3.28. *If* $\Delta \vdash \mathrm{ds}(\pi, \pi')\#t$ *then* $\Delta \vdash_{\mathsf{CORE}} \pi \cdot t = \pi' \cdot t$.

Proof. We work by induction on the number of elements in $\mathrm{ds}(\pi, \pi')$. If this set is empty then $\pi = \pi'$ and the result follows easily by (**refl**). Now suppose $a \in \mathrm{ds}(\pi, \pi')$. We construct a partial derivation of the proof obligation:

$$\frac{\pi \cdot t = ((\pi(a)\ \pi'(a)) \circ \pi') \cdot t \qquad \dfrac{\pi(a)\#\pi' \cdot t \quad \pi'(a)\#\pi' \cdot t}{((\pi(a)\ \pi'(a)) \circ \pi') \cdot t = \pi' \cdot t}\ (\mathbf{perm})}{\pi \cdot t = \pi' \cdot t}\ (\mathbf{tran})$$

The following proof obligations remain:

1. $\pi \cdot t = ((\pi(a)\ \pi'(a)) \circ \pi') \cdot t$ follows from $\mathrm{ds}(\pi, (\pi(a)\ \pi'(a)) \circ \pi')\#t$ by the inductive hypothesis, provided that
$$|\mathrm{ds}(\pi, (\pi(a)\ \pi'(a)) \circ \pi')| < |\mathrm{ds}(\pi, \pi')|.$$
This condition is satisfied, since
$$\mathrm{ds}(\pi, (\pi(a)\ \pi'(a)) \circ \pi') = \mathrm{ds}(\pi, \pi') \setminus \{a\}.$$
The remaining proof obligation $\mathrm{ds}(\pi, (\pi(a)\ \pi'(a)) \circ \pi')\#t$ follows by assumption $\mathrm{ds}(\pi, \pi')\#t$.
2. $\pi(a)\#\pi' \cdot t$ follows from $\pi'^{-1}(\pi(a))\#t$ by object-level equivariance (Theorem 3.22). Now this is one of the assumptions $\mathrm{ds}(\pi, \pi')\#t$: by Definition 3.27, $\pi'^{-1}(\pi(a)) \in \mathrm{ds}(\pi, \pi')$ when $\pi(\pi'^{-1}((\pi(a)))) \neq \pi(a)$, and, using the fact that $\neq$ is invariant under permutation, this follows from the assumption $\pi(a) \neq \pi'(a)$.
3. $\pi'(a)\#\pi' \cdot t$ follows from $a\#t$ by object-level equivariance (Theorem 3.22). This follows directly from assumption $\mathrm{ds}(\pi, \pi')\#t$, since $a \in \mathrm{ds}(\pi, \pi')$.

□

Definition 3.29. Let $t \approx_\Delta u$ be an ordered tuple of a term t, a freshness context Δ, and a term u. Let the **derivable equalities of** $t \approx_\Delta u$ be inductively defined by the rules in Figure 3.

Theorem 3.30 (Equivalence of CORE and $\approx_\Delta$)**.** $\Delta \vdash_{\mathsf{CORE}} t = u$ *is derivable if and only if* $t \approx_\Delta u$ *is derivable using the rules of Figure 3.*

Proof. The left-to-right direction is by induction on the structure of nominal algebra derivations of $\Delta \vdash_{\mathsf{CORE}} t = u$. By the inductive hypothesis it suffices to show:

- Syntax-directed equality $\approx_\Delta$ is an equivalence relation and a congruence. This is [FG07, Theorem 24].
- If $\Delta \vdash a\#t$ and $\Delta \vdash b\#t$ then $(a\ b) \cdot t \approx_\Delta t$. By induction on t.

- If $t \approx_{\Delta, a\#X_1, \ldots, a\#X_n} u$ where $a \notin t, u, \Delta$ then $t \approx_\Delta u$. By straightforward induction on the structure of derivations of $t \approx_{\Delta, a\#X_1, \ldots, a\#X_n} u$. The case of (**Absab**) uses strengthening (Lemma 3.25) to strengthen the assumption $\Delta, a\#X_1, \ldots, a\#X_n \vdash c\#t$ to $\Delta \vdash c\#t$.

For the right-to-left direction we work by induction on derivations of $t \approx_\Delta u$. By the inductive hypothesis it suffices to show:

- $\Delta \vdash_{\mathsf{CORE}} a = a$. This is an instance of (**refl**).
- If $\Delta \vdash \mathrm{ds}(\pi, \pi')\#X$ then $\Delta \vdash_{\mathsf{CORE}} \pi \cdot X = \pi' \cdot X$. This is Lemma 3.28.
- If $\Delta \vdash_{\mathsf{CORE}} t_i = u_i$ for $1 \leq i \leq n$, then $\Delta \vdash_{\mathsf{CORE}} \mathsf{f}(t_1, \ldots, t_n) = \mathsf{f}(u_1, \ldots, u_n)$. Using a number of instances of (**tran**) and (**congf**).
- If $\Delta \vdash_{\mathsf{CORE}} t = u$ then $\Delta \vdash_{\mathsf{CORE}} [a]t = [a]u$. This is (**cong**[]).
- If $\Delta \vdash_{\mathsf{CORE}} (b\ a) \cdot t = u$ and $\Delta \vdash b\#t$ then $\Delta \vdash_{\mathsf{CORE}} [a]t = [b]u$. Suppose that Π and Π' are derivations of $\Delta \vdash_{\mathsf{CORE}} (b\ a) \cdot t = u$ and $\Delta \vdash b\#t$ respectively. Then the following is a derivation of $\Delta \vdash_{\mathsf{CORE}} [a]t = [b]u$:

$$\dfrac{\dfrac{\dfrac{\dfrac{\vdots\ \Pi'}{b\#t}}{b\#[a]t}(\#[]\mathbf{b}) \qquad \dfrac{}{a\#[a]t}(\#[]\mathbf{a})}{[b](b\ a) \cdot t = [a]t}(\mathbf{perm})}{[a]t = [b](b\ a) \cdot t}(\mathbf{symm}) \qquad \dfrac{\dfrac{\vdots\ \Pi}{(b\ a) \cdot t = u}}{[b](b\ a) \cdot t = [b]u}(\mathbf{cong}[])}{[a]t = [b]u}(\mathbf{tran})$$

□

As corollaries of Theorem 3.30, we obtain syntactic criteria for determining equality in CORE, consistency of CORE, and preservation of freshness for CORE.

Corollary 3.31 (Decidability of CORE). *$\Delta \vdash_{\mathsf{CORE}} t = u$ precisely when one of the following holds:*

1. *$t \equiv a$ and $u \equiv a$.*
2. *$t \equiv \pi \cdot X$ and $u \equiv \pi' \cdot X$ and $\Delta \vdash \mathrm{ds}(\pi, \pi')\#X$.*
3. *$t \equiv [a]t'$ and $u \equiv [a]u'$ and $\Delta \vdash_{\mathsf{CORE}} t' = u'$.*
4. *$t \equiv [a]t'$ and $u \equiv [b]u'$ and $\Delta \vdash b\#t'$ and $\Delta \vdash_{\mathsf{CORE}} (b\ a) \cdot t' = u'$.*
5. *$t \equiv \mathsf{f}(t_1, \ldots, t_n)$ and $u \equiv \mathsf{f}(u_1, \ldots, u_n)$ and $\Delta \vdash_{\mathsf{CORE}} t_i = u_i$ for $1 \leq i \leq n$.*

Proof. By Theorem 3.30 it suffices to inspect the rules for $t \approx_\Delta u$, which are just a rendering of the above criteria in terms of derivation rules. □

Corollary 3.32 (Consistency of CORE). *For all Δ there are t and u such that $\Delta \nvdash_{\mathsf{CORE}} t = u$.*

Proof. By Corollary 3.31, $\Delta \vdash_{\mathsf{CORE}} a = b$ is never derivable. □

Corollary 3.33 (Preservation of freshness for CORE). *If $\Delta \vdash_{\mathsf{CORE}} t = u$ then*

$$\Delta \vdash a\#t \quad \textit{if and only if} \quad \Delta \vdash a\#u.$$

Proof. By induction on t using the syntactic criteria of Corollary 3.31. □

4 Denotations

In this section we provide a denotation of nominal algebra in terms in nominal sets. We give a brief overview of nominal sets in Subsection 4.1. In Subsection 4.2 we show how we can interpret freshness and equality in nominal sets, we define what constitutes a model of a theory (a signature

with a set of axioms), and we show that our notion of derivability is sound with respect to the semantics. In Subsection 4.3 we define the notion of free term models. We need this to show completeness of equality derivations in Subsection 4.4. Finally, we express a sense in which semantic freshness is complete in Subsection 4.5.

4.1 Nominal sets

We briefly review the parts of nominal sets relevant to this paper. For full treatments see [GP01] or [Pit03] ([Pit03] contains a simplified presentation of [GP01]).

Recall from Definitions 2.2 and 2.3 that we write $\mathbb{A}$ for the set of all atoms, $\mathbb{P}$ for the set of all permutations, id and $\circ$ for the identity and composition of permutations, and recall from Definition 3.27 that we write $\mathrm{ds}(\pi, \pi')$ for the difference set of π and π'.

Definition 4.1. A **$\mathbb{P}$-action** $\cdot$ on a set $\mathbb{X}$ is a function $\cdot : \mathbb{P} \times \mathbb{X} \to \mathbb{X}$, write it infix as $\pi \cdot x$, such that $id \cdot x = x$ and $\pi \cdot (\pi' \cdot x) = (\pi \circ \pi') \cdot x$ for all $x \in \mathbb{X}$. Say that a *finite* set of atoms A **supports** x when for any permutation π:

$$\text{if } \pi(a) = a \text{ for each } a \in A, \text{ then } \pi \cdot x = x.$$

Say that x has **finite support** when there exists such a set of atoms.

A **nominal set** is a set $\mathbb{X}$ equipped with a $\mathbb{P}$-action on $\mathbb{X}$ such that each $x \in \mathbb{X}$ has finite support.

In [GP01, Proposition 3.4] it is shown that if an element $x \in \mathbb{X}$ has finite support, then there is a unique least finite set of atoms that supports x.

Definition 4.2. When $x \in \mathbb{X}$ has finite support, call the least finite set of atoms that supports x the **support** of x, and write it as $\mathrm{supp}(x)$.

When $a \notin \mathrm{supp}(x)$ we write $a \#_{\mathrm{sem}} x$ and we say that a **is fresh for** x.

Lemma 4.3. *Basic results on nominal sets are:*

1. $\mathrm{supp}(x) = \{a \in \mathbb{A} \mid \{b \in \mathbb{A} \mid (a\ b) \cdot x \neq x\} \text{ is not finite}\}$.
2. *If* $\mathrm{ds}(\pi, \pi') \cap \mathrm{supp}(x) = \emptyset$ *then* $\pi \cdot x = \pi' \cdot x$.
3. $a \#_{\mathrm{sem}} x$ *if and only if* $\pi(a) \#_{\mathrm{sem}} \pi \cdot x$.

Proof. Elsewhere [GP01, Proposition 3.4] and by calculations. □

A corollary will be useful later:

Corollary 4.4. *Suppose* $\mathbb{X}$ *is a nominal set and suppose* $x \in \mathbb{X}$. *Suppose that* $b \#_{\mathrm{sem}} x$. *Then* $a \#_{\mathrm{sem}} x$ *if and only if* $(b\ a) \cdot x = x$.

Proof. Suppose that $(b\ a) \cdot x = x$. By assumption $b \notin \mathrm{supp}(x)$. By part 3 of Lemma 4.3 it follows that $a \notin \mathrm{supp}((b\ a) \cdot x)$. The result follows.

Suppose that $a \#_{\mathrm{sem}} x$. Then $a \notin \mathrm{supp}(x)$. By part 3 of Lemma 4.3 it follows that $b \notin \mathrm{supp}((b\ a) \cdot x)$. The result follows. □

Remark 4.5. Note that properties of denotational equality like 'if $x = y$ then $\pi \cdot x = \pi \cdot y$' and 'if $x = y$ then $a \in \mathrm{supp}(x)$ if and only if $a \in \mathrm{supp}(y)$' are immediate since an element has exactly the same properties as itself. This contrasts with similar properties of derivable equality, which as discussed in the Introduction we also write $=$, such as Theorem 3.22 and Corollary 3.33. These require proof.

Note further that in this paper we also use notions of derivable and denotational freshness. While the reader is probably familiar with derivable and denotational equality, they are perhaps not as familiar with derivable and denotational freshness. Correspondingly we go to more effort to be precise in our notation: we use $\#$ for derivable freshness as in $\Delta \vdash a\#t$; and we write $\#_{\mathrm{sem}}$ for denotational freshness (Definition 4.2) as in $a \#_{\mathrm{sem}} x$.

Example 4.6.

1. The set $\mathbb{A}$ of all atoms with action $\pi \cdot a = \pi(a)$ is a nominal set; $\mathrm{supp}(a) = \{a\}$. Note that for $x, y \in \mathbb{A}$, $x \in \mathrm{supp}(y)$ when $x = y$.
2. The powerset $\mathcal{P}(\mathbb{A}) = \{U \mid U \subseteq \mathbb{A}\}$ of $\mathbb{A}$ with action $\pi \cdot U = \{\pi \cdot u \mid u \in U\}$, is *not* a nominal set: enumerate atoms as $a_1, a_2, a_3, \ldots$; then *the comb*
$$comb = \{a_1, a_3, a_5, \ldots\} \in \mathcal{P}(\mathbb{A})$$
does not have finite support.
3. It is routine to use Lemma 4.3 to verify that the set of all finite sets of atoms, with the pointwise action inherited from $\mathcal{P}(\mathbb{A})$, is a nominal set. If $U \subseteq \mathbb{A}$ is finite then $\mathrm{supp}(U) = A$.
4. Call $U \subseteq \mathbb{A}$ **cofinite** when $\mathbb{A}\backslash U$ is finite. It is also routine to use Lemma 4.3 to verify that the set of all cofinite sets of atoms, with the pointwise action, is a nominal set. If $U \subseteq \mathbb{A}$ is cofinite, then $\mathrm{supp}(U) = \mathbb{A}\backslash U$.
Note that the support of $\mathbb{A}\backslash\{a\}$ is $\{a\}$, so $b \#_{\mathrm{sem}} \mathbb{A}\backslash\{a\}$ but not $a \#_{\mathrm{sem}} \mathbb{A}\backslash\{a\}$.
5. The set $\mathcal{P}_{fs}(\mathbb{A})$ of finite and cofinite subsets of $\mathbb{A}$ is a nominal set (*fs* stands for finite support). It can be proved that a set of atoms is finitely-supported if and only if it is finite or cofinite. Generalisations of the notion of finite support have been considered; to any ultrafilter [Che06], and to any well-orderable set of atoms [Gab07b].
6. The empty set $\emptyset$ with the trivial action is a nominal set.
7. The set of infinitary λ-calculus expressions [KKSdV97] with the pointwise action is *not* a nominal set: expressions might mention an infinite number of different atoms, so they do not adhere to the finite support property. This problem can be overcome by moving to FMG (Fraenkel Mostowski Generalised) [Gab07b]. This generalises the countable set of atoms to any large cardinality, and finite sets of atoms to any strictly smaller cardinality (well-orderable sets to be precise).

Remark 4.7. We should think of $\mathrm{supp}(x)$ as a sets-based notion of 'occurs in x *conspicuously*'. An element can be conspicuous by its absence, as well as its presence; consider the example of $\mathrm{supp}(\mathbb{A}\backslash\{a\}) = \{a\}$ from $\mathcal{P}_{fs}(\mathbb{A})$ above.

Definition 4.8. If $\mathbb{X}$ and $\mathbb{Y}$ are nominal sets write $\mathbb{X} \times \mathbb{Y}$ for
$$\{(x, y) \mid x \in \mathbb{X},\ y \in \mathbb{Y}\} \quad \text{with action} \quad \pi \cdot (x, y) = (\pi \cdot x, \pi \cdot y).$$
This is a nominal set; the support of $(x, y) \in \mathbb{X} \times \mathbb{Y}$ is the union of the supports of x and y. In symbols, $\mathrm{supp}((x, y)) = \mathrm{supp}(x) \cup \mathrm{supp}(y) \subseteq \mathbb{A}$.

If $\mathbb{X}$ is a nominal set write $\mathbb{X}^n$ for
$$\{(x_1, \ldots, x_n) \mid x_i \in \mathbb{X}, 1 \leq i \leq n\} \quad \text{with action} \quad \pi \cdot (x_1, \ldots, x_n) = (\pi \cdot x_1, \ldots, \pi \cdot x_n).$$
Again, this is a nominal set; the support of an element $(x_1, \ldots, x_n)$ is the union of the supports of the x_i. In symbols: $\mathrm{supp}((x_1, \ldots, x_n)) = \bigcup\{\mathrm{supp}(x_i) \mid 1 \leq i \leq n\} \subseteq \mathbb{A}$.

Definition 4.9. For any nominal sets $\mathbb{X}, \mathbb{Y}$, call a function $f \in \mathbb{X} \to \mathbb{Y}$ (on the underlying sets) **equivariant** when $\pi \cdot f(x) = f(\pi \cdot x)$ for any $x \in \mathbb{X}$.

Lemma 4.10. *Suppose $\mathbb{X}$ and $\mathbb{Y}$ are nominal sets. Suppose $f \in \mathbb{X} \to \mathbb{Y}$ is equivariant and suppose $x \in \mathbb{X}$. Then*
$$\mathrm{supp}(f(x)) \subseteq \mathrm{supp}(x).$$
As a corollary, $a \#_{\mathrm{sem}} x$ implies $a \#_{\mathrm{sem}} f(x)$.

Proof. By Definition 4.9 $\pi \cdot f(x) = f(\pi \cdot x)$, so if $\pi \cdot x = x$ then $\pi \cdot f(x) = f(x)$.
The corollary follows by Definition 4.2. □

Subsets of (the underlying set of) a nominal set will be important later when we build free term algebras.

Definition 4.11. $\mathcal{X} \subseteq \mathbb{X}$ inherits a pointwise action $\pi \cdot \mathcal{X} = \{\pi \cdot x \mid x \in \mathcal{X}\}$.
We will always use this action on $\mathcal{X} \subseteq \mathbb{X}$.

$a\#_{\text{sem}}\mathcal{X}$ does *not* imply that $a\#_{\text{sem}}x$ for every $x \in \mathcal{X}$. For example $\mathbb{A} \subseteq \mathbb{A}$ and it is a fact that $a\#_{\text{sem}}\mathbb{A}$ — but $a \in \mathbb{A}$ and not $a\#_{\text{sem}}a$. Furthermore $\mathcal{X} \subseteq \mathbb{X}$ does *not* imply that $\mathcal{X}$ is finitely supported. For example recall *comb* from Example 4.6; *comb* $\subseteq \mathbb{A}$ but *comb* is not finitely supported. However, the finitely-supported subsets of $\mathbb{X}$ form a nominal set — they have a permutation action, and are finitely supported.
We conclude with a useful technical lemma:

Lemma 4.12. *Suppose $\mathbb{X}$ is a nominal set and suppose $\mathcal{X} \subseteq \mathbb{X}$ is finitely-supported and nonempty. Suppose that*

$$a_1\#_{\text{sem}}\mathcal{X}, \ldots, a_n\#_{\text{sem}}\mathcal{X}$$

Then there exists some $x \in \mathcal{X}$ such that

$$a_1\#_{\text{sem}}x, \ldots, a_n\#_{\text{sem}}x.$$

Proof. Choose any $y \in \mathcal{X}$. Since $\mathcal{X}$ and y are finitely-supported, we can find (distinct) atoms $b_1, \ldots, b_n$ such that $b_i\#_{\text{sem}}\mathcal{X}$ and $b_i\#_{\text{sem}}y$ for $1 \leq i \leq n$. Then by part 2 of Lemma 4.3 we have that $(b_1\ a_1) \cdots (b_n\ a_n) \cdot \mathcal{X} = \mathcal{X}$.
Write $x = (b_1\ a_1) \cdots (b_n\ a_n) \cdot y$. Then $x \in \mathcal{X}$ by Definition 4.11 and we conclude $a_i\#_{\text{sem}}x$ for $1 \leq i \leq n$ by part 3 of Lemma 4.3 and the assumption $b_i\#_{\text{sem}}y$. □

4.2 Interpretations, models and validity

Nominal algebra is a logic of equality, tailored to nominal sets; in this subsection we give that observation concrete mathematical force, by defining a notion of semantics for nominal algebra theories, in nominal sets. Recall the definition of an *equivariant* function (Definition 4.9).

Definition 4.13. An **interpretation of a signature** Σ is a tuple

$$\mathcal{I} = \big(|\mathcal{I}|,\ \mathcal{I}_{atm},\ \mathcal{I}_{abs},\ \{\mathcal{I}_{\mathsf{f}} \mid \mathsf{f} \in \Sigma\}\big)$$

where:

- $|\mathcal{I}|$ is a nominal set.
  This is the **underlying** set (also often called the *carrier* set) of the interpretation.
- $\mathcal{I}_{atm} \in \mathbb{A} \to |\mathcal{I}|$ is an equivariant function.
  We use this to interpret atoms.
- $\mathcal{I}_{abs} \in |\mathcal{I}| \times |\mathcal{I}| \to |\mathcal{I}|$ is an equivariant function such that $a\#_{\text{sem}}\mathcal{I}_{abs}(\mathcal{I}_{atm}(a), x)$ for all $a \in \mathbb{A}$ and $x \in |\mathcal{I}|$.
  We use this to interpret abstraction.
- $\mathcal{I}_{\mathsf{f}} \in |\mathcal{I}|^n \to |\mathcal{I}|$ is an equivariant function for each term-former $\mathsf{f} : n$ in Σ.
  We use this to interpret term-formers. ($|\mathcal{I}|^n$ is defined in Definition 4.8.)

We extend the notion of interpretation to terms, where we use a valuation to map unknowns to elements in $|\mathcal{I}|$:

Definition 4.14. A **valuation (to $|\mathcal{I}|$)** ς maps unknowns X to elements $\varsigma(X) \in |\mathcal{I}|$. We write $[\![t]\!]^{\mathcal{I}}_{\varsigma}$ for the **interpretation of a term** t under a valuation ς, inductively defined by:

$$[\![a]\!]^{\mathcal{I}}_{\varsigma} = \mathcal{I}_{atm}(a) \qquad [\![\pi \cdot X]\!]^{\mathcal{I}}_{\varsigma} = \pi \cdot \varsigma(X) \qquad [\![[a]t]\!]^{\mathcal{I}}_{\varsigma} = \mathcal{I}_{abs}(\mathcal{I}_{atm}(a), [\![t]\!]^{\mathcal{I}}_{\varsigma})$$
$$[\![\mathsf{f}(t_1, \ldots, t_n)]\!]^{\mathcal{I}}_{\varsigma} = \mathcal{I}_{\mathsf{f}}([\![t_1]\!]^{\mathcal{I}}_{\varsigma}, \ldots, [\![t_n]\!]^{\mathcal{I}}_{\varsigma})$$

Remark 4.15. In view of the fact that in an atoms-abstraction $[a]t$ the syntax insists on an atom and a term, the reader might have expected correspondingly that $\mathcal{I}_{abs}$ should be a function in $\mathbb{A} \times |\mathcal{I}| \to |\mathcal{I}|$ (instead of a function in $|\mathcal{I}| \times |\mathcal{I}| \to |\mathcal{I}|$), such that $a \#_{\text{sem}} \mathcal{I}_{abs}(a, x)$ for all $(a, x) \in \mathbb{A} \times |\mathcal{I}|$. If we do that, then a model and valuation may exist such that $[\![a]\!]^{\mathcal{I}}_{\varsigma} = [\![b]\!]^{\mathcal{I}}_{\varsigma}$, and $[\![[a]X]\!]^{\mathcal{I}}_{\varsigma} \neq [\![[b]X]\!]^{\mathcal{I}}_{\varsigma}$. There is nothing actually *wrong* with that (in the formal sense that soundness and completeness will still hold, see [Mat07] for details), but we find it convenient to exclude such models.

Interpretations are equivariant:

Lemma 4.16. *For any* π, $\quad \pi \cdot [\![t]\!]^{\mathcal{I}}_{\varsigma} = [\![\pi \cdot t]\!]^{\mathcal{I}}_{\varsigma}$.

Proof. By induction on the structure of t, using Lemma 4.10 for the cases of a, $[a]t$ and $\mathsf{f}(t_1, \ldots, t_n)$. □

Lemma 4.17 is the semantic analogue of Corollary 3.26:

Lemma 4.17. *If* $[\![a]\!]^{\mathcal{I}}_{\varsigma} = [\![b]\!]^{\mathcal{I}}_{\varsigma}$ *then* $[\![c]\!]^{\mathcal{I}}_{\varsigma'} = [\![d]\!]^{\mathcal{I}}_{\varsigma'}$ *for all* c, d, *and* ς'.

Proof. Unpacking the definitions, it suffices to prove that if $\mathcal{I}_{atm}(a) = \mathcal{I}_{atm}(b)$ then $\mathcal{I}_{atm}$ is a constant function. Choose any other atom d. We reason as follows:

$$\begin{array}{ll} \mathcal{I}_{atm}(d) = (d\ a) \cdot \mathcal{I}_{atm}(a) & \text{Equivariance (Definition 4.9)} \\ = (d\ a) \cdot \mathcal{I}_{atm}(b) & \text{Assumption} \\ = \mathcal{I}_{atm}((d\ a)(b)) & \text{Equivariance} \\ = \mathcal{I}_{atm}(b) & \text{Fact} \end{array}$$

The result follows. □

Using the interpretations of signatures and terms, we define the notion of *validity* on judgement forms as follows:

Definition 4.18. For any interpretation $\mathcal{I}$, say that:

$[\![\Delta]\!]^{\mathcal{I}}_{\varsigma}$ **(is valid)**	when	$a \#_{\text{sem}} \varsigma(X)$ for each $a \# X \in \Delta$
$[\![\Delta \vdash a \# t]\!]^{\mathcal{I}}_{\varsigma}$	when	$[\![\Delta]\!]^{\mathcal{I}}_{\varsigma}$ implies $a \#_{\text{sem}} [\![t]\!]^{\mathcal{I}}_{\varsigma}$
$[\![\Delta \vdash t = u]\!]^{\mathcal{I}}_{\varsigma}$	when	$[\![\Delta]\!]^{\mathcal{I}}_{\varsigma}$ implies $[\![t]\!]^{\mathcal{I}}_{\varsigma} = [\![u]\!]^{\mathcal{I}}_{\varsigma}$
$[\![\Delta \vdash a \# t]\!]^{\mathcal{I}}$	when	$[\![\Delta \vdash a \# t]\!]^{\mathcal{I}}_{\varsigma}$ for all valuations ς
$[\![\Delta \vdash t = u]\!]^{\mathcal{I}}$	when	$[\![\Delta \vdash t = u]\!]^{\mathcal{I}}_{\varsigma}$ for all valuations ς

Validity is equivariant in a sense very similar to that described in Theorem 3.21:

Lemma 4.19. *For any* π,

1. $[\![\Delta \vdash a \# t]\!]^{\mathcal{I}}$ *if and only if* $[\![\Delta^{\pi} \vdash \pi(a) \# t^{\pi}]\!]^{\mathcal{I}}$.
2. $[\![\Delta \vdash t = u]\!]^{\mathcal{I}}$ *if and only if* $[\![\Delta^{\pi} \vdash t^{\pi} = u^{\pi}]\!]^{\mathcal{I}}$.

Proof. Direct from ZFA equivariance (Theorem A.4). □

Then a model of a theory is an interpretation that validates its axioms:

Definition 4.20. An interpretation $\mathcal{I}$ of T is a **model** when

$$[\![\nabla \vdash t = u]\!]^{\mathcal{I}} \quad \text{for all axioms } \nabla \vdash t = u \text{ of } \mathsf{T}.$$

Lemma 4.21. *Define the* ***singleton interpretation*** $\mathcal{S}$ *to have underlying set* $\{*\}$ *(a singleton set containing a single, equivariant element), with* $\mathcal{S}_{atm}$, $\mathcal{S}_{abs}$, *and* $\mathcal{S}_{\mathsf{f}}$ *constant functions with value* $*$.

Then $\mathcal{S}$ *is an interpretation and a model for every theory.*

Proof. By routine verifications. □

Remark 4.22. By Lemma 4.17, $\mathcal{I}_{atm}$ is either injective (so atoms 'live in' $|\mathcal{I}|$), or constant. We do not insist that $\mathcal{I}_{atm} \in \mathbb{A} \to |\mathcal{I}|$ is injective (i.e. we allow $\mathcal{I}_{atm}$ to be constant). We prefer this, for two reasons:

- We want the singleton interpretation of Lemma 4.21 to be an interpretation and to be a model for all theories, just as it is for universal algebra.
- We want to avoid that $\Delta \vdash_{\mathsf{T}} a = b$ implies that T has no model; by Lemma 4.17 this would happen, if we insisted that $\mathcal{I}_{atm}$ is a constant function.

  In particular, we want the theory with a single axiom $\vdash a = b$ to have a model, and we want the 'universal theory' with axiom $\vdash X = Y$ to have a model.

We find this more elegant, and more in keeping with our completeness result (Theorem 4.39).

Definition 4.23. For any theory T, define **validity with respect to** T for judgement forms as follows:

- Write $\Delta \models_{\mathsf{T}} a\#t$ when $[\![\Delta \vdash a\#t]\!]^{\mathcal{I}}$ for all models $\mathcal{I}$ of T.
- Write $\Delta \models_{\mathsf{T}} t = u$ when $[\![\Delta \vdash t = u]\!]^{\mathcal{I}}$ for all models $\mathcal{I}$ of T.

Note the T subscript in $\models_{\mathsf{T}}$, which indicates that semantic freshness is not a purely syntax-directed affair, as is freshness derivability, but it also depends on the axioms of theory T. More on this in Subsections 4.5 and 5.2.

Derivability of freshness and equality is sound for the semantics:

Theorem 4.24 (Soundness)**.** *For any* T, Δ, a, t, u*:*

1. *If* $\Delta \vdash a\#t$ *then* $\Delta \models_{\mathsf{T}} a\#t$.
2. *If* $\Delta \vdash_{\mathsf{T}} t = u$ *then* $\Delta \models_{\mathsf{T}} t = u$.

Proof. Let $\mathcal{I}$ be a model of T. We must show for any valuation ς that if $\Delta \vdash a\#t$ and $[\![\Delta]\!]^{\mathcal{I}}_{\varsigma}$ then $a \#_{\mathrm{sem}} [\![t]\!]^{\mathcal{I}}_{\varsigma}$, and that if $\Delta \vdash_{\mathsf{T}} t = u$ and $[\![\Delta]\!]^{\mathcal{I}}_{\varsigma}$ then $[\![t]\!]^{\mathcal{I}}_{\varsigma} = [\![u]\!]^{\mathcal{I}}_{\varsigma}$.

We work by induction on derivations (Figures 1 and 2). Fix some valuation ς.

- (**#ab**). We must show $a \#_{\mathrm{sem}} [\![b]\!]^{\mathcal{I}}_{\varsigma}$, i.e. $a \#_{\mathrm{sem}} \mathcal{I}_{abs}(b)$. By Lemma 4.10 this follows from $a \#_{\mathrm{sem}} b$ (see part 1 of Example 4.6).
- (**#X**). By the inductive hypothesis we know $\pi^{-1}(a) \#_{\mathrm{sem}} \varsigma(X)$. By part 3 of Lemma 4.3 we conclude $a \#_{\mathrm{sem}} \pi \cdot \varsigma(X)$.
- (**#[]a**). $a \#_{\mathrm{sem}} \mathcal{I}_{abs}(\mathcal{I}_{atm}(a), [\![t]\!]^{\mathcal{I}}_{\varsigma})$ holds by assumption (recall Definition 4.13).
- (**#[]b**). $a \#_{\mathrm{sem}} [\![t]\!]^{\mathcal{I}}_{\varsigma}$ implies $a \#_{\mathrm{sem}} \mathcal{I}_{abs}(\mathcal{I}_{atm}(b), [\![t]\!]^{\mathcal{I}}_{\varsigma})$, by Lemma 4.10.
- (**#f**). If $a \#_{\mathrm{sem}} [\![t_i]\!]^{\mathcal{I}}_{\varsigma}$ for $1 \leq i \leq n$ then $a \#_{\mathrm{sem}} \mathcal{I}_{\mathsf{f}}([\![t_1]\!]^{\mathcal{I}}_{\varsigma}, \ldots, [\![t_n]\!]^{\mathcal{I}}_{\varsigma})$ follows using Lemma 4.10.
- (**refl**), (**symm**), (**tran**), (**cong**[]), (**congf**). By properties of equality.
- (**perm**). We know that $a \#_{\mathrm{sem}} [\![t]\!]^{\mathcal{I}}_{\varsigma}$ and $b \#_{\mathrm{sem}} [\![t]\!]^{\mathcal{I}}_{\varsigma}$ imply $(a\ b) \cdot [\![t]\!]^{\mathcal{I}}_{\varsigma} = [\![t]\!]^{\mathcal{I}}_{\varsigma}$ by part 2 of Lemma 4.3. We conclude $[\![(a\ b) \cdot t]\!]^{\mathcal{I}}_{\varsigma} = [\![t]\!]^{\mathcal{I}}_{\varsigma}$ by Lemma 4.16.
- ($\mathbf{ax}_{\nabla \vdash \mathbf{t}=\mathbf{u}}$). Suppose $[\![\nabla^{\pi}\sigma]\!]^{\mathcal{I}}_{\varsigma}$. Then $\pi(a) \#_{\mathrm{sem}} [\![\sigma(X)]\!]^{\mathcal{I}}_{\varsigma}$ holds for all $a\#X \in \nabla$. By part 3 of Lemma 4.3 also $a \#_{\mathrm{sem}} \pi^{-1} \cdot [\![\sigma(X)]\!]^{\mathcal{I}}_{\varsigma}$ for all $a\#X \in \nabla$. Let ς' be defined as

  $$\varsigma'(X) = \pi^{-1} \cdot [\![\sigma(X)]\!]^{\mathcal{I}}_{\varsigma} \quad \text{for every } X.$$

  Then $a \#_{\mathrm{sem}} \varsigma'(X)$ for all $a\#X \in \nabla$, so $[\![\nabla]\!]^{\mathcal{I}}_{\varsigma'}$ holds. Since $\nabla \vdash t = u$ is an axiom of T, we know $[\![t]\!]^{\mathcal{I}}_{\varsigma'} = [\![u]\!]^{\mathcal{I}}_{\varsigma'}$. Then trivially also $\pi \cdot [\![t]\!]^{\mathcal{I}}_{\varsigma'} = \pi \cdot [\![u]\!]^{\mathcal{I}}_{\varsigma'}$, and by Lemma 4.16 we obtain $[\![\pi \cdot t]\!]^{\mathcal{I}}_{\varsigma'} = [\![\pi \cdot u]\!]^{\mathcal{I}}_{\varsigma'}$. Now by a straightforward induction on syntax we can verify that $[\![\pi \cdot t]\!]^{\mathcal{I}}_{\varsigma'} = [\![t^{\pi}\sigma]\!]^{\mathcal{I}}_{\varsigma}$ and $[\![\pi \cdot u]\!]^{\mathcal{I}}_{\varsigma'} = [\![u^{\pi}\sigma]\!]^{\mathcal{I}}_{\varsigma}$, and we conclude $[\![t^{\pi}\sigma]\!]^{\mathcal{I}}_{\varsigma} = [\![u^{\pi}\sigma]\!]^{\mathcal{I}}_{\varsigma}$.

- (**fr**). Suppose $\Delta \vdash_{\mathsf{T}} t = u$ is derived from $\Delta, a\#X_1, \ldots, a\#X_n \vdash_{\mathsf{T}} t = u$, where $a \notin t, u, \Delta$. By equivariance (Theorem A.4) then also

$$\Delta, a'\#X_1, \ldots, a'\#X_n \vdash_{\mathsf{T}} t = u$$

for all other a' not occurring in Δ, t or u, and we retain the inductive hypothesis for $\Delta, a'\#X_1, \ldots, a'\#X_n \vdash_{\mathsf{T}} t = u$.
We must show that $[\![\Delta \vdash_{\mathsf{T}} t = u]\!]^{\mathcal{I}}_{\varsigma}$ for any ς. Given ς, pick an $a' \notin \Delta, t, u$ such that $a' \#_{sem} \varsigma(X_i)$ for $1 \leq i \leq n$. Then by the inductive hypothesis we obtain $[\![\Delta, a'\#X_1, \ldots, a'\#X_n \vdash_{\mathsf{T}} t = u]\!]^{\mathcal{I}}_{\varsigma}$. But this is equivalent to $[\![\Delta \vdash_{\mathsf{T}} t = u]\!]^{\mathcal{I}}_{\varsigma}$, since $[\![a'\#X_1, \ldots, a'\#X_n]\!]^{\mathcal{I}}_{\varsigma}$. The result follows.

□

4.3 Free term models

In order to show that derivability of equality is complete with respect to the semantics, we need the notion of a *free term model*, which is slightly but significantly different from the definition normally used for universal algebra.

Definition 4.25. In this subsection fix a signature Σ, a theory $\mathsf{T} = (\Sigma, Ax)$, and fix a set of term-formers D disjoint from Σ.

In Subsection 4.4 we will construct a specific D and use it to prove Completeness, but nothing in this subsection depends on that choice; here, D can be any set of term-formers, including an infinite set, or the empty set.

The usual technique to obtain models for a theory T is to add constant symbols to the language (to ensure a supply of 'arbitrary elements') and quotient by provable equality. But in nominal algebra constants have empty support; if d has arity 0 then $\vdash a\#\mathsf{d}$ is derivable for any a. Adding constants only ensures a supply of elements with empty support.

To reflect in syntax that an element of a nominal set can have support, we use n-ary term-formers d applied to n distinct atoms. This idea goes back to [Gab07a]. We now give the construction in detail.

Definition 4.26. Let **free terms** be inductively generated by the following grammar:

$$g \quad ::= \quad a \;\mid\; [a]g \;\mid\; \mathsf{f}(g_1, \ldots, g_n) \;\mid\; \mathsf{d}(a_1, \ldots, a_n)$$

Here $\mathsf{f} : n$ ranges over elements of Σ, and $\mathsf{d} : n$ ranges over elements of D.

Recall the notation $\pi \cdot t$ for the object-level permutation action on a term t from Definition 3.1.

Lemma 4.27. *The set of free terms with action $\pi \cdot g$ is a nominal set; the support of g is $\{a \in \mathbb{A} \mid a \in g\}$.*
As a corollary, $a \notin g$ if and only if $a \#_{sem} g$.

Proof. $\mathrm{supp}(g) = \{a \in \mathbb{A} \mid a \notin g\}$ follows by an induction on the structure of g, using part 1 of Lemma 4.3. This result is obvious; we are now thinking of g just as a labelled tree structure. Our notion of equality is syntactic identity $\equiv$ so that, for example, $[a]a \not\equiv [b]b$ and thus $a \#_{sem} [a]a$ is false.[6] The corollary follows by Definition 4.2. □

Definition 4.28. Write (**congd**) for an instance of the (**congf**) rule when $\mathsf{f} \in \mathsf{D}$. Write $[g]_{\mathsf{T}}$ for the set of free terms g' such that a derivation of $\vdash_{\mathsf{T}} g = g'$ exists that does not mention (**congd**) for any $\mathsf{d} \in \mathsf{D}$.

Let the **set of free terms up to** T be the set $\{[g]_{\mathsf{T}} \mid g \text{ a free term}\}$.

[6] The reader may find it useful to recall the meta-level permutation action (Definition 3.6). It is a fact that on free terms this coincides with the object-level permutation action (Definition 3.1). Intuitively, the native notion of equality for the object-level permutation action is derivable equality, and the native notion of equality for the meta-level permutation action is synactic identity. It is this latter notion of identity which we are concerned with when we judge whether a is fresh for g as a labelled tree structure.

Lemma 4.29. *The set of free terms up to* T *with action* $\pi \cdot [g]_{\mathsf{T}} = [\pi \cdot g]_{\mathsf{T}}$ *is a nominal set;* $[g]_{\mathsf{T}}$ *is supported by* $\{a \in \mathbb{A} \mid a \in g\}$.

Proof. It is easy to check that $\pi \cdot [g]_{\mathsf{T}} = [\pi \cdot g]_{\mathsf{T}}$ defines a permutation action. Let π be a permutation such that $\pi(a) = a$ for all $a \in g$. We must show $\pi \cdot [g]_{\mathsf{T}} = [g]_{\mathsf{T}}$. By assumption $\pi \cdot [g]_{\mathsf{T}} = [\pi \cdot g]_{\mathsf{T}}$, and by part 2 of Lemma 4.3 and Lemma 4.27 we obtain that (or we can inductively prove that) $\pi \cdot g \equiv g$. The result follows by (**refl**). □

The following technical corollary will be useful later:

Corollary 4.30. $a_1 \#_{sem} [g]_{\mathsf{T}}, \ldots, a_n \#_{sem} [g]_{\mathsf{T}}$ *if and only if one of the following holds:*

- *There is some* $g' \in [g]_{\mathsf{T}}$ *such that* $a_1 \#_{sem} g', \ldots, a_n \#_{sem} g'$.
- *There is some* $g' \in [g]_{\mathsf{T}}$ *such that* $a_1 \notin g', \ldots, a_n \notin g'$.

Proof. By Lemma 4.27 the two alternatives are equivalent.

The left-to-right implication is from Lemma 4.12. For the right-to-left implication, suppose there is some $g' \in [g]_{\mathsf{T}}$ (so $[g]_{\mathsf{T}} = [g']_{\mathsf{T}}$) such that $a_1 \notin g', \ldots, a_n \notin g'$. The result follows since by Lemma 4.29 $a_1 \#_{\mathrm{sem}} [g']_{\mathsf{T}}, \ldots, a_n \#_{\mathrm{sem}} [g']_{\mathsf{T}}$. □

The following example illustrates how Corollary 4.30 is non-trivial.

Example 4.31. Consider a theory ATOM with one axiom $\vdash a = b$. It is easy to verify that $a \#_{\mathrm{sem}} [a]_{\mathsf{ATOM}}$ (since $[a]_{\mathsf{ATOM}} = \mathbb{A}$), but $a \#_{\mathrm{sem}} a$ is false. Of course, $\vdash_{\mathsf{ATOM}} a = b$ is derivable and $a \#_{\mathrm{sem}} b$ is true. Similarly in LAM it is a fact that $a \#_{\mathrm{sem}} [(\lambda[a]b)a]_{\mathsf{LAM}}$ but not $a \#_{\mathrm{sem}} (\lambda[a]b)a$. Of course, $\vdash_{\mathsf{LAM}} (\lambda[a]b)a = b$ is derivable and $a \#_{\mathrm{sem}} b$ is true.

Definition 4.32. We construct the **free term model** $\mathcal{T}$ of T over D as follows:

- Take $|\mathcal{T}|$ equal to the set of free terms up to T (Definition 4.28) with action $\pi \cdot [g]_{\mathsf{T}} = [\pi \cdot g]_{\mathsf{T}}$.
- $\mathcal{T}_{atm}(a) = [a]_{\mathsf{T}}$.
- We set $\mathcal{T}_{abs}(y, x) = [[a]g]_{\mathsf{T}}$ for some $g \in x$ and atom $a \in y$, if some such atom a exists (we prove this is well-defined in Theorem 4.36 below).
  Otherwise, we set $\mathcal{T}_{abs}(y, x) = x$.
- $\mathcal{T}_{\mathsf{f}}(x_1, \ldots, x_n) = [\mathsf{f}(g_1, \ldots, g_n)]_{\mathsf{T}}$ for some $g_1 \in x_1, \ldots, g_n \in x_n$
  (for each term-former $\mathsf{f} : n$ in Σ).

Remark 4.33. It is usual to build free term models by quotienting by derivable equality; we exclude (**congd**) to avoid the following degenerate case:

If we allow (**congd**) and T contains an axiom such as $\vdash a = b$, then $\mathrm{supp}[\mathsf{d}(a_1, \ldots, a_n)]_{\mathsf{T}} = \emptyset$. This is not the behaviour we want. Similarly our syntax of free terms does not allow terms of the form $\mathsf{d}(g_1, \ldots, g_n)$ in general.

Consistent with a construction used in [Gab07a], the only purpose of $[\mathsf{d}(a_1, \ldots, a_n)]_{\mathsf{T}}$ is to 'be an unknown element with support $a_1, \ldots, a_n$'.

We need two technical lemmas:

Lemma 4.34. *For any free term* g, *if* $a \notin g$ *or* $a \#_{sem} g$ *then* $\vdash a \# g$.

Proof. We note that by Lemma 4.27, $a \notin g$ and $a \#_{\mathrm{sem}} g$ are equivalent. We prove that $a \notin g$ implies $\vdash a \# g$ by an easy induction on syntax, using the rules in Figure 1. □

Lemma 4.35. $a \#_{sem} [[a]g]_{\mathsf{T}}$ *always.*

Proof. By Corollary 4.30 it suffices to exhibit some $g' \in [[a]g]_{\mathsf{T}}$ such that $a \notin g'$. Unpacking definitions and using Corollary 4.30 it suffices to exhibit some g' such that $\vdash_{\mathsf{T}} g' = [a]g$ and such that $a \notin g'$. We choose fresh b (so $b \notin [a]g$) and set $g' \equiv [b](b\ a) \cdot g$; it is easy to check that $a \notin g'$. By (#[]**a**) we know $\vdash a \#[a]g$. By assumption $b \notin [a]g$ so by Lemma 4.34 also $\vdash b \# g$. It follows by (**perm**) that $\vdash_{\mathsf{T}} (b\ a) \cdot [a]g = [a]g$; since $g' \equiv (b\ a) \cdot [a]g$ we deduce $g' \in [[a]g]_{\mathsf{T}}$. The result follows. □

Theorem 4.36. *$\mathcal{T}$ is an interpretation of Σ.*

Proof. $|\mathcal{T}|$ is a nominal set by Lemma 4.29.
We must show that $\mathcal{T}_{atm}$, $\mathcal{T}_{abs}$, and $[\![\mathsf{f}]\!]^{\mathcal{T}}$ are equivariant.
We must show that $\mathcal{T}_{abs}$ and $[\![\mathsf{f}]\!]^{\mathcal{T}}$ are well-defined.

- *$\mathcal{T}_{abs}(y,x)$ is well-defined.* There are three cases.
  - If there exists no $a \in \mathbb{A}$ such that $a \in y$ then $\mathcal{T}_{abs}(y,x) = x$ and there is nothing to prove.
  - Suppose that $a \in \mathbb{A}$ is unique such that $a \in y$. Suppose that $g \in x$ and $h \in x$. That is, $[g]_{\mathsf{T}} = [h]_{\mathsf{T}}$, so $\vdash_{\mathsf{T}} g = h$. It follows by (**cong**[]) that $\vdash_{\mathsf{T}} [a]g = [a]h$. Therefore $[[a]g]_{\mathsf{T}} = [[a]h]_{\mathsf{T}}$ as required.
  - Suppose that $a \in \mathbb{A}$ and $b \in \mathbb{A}$ and $a \in y$ and $b \in y$. Thus, $\vdash_{\mathsf{T}} a = b$, so by Corollary 3.26 for every pair of atoms c and d, $\vdash_{\mathsf{T}} c = d$. Now suppose that $g \in x$ and $h \in x$, so $\vdash_{\mathsf{T}} g = h$. It is now routine to construct a derivation of $\vdash_{\mathsf{T}} [a]g = [b]h$. Therefore $[[a]g]_{\mathsf{T}} = [[b]h]_{\mathsf{T}}$ as required.
- $a\#_{sem}\mathcal{T}_{abs}([a]_{\mathsf{T}},[g]_{\mathsf{T}})$ *always.* $a \in [a]_{\mathsf{T}}$ and $g \in [g]_{\mathsf{T}}$ so by definition $\mathcal{T}_{abs}([a]_{\mathsf{T}},[g]_{\mathsf{T}}) = [[a]g]_{\mathsf{T}}$. The result follows by Lemma 4.35.
- $[\![\mathsf{f}]\!]^{\mathcal{T}}$ *is well-defined.* Suppose that $g_1 \in x_1, \ldots, g_n \in x_n$ and $h_1 \in x_1, \ldots, h_n \in x_n$. Thus, $\vdash_{\mathsf{T}} g_i = h_i$ for $1 \leq i \leq n$. The result follows using (**cong**f).

□

Theorem 4.37. *$\mathcal{T}$ is a model of T.*

Proof. Theorem 4.36 states that $\mathcal{T}$ is an interpretation of T. It remains to show that $\mathcal{T}$ validates the axioms.

Suppose $\nabla \vdash t = u$ is an axiom of T. Suppose that ς is a valuation to $|\mathcal{T}|$ and that $a\#_{sem}\varsigma(X)$ for every $a\#X \in \nabla$. We must show that $[\![t]\!]^{\mathcal{T}}_{\varsigma} = [\![u]\!]^{\mathcal{T}}_{\varsigma}$.

Let $\mathcal{X}$ be the set of all unknowns mentioned in ∇, t, or u. By Corollary 4.30, for every $X \in \mathcal{X}$ there is an element $g_X \in \varsigma(X)$ such that $a\#_{sem}g_X$ for every $a\#X \in \nabla$. Let σ be the substitution such that $\sigma(X) = g_X$ when $X \in \mathcal{X}$ and $\sigma(X) = X$ when $X \notin \mathcal{X}$. By Lemma 4.34 $\vdash \nabla\sigma$, and so $\vdash_{\mathsf{T}} t\sigma = u\sigma$ by ($\mathbf{ax}_{\nabla\vdash\mathbf{t=u}}$). Since this derivation does not mention (**cong**d), we know $[t\sigma]_{\mathsf{T}} = [u\sigma]_{\mathsf{T}}$ by Definition 4.28. By an induction on syntax we verify that $[t\sigma]_{\mathsf{T}} = [\![t]\!]^{\mathcal{T}}_{\varsigma}$ and $[u\sigma]_{\mathsf{T}} = [\![u]\!]^{\mathcal{T}}_{\varsigma}$, and the result follows. □

4.4 Completeness for equality derivations

Definition 4.38. For this subsection, fix a signature Σ, a theory $\mathsf{T} = (\Sigma, Ax)$, and terms t, u and a freshness context Δ in signature Σ.

We will show that derivability of $\Delta \vdash_{\mathsf{T}} t = u$ is complete. That is, we will prove:

Theorem 4.39 (Completeness)**.** *If $\Delta \models_{\mathsf{T}} t = u$ then $\Delta \vdash_{\mathsf{T}} t = u$.*

The proof takes up the rest of this subsection.
We shall consider a specific free term model and a specific valuation that preserves sufficient information to allow us to reconstruct a derivation of $\Delta \vdash_{\mathsf{T}} t = u$.

Definition 4.40. Let $\mathcal{X}$ be the unknowns mentioned in Δ, t, u, and let $\mathcal{A}$ be the atoms mentioned in Δ, t, u. For each $X \in \mathcal{X}$:

- let $a_{X1}, \ldots, a_{Xk_X}$ be the atoms in $\mathcal{A}$ (in some arbitary but fixed order) such that $a_{Xi}\#X \notin \Delta$;
- let $\mathsf{d}_X : k_X$ be a term-former.

For each unknown $X \notin \mathcal{X}$, let $\mathsf{d}_X : 0$ be a term-former.
Let D be the set of all d_Xs (so $\mathsf{d}_X \in \mathsf{D}$ for each X).

Definition 4.41. Let σ be the following substitution:

$$\begin{array}{ll} \sigma(X) = \mathsf{d}_X(a_{X1}, \ldots, a_{Xk_X}) & (X \in \mathcal{X}) \\ \sigma(X) = \mathsf{d}_X() & (X \notin \mathcal{X}) \end{array}$$

We consider the free term model $\mathcal{T}$ of T over D (Definition 4.32), and the valuation ς specified by $\varsigma(X) = [\sigma(X)]_{\mathsf{T}}$.

Lemma 4.42. $[t\sigma]_{\mathsf{T}} = [\![t]\!]^{\mathcal{T}}_{\varsigma}$ *and* $[u\sigma]_{\mathsf{T}} = [\![u]\!]^{\mathcal{T}}_{\varsigma}$.

Proof. By an induction on syntax. □

Lemma 4.43. $[\![\Delta]\!]^{\mathcal{T}}_{\varsigma}$ *holds.*

Proof. Suppose $a\#X \in \Delta$. We must show that $a \#_{\text{sem}} \varsigma(X)$.
By construction $X \in \mathcal{X}$ so $\varsigma(X) = [\mathsf{d}_X(a_{X1}, \ldots, a_{Xk_X})]_{\mathsf{T}}$. But also $a \notin \{a_{X1}, \ldots, a_{Xk_X}\}$ by construction so $a \notin \mathsf{d}_X(a_{X1}, \ldots, a_{Xk_X})$. The result follows by Corollary 4.30. □

Definition 4.44. Let Π be a derivation of $\vdash_{\mathsf{T}} t\sigma = u\sigma$ without using (**congd**). By Corollary 3.24 we assume that Π does not contain unknowns or instances of (#**X**) and (**fr**).
Let $\mathcal{A}'$ be $\mathcal{A}$ extended with:

- atoms mentioned anywhere in Π (that were not already in $\mathcal{A}$);
- a set $\mathcal{B}$ of fresh atoms, in bijection with $\mathcal{A}$ — for convenience, we fix a bijection and write b_{Xi} for the atom corresonding under that bijection with a_{Xi} — and
- one fresh atom c (so c does not occur in $\mathcal{A}$, Π, or $\mathcal{B}$).

Let Δ' be Δ extended with freshness assumptions $a'\#X$ for every $X \in \mathcal{X}$ and every $a' \in \mathcal{A}' \setminus \mathcal{A}$.

Definition 4.45. For the rest of this subsection let g and h range over free terms in $\Sigma \cup \mathsf{D}$ that mention only atoms from $\mathcal{A}' \setminus (\mathcal{B} \cup \{c\})$. Define an **inverse mapping** -^{-1} from such free terms to terms in Σ inductively as follows:

$$\begin{array}{rll} a^{-1} & \equiv a & \\ ([a]g)^{-1} & \equiv [a]g^{-1} & \\ \mathsf{f}(g_1, \ldots, g_n)^{-1} & \equiv \mathsf{f}(g_1^{-1}, \ldots, g_n^{-1}) & \\ \mathsf{d}_X(a'_{X1}, \ldots, a'_{Xk_X})^{-1} & \equiv (a'_{X1}\ b_{X1}) \circ \cdots \circ (a'_{Xk_X}\ b_{Xk_X}) \circ (b_{X1}\ a_{X1}) \circ \cdots \circ (b_{Xk_X}\ a_{Xk_X}) \cdot X & (X \in \mathcal{X}) \\ \mathsf{d}_X()^{-1} & \equiv c & (X \notin \mathcal{X}) \end{array}$$

(Here, we relax our permutative convention and permit the possibility that $a'_{Xi} = a_{Xj}$ for some i and j.) We extend the notation -^{-1} to freshnesses and freshness contexts by acting on the terms they mention.

The inverse mapping is equivariant (for the terms we care about):

Lemma 4.46. $\Delta' \vdash_{\mathsf{CORE}} (\pi \cdot g)^{-1} = \pi \cdot g^{-1}$ *when* $\{a \mid \pi(a) \neq a\} \subseteq \mathcal{A}' \setminus (\mathcal{B} \cup \{c\})$.

Proof. By induction on the structure of g. The only non-trivial case is when $g \equiv \mathsf{d}_X(a'_{X1}, \ldots, a'_{Xk_X})$ with $X \in \mathcal{X}$. Then we must show

$$\Delta' \vdash_{\mathsf{CORE}} \pi' \cdot X = \pi'' \cdot X,$$

where we used the following abbreviations:

$$\begin{array}{l} \pi' = (\pi(a'_{X1})\ b_{X1}) \circ \cdots \circ (\pi(a'_{Xk_X})\ b_{Xk_X}) \circ (b_{X1}\ a_{X1}) \circ \cdots \circ (b_{Xk_X}\ a_{Xk_X}) \\ \pi'' = \pi \circ (a'_{X1}\ b_{X1}) \circ \cdots \circ (a'_{Xk_X}\ b_{Xk_X}) \circ (b_{X1}\ a_{X1}) \circ \cdots \circ (b_{Xk_X}\ a_{Xk_X}) \end{array}$$

By the syntactic criteria for CORE derivability (Corollary 3.31) it suffices to show

$$\Delta' \vdash \mathrm{ds}(\pi', \pi'')\#X.$$

The result follows by a case analysis on the atoms in the difference set, using the fact that $\pi(b_{Xi}) = b_{Xi}$. □

Lemma 4.47. *$\Delta' \vdash_{\mathsf{CORE}} t\sigma^{-1} = t$ and $\Delta' \vdash_{\mathsf{CORE}} u\sigma^{-1} = u$.*

Proof. We show $\Delta' \vdash_{\mathsf{CORE}} v\sigma^{-1} = v$ for each subterm v of t and u. We do this by induction on the structure of v. The proof of the case of $v \equiv \pi \cdot X$ is analogous to the $\mathsf{d}_X(a'_{X1}, \ldots, a'_{Xk_X})$ case in the proof of Lemma 4.46. □

Lemma 4.48. *If $\vdash_{\mathsf{T}} t\sigma = u\sigma$ without using (**congd**) then $\Delta \vdash_{\mathsf{T}} t = u$.*
If $\vdash a\#t\sigma$ then $\Delta \vdash a\#t$.

Proof. Suppose we could transform derivations of

$$\vdash a\#t\sigma \qquad \text{or} \qquad \vdash_{\mathsf{T}} t\sigma = u\sigma$$

into derivations of

$$\Delta' \vdash a\#t\sigma^{-1} \qquad \text{or} \qquad \Delta' \vdash_{\mathsf{T}} t\sigma^{-1} = u\sigma^{-1} \quad \text{respectively.}$$

Given this, the result follows because:

- For freshness, by Lemmas 4.47 and 3.33 we deduce $\Delta' \vdash a\#t$. We obtain $\Delta \vdash a\#t$ by part 1 of Lemma 3.25.
- For equality, by Lemma 4.47, (**symm**) and (**tran**) we deduce $\Delta' \vdash_{\mathsf{T}} t = u$ and we obtain $\Delta \vdash_{\mathsf{T}} t = u$ with (**fr**).

Our transformation is inductive on derivations. Suppose the derivation of $\vdash a\#t\sigma$ or $\vdash_{\mathsf{T}} t\sigma = u\sigma$ concludes with an instance of ...

- (#**ab**), (#[]**a**), (#[]**b**), (**refl**), (**symm**), (**tran**) or (**cong**[]). Then the result trivially follows by an instance of the same rule, possibly using the inductive hypothesis.
- (#**X**) or (**fr**). This is impossible by assumption (see Definition 4.44).
- (#**f**). There are three cases to consider:
  - The case of $\vdash a\#\mathsf{f}(g_1, \ldots, g_n)$ for $\mathsf{f} \in \Sigma$.
    It follows that $\vdash a\#g_i$ for $1 \leq i \leq n$, and $\Delta' \vdash a\#g_i^{-1}$ by the inductive hypothesis. We conclude $\Delta' \vdash a\#\mathsf{f}(g_1^{-1}, \ldots, g_n^{-1})$ using (#**f**).
  - The case of $\vdash a\#\mathsf{d}_X(a'_{X1}, \ldots, a'_{Xk_X})$ for $\mathsf{d}_X \in \mathsf{D}$ and $X \in \mathcal{X}$, where a is not necessarily distinct from the a_{Xi} or a'_{Xi}.
    It follows that $\vdash a\#a'_{Xi}$ for $1 \leq i \leq k_X$, and so, examining (#**ab**), it is the case that $a \neq a'_{Xi}$. We must show $\Delta' \vdash a\#\pi \cdot X$, where

    $$\pi = (a'_{X1}\ b_{X1}) \circ \cdots \circ (a'_{Xk_X}\ b_{Xk_X}) \circ (b_{X1}\ a_{X1}) \circ \cdots \circ (b_{Xk_X}\ a_{Xk_X}).$$

    By (#**X**), this follows from $\Delta' \vdash \pi^{-1}(a)\#X$. Since $a \neq a'_{Xi}$ and also $a \neq b_{Xi}$ for all i, we have $\pi^{-1}(a) = (b_{X1}\ a_{X1}) \circ \cdots \circ (b_{Xk_X}\ a_{Xk_X})(a)$. We proceed by a case distinction on a:
    - If $a = a_{Xi}$ for some i, then $(b_{X1}\ a_{X1}) \circ \cdots \circ (b_{Xk_X}\ a_{Xk_X})(a) = b_{Xi}$, and the result follows since $b_{Xi}\#X \in \Delta'$ by construction.
    - If $a \neq a_{Xi}$ for all i, then $(b_{X1}\ a_{X1}) \circ \cdots \circ (b_{Xk_X}\ a_{Xk_X})(a) = a$ since also $a \neq b_{Xj}$ for any j. Then by construction $a\#X \in \Delta$ since the a_{Xi} are the only atoms in $\mathcal{A}$ for which $a_{Xi}\#X \notin \Delta$. The result follows.

  - The case of $\vdash a\#\mathsf{d}_X()$ for $\mathsf{d}_X \in \mathsf{D}$ and $X \notin \mathcal{X}$.
    It is immediate by (#**ab**) that $\vdash a\#c$.
- (**congf**). We consider two cases:
  - The case of $\mathsf{f} \in \Sigma$ follows using the inductive hypothesis.
  - The case of $\mathsf{d} \in \mathsf{D}$ is impossible, since we assumed that Π does not mention (**congd**).
- (**perm**). By the inductive hypothesis we have $\Delta' \vdash a\#g^{-1}$ and $\Delta' \vdash b\#g^{-1}$. Then

$$\Delta' \vdash_{\mathsf{T}} (a\ b) \cdot g^{-1} = g^{-1}$$

  by (**perm**). Using Lemma 4.46, we conclude $\Delta' \vdash_{\mathsf{T}} ((a\ b) \cdot g)^{-1} = g^{-1}$.
- (**ax**$_{\nabla \vdash \mathbf{v}=\mathbf{w}}$). Then $\vdash \nabla^\pi\tau$ and $\vdash_{\mathsf{T}} v^\pi\tau = w^\pi\tau$ for some permutation π and substitution τ such that $\nabla\tau$, $v\tau$ and $w\tau$ do not mention any unknowns. We must show $\Delta' \vdash_{\mathsf{T}} (v^\pi\tau)^{-1} = (w^\pi\tau)^{-1}$. Let τ' be the substitution such that $\tau'(X) = \tau(X)^{-1}$ when $\tau(X) \neq X$ and $\tau'(X) = X$ when $\tau(X) = X$. Then $(v^\pi\tau)^{-1} \equiv v^\pi\tau'$, $(w^\pi\tau)^{-1} \equiv w^\pi\tau'$ and $(\nabla^\pi\tau)^{-1} \equiv \nabla^\pi\tau'$, so it suffices to show $\Delta' \vdash_{\mathsf{T}} v^\pi\tau' = w^\pi\tau'$. By (**ax**$_{\nabla \vdash \mathbf{v}=\mathbf{w}}$), this follows from $\Delta' \vdash \nabla^\pi\tau'$, i.e. $\Delta' \vdash (\nabla^\pi\tau)^{-1}$. By the inductive hypothesis, this follows from the assumption $\vdash \nabla^\pi\tau$.

□

We are now ready for the main result of this subsection:

Proof of Theorem 4.39. Suppose $\Delta \models_{\mathsf{T}} t = u$, so $[\![\Delta \vdash t = u]\!]^{\mathcal{T}}_{\varsigma}$ for the free term model $\mathcal{T}$ and valuation ς constructed above. Now $[\![\Delta]\!]^{\mathcal{T}}_{\varsigma}$ by Lemma 4.43 so $[\![t]\!]^{\mathcal{T}}_{\varsigma} = [\![u]\!]^{\mathcal{T}}_{\varsigma}$. By Lemma 4.42 $[t\sigma]_{\mathsf{T}} = [\![t]\!]^{\mathcal{T}}_{\varsigma}$ and $[u\sigma]_{\mathsf{T}} = [\![u]\!]^{\mathcal{T}}_{\varsigma}$. Therefore by construction $\vdash_{\mathsf{T}} t\sigma = u\sigma$ without using (**congd**). It follows by Lemma 4.48 that $\Delta \vdash_{\mathsf{T}} t = u$. □

4.5 Completeness for freshness

Remark 4.49. Recall that the design of nominal algebra is such that

- atoms a model object-level variable symbols,
- unknowns X model meta-variables, and
- freshnesses $a\#X$ model freshness side-conditions.

We have seen examples of this scheme in the Introduction; it is common in informal practice. This intuition guides the design of the derivation rules for freshness in Figure 1. For example, recall from Example 2.15 the theory LAM. Note that

$$\nvdash a\#(\lambda[a]b)a \quad \text{and} \quad \vdash a\#b \quad \text{and} \quad \vdash_{\mathsf{LAM}} (\lambda[a]b)a = b.$$

This corresponds with the informal judgements 'x is free in the expression $(\lambda x.y)x$', 'x is not a free variable symbol in the syntax y', '$(\lambda x.y)x$ is $\alpha\beta\eta$-convertible with y'.

So derivable freshness, modelling 'not in the free variables of', does not respect derivable equality and cannot be complete for semantic freshness.

Nominal algebra does satisfy an indirect notion of completeness for semantic freshness, which we develop in this subsection (Theorem 4.52), and we return to this issue in Subsection 5.2.

We need a definition:

Definition 4.50. Let $\Delta \vdash a\#t$ be a freshness judgement. Let $S \subseteq \mathbb{A}$ be the collection of atoms appearing in $\Delta \vdash a\#t$ (so $S = \{c \mid \exists Z.c\#Z \in \Delta\} \cup \{a\} \cup \{c \mid c \in t\}$). Now make a fixed but arbitrary choice of fresh atom b (so $b \notin S$).

Write Δ^+ for the freshness context $\Delta, b\#X_1, \ldots, b\#X_n$ where $\{X_1, \ldots, X_n\} = \{X \mid X \in t\}$ (the unknowns mentioned in t).

Now and for the rest of this paper, we will write

$$\Delta^+ \vdash (b\ a) \cdot t = t$$

for the equality judgement obtained from $\Delta \vdash a\#t$ as outlined above.

Lemma 4.51. *Suppose t is a term in a signature Σ and suppose $\mathcal{I}$ is an interpretation of Σ. Then*

$$[\![\Delta \vdash a\#t]\!]^{\mathcal{I}} \quad \textit{if and only if} \quad [\![\Delta^+ \vdash (b\ a) \cdot t = t]\!]^{\mathcal{I}}.$$

Proof. We prove two implications.

Suppose $[\![\Delta \vdash a\#t]\!]^{\mathcal{I}}$ and suppose $[\![\Delta^+]\!]^{\mathcal{I}}_{\varsigma}$ holds. Since $\Delta \subseteq \Delta'$ it follows that $[\![\Delta]\!]^{\mathcal{I}}_{\varsigma}$ holds. By assumption $a\#_{\text{sem}}[\![t]\!]^{\mathcal{I}}_{\varsigma}$. By an induction on syntax using Lemma 4.10 we prove that $b\#_{\text{sem}}[\![t]\!]^{\mathcal{I}}_{\varsigma}$. It follows by part 2 of Lemma 4.3 that $(b\ a) \cdot [\![t]\!]^{\mathcal{I}}_{\varsigma} = [\![t]\!]^{\mathcal{I}}_{\varsigma}$. The result follows by Lemma 4.16.

Assume $[\![\Delta^+ \vdash (b\ a) \cdot t = t]\!]^{\mathcal{I}}$ and suppose $[\![\Delta]\!]^{\mathcal{I}}_{\varsigma}$ holds. By part 1 of Lemma 4.19 we may assume without loss of generality (freshening b if necessary) that $b\#_{\text{sem}}\varsigma(X)$ for every $X \in t$. It follows that $[\![\Delta^+]\!]^{\mathcal{I}}_{\varsigma}$ holds. By assumption

$$[\![(b\ a) \cdot t]\!]^{\mathcal{I}}_{\varsigma} = [\![t]\!]^{\mathcal{I}}_{\varsigma}.$$

By an induction on syntax using Lemma 4.10 we prove that $b\#_{\text{sem}}[\![t]\!]^{\mathcal{I}}_{\varsigma}$. By Corollary 4.4

$$a\#_{\text{sem}}[\![t]\!]^{\mathcal{I}}_{\varsigma} \quad \text{if and only if} \quad (b\ a) \cdot [\![t]\!]^{\mathcal{I}}_{\varsigma} = [\![t]\!]^{\mathcal{I}}_{\varsigma}.$$

The result follows by Lemma 4.16. □

Theorem 4.52. *$\Delta \models_{\mathsf{T}} a\#t$ if and only if $\Delta^+ \vdash_{\mathsf{T}} (b\ a) \cdot t = t$.*

Proof. Direct from Lemma 4.51 and from completeness for equality (Theorem 4.39). □

In words: "semantic freshness in nominal algebra models is captured within the theory of equality".

5 Design alternatives

In this section we consider two design alternatives to nominal algebra as presented in the rest of the paper; we call them N-abs (Subsection 5.1) and N+feq (Subsection 5.2).

We will write $\Delta \vdash^{\text{NA}} a\#t$ and $\Delta \vdash^{\text{NA}}_{\mathsf{T}} t = u$ for the freshness and equality judgements when considered part of nominal algebra, and $\Delta \models^{\text{NA}} a\#t$ and $\Delta \models^{\text{NA}}_{\mathsf{T}} t = u$ for nominal algebra validity; see Definitions 3.10 and 4.23. This is to differentiate from the notions of derivability and validity of N-abs and N+feq which we are about to develop.

5.1 N-abs: nominal algebra without atoms-abstraction

In this subsection we show that atoms-abstraction $[a]t$ is redundant given the rest of the nominal algebra framework. This is expressed formally by Theorem 5.7 and Corollary 5.8.

Let N-abs be the logic derived from nominal algebra by deleting everything to do with atoms-abstraction. That is:

- In Definition 2.4 we delete $[a]t$ from the syntax.
- In Definition 2.13 and 2.14 we only admit t and u in judgements and theories if they are in the restricted syntax (that is, if they do not mention atoms-abstraction), and similarly throughout the rest of the syntax, for example in the syntax of derivations (Figures 1 and 2).
- In Figure 1 we delete (#[]**a**) and (#[]**b**) from the freshness derivation rules.
- In Figure 2 we delete (**cong**[]) from the equality derivation rules.

- In Definition 4.13 we delete $\mathcal{I}_{abs}$ from the notion of an interpretation of a signature, and thus also from the notion of model in Definition 4.20.

We write $\Delta \vdash^{\text{N-abs}} a\#t$ and $\Delta \vdash^{\text{N-abs}}_{\mathsf{T}} t = u$ for the freshness and equality judgements in N-abs, and $\Delta \models^{\text{N-abs}}_{\mathsf{T}} a\#t$ and $\Delta \models^{\text{N-abs}}_{\mathsf{T}} t = u$ for the corresponding notions of validity in N-abs interpretations.

Theorem 5.1. *Derivable freshness in N-abs is sound for the N-abs notion of interpretation and model. Also, derivable equality in N-abs is sound and complete for the N-abs notion of interpretation and model.*

Proof. By a routine modification of the proofs of Theorem 4.24 and 4.39. We merely remove the case of abstraction. We sketch what is required:

In the proof of Theorem 4.24 we delete the cases for $(\#[]\mathbf{a})$, $(\#[]\mathbf{b})$, and $(\mathbf{cong}[])$.

In the proof of Theorem 4.39, which occupies Subsection 4.4, we delete the construction of $\mathcal{T}_{abs}$ in Definition 4.32 and consideration of it in Theorem 4.36, we delete the case of $[a]g$ in Definition 4.45 and the consideration of $(\#[]\mathbf{a})$, $(\#[]\mathbf{b})$, and $(\mathbf{cong}[])$ in Lemma 4.48. □

We intend to exploit Theorems 4.24, 4.39 and 5.1 to give a precise sense in which nominal algebra and N-abs are equivalent. This takes up the rest of this subsection.

Define a translation from nominal algebra signatures to N-abs signatures by:

Definition 5.2. For each nominal algebra signature Σ we make a fixed but arbitrary choice of fresh binary term-former abs (so $\mathsf{abs} \notin \Sigma$). We define $\Sigma' = \Sigma \cup \{\mathsf{abs}\}$.

We define a translation from nominal algebra terms and judgements in Σ to N-abs terms and judgements in Σ' by:

Definition 5.3. We define a translation $\text{-}'$ taking t a nominal algebra term in Σ to t' an N-abs term in Σ', inductively by:

$$a' \equiv a \qquad (\pi \cdot X)' \equiv \pi \cdot X \qquad ([a]t)' \equiv \mathsf{abs}(a, t') \qquad \mathsf{f}(t_1, \ldots, t_n')' \equiv \mathsf{f}(t_1', \ldots, t_n')$$

We extend the translation to judgement forms by defining

$$(\Delta \vdash a\#t)' = (\Delta \vdash a\#t') \quad \text{and} \quad (\Delta \vdash t = u)' = (\Delta \vdash t' = u').$$

We define a translation from nominal algebra theories to N-abs theories by:

Definition 5.4. Given a nominal algebra theory $\mathsf{T} = (\Sigma, Ax)$ let $\mathsf{T}' = (\Sigma', Ax')$ be the N-abs theory such that Ax' has:

- An axiom $\nabla \vdash t' = u'$, for each $(\nabla \vdash t = u) \in Ax$.
- The axiom $b\#X \vdash \mathsf{abs}(b, (b\ a) \cdot X) = \mathsf{abs}(a, X)$.

We define a map from interpretations $\mathcal{I}$ of Σ to interpretations $\mathcal{I}'$ of Σ':

Definition 5.5. Let Σ be a signature. We map from a nominal algebra interpretation $\mathcal{I}$ of Σ to an N-abs interpretation $\mathcal{I}'$ of Σ' by:

- $|\mathcal{I}'| = |\mathcal{I}|$. $\mathcal{I}'_{atm} = \mathcal{I}_{atm}$. $\mathcal{I}'_{\mathsf{f}} = \mathcal{I}_{\mathsf{f}}$ for every $\mathsf{f} \in \Sigma$.
- $\mathcal{I}'_{\mathsf{abs}} = \mathcal{I}_{abs}$.

Lemma 5.6. *Let Σ be a signature. If $\mathcal{I}$ is a NA interpretation of Σ then $\mathcal{I}'$ from Definition 5.5* is *an N-abs interpretation of Σ'.*

Proof. Routine. □

Theorem 5.7. $[\![t']\!]^{\mathcal{I}}_{\varsigma} = [\![t]\!]^{\mathcal{I}'}_{\varsigma}$.
As a corollary, $\mathcal{I}$ is a model of $\mathsf{T} = (\Sigma, Ax)$ if and only if $\mathcal{I}'$ is a model of $\mathsf{T}' = (\Sigma', Ax')$.

Proof. The first part is by a routine induction on the syntax of t. For the second part we prove two implications, starting with the left-to-right implication.

Suppose $\mathcal{I}$ is a model of T. Then $\mathcal{I}$ is a NA interpretation of Σ and $[\![\nabla \vdash t = u]\!]^{\mathcal{I}}$ for every $(\nabla \vdash t = u) \in Ax$. By Lemma 5.6 $\mathcal{I}'$ is a N-abs interpretation of Σ'. We must check that every axiom in Ax' is valid in $\mathcal{I}'$. Fix a valuation ς to $|\mathcal{I}| = |\mathcal{I}'|$. There are two cases:

- *The case of $(\nabla \vdash t' = u') \in Ax'$ because $(\nabla \vdash t = u) \in Ax$.* Suppose that $a \#_{\text{sem}} \varsigma(X)$ for every $a\#X \in \nabla$. By assumption $[\![t]\!]^{\mathcal{I}}_{\varsigma} = [\![u]\!]^{\mathcal{I}}_{\varsigma}$. It follows by the first part of this result that $[\![t']\!]^{\mathcal{I}'}_{\varsigma} = [\![u']\!]^{\mathcal{I}'}_{\varsigma}$.
- *The case of $(b\#X \vdash \mathsf{abs}(b, (b\ a) \cdot X) = \mathsf{abs}(a, X)) \in Ax'$.* We must show that

  $$[\![b\#X \vdash \mathsf{abs}(b, (b\ a) \cdot X) = \mathsf{abs}(a, X)]\!]^{\mathcal{I}'}.$$

  Expanding definitions, we must show that

  $$\text{if } x \in |\mathcal{I}| \text{ and } b \in \mathbb{A} \text{ is such that } b \#_{\text{sem}} x, \text{ then } \quad \mathcal{I}_{abs}(\mathcal{I}_{atm}(b), (b\ a) \cdot x) = \mathcal{I}_{abs}(\mathcal{I}_{atm}(a), x).$$

  By assumption $a \#_{\text{sem}} \mathcal{I}_{abs}(\mathcal{I}_{atm}(a), x)$. Using Lemma 4.10 also $b \#_{\text{sem}} \mathcal{I}_{abs}(\mathcal{I}_{atm}(a), x)$. By part 2 of Lemma 4.3

  $$(b\ a) \cdot \mathcal{I}_{abs}(\mathcal{I}_{atm}(a), x) = \mathcal{I}_{abs}(\mathcal{I}_{atm}(a), x).$$

  The result follows by equivariance of $\mathcal{I}_{abs}$ and $\mathcal{I}_{atm}$.

Conversely, suppose that $\mathcal{I}'$ is a model of T', so that $\mathcal{I}'$ is a N-abs interpretation of Σ, $[\![\nabla \vdash t' = u']\!]^{\mathcal{I}'}$ for every $(\nabla \vdash t = u) \in Ax$, and $[\![b\#X \vdash \mathsf{abs}(b, (b\ a) \cdot X) = \mathsf{abs}(a, X)]\!]^{\mathcal{I}'}$.

Suppose $\nabla \vdash t = u \in Ax$. We must check that $[\![\nabla \vdash t = u]\!]^{\mathcal{I}}$. Fix a valuation ς to $|\mathcal{I}| = |\mathcal{I}'|$ and suppose $a \#_{\text{sem}} \varsigma(X)$ for every $a\#X \in \nabla$. By construction $(\nabla \vdash t' = u') \in Ax'$ so by assumption $[\![t']\!]^{\mathcal{I}'}_{\varsigma} = [\![u']\!]^{\mathcal{I}'}_{\varsigma}$. It follows by the first part of this result that $[\![t]\!]^{\mathcal{I}}_{\varsigma} = [\![u]\!]^{\mathcal{I}}_{\varsigma}$.

We must also check that $\mathcal{I}$ is a NA interpretation of Σ. The only non-trivial part here is to verify that $a \#_{\text{sem}} \mathcal{I}_{abs}(\mathcal{I}_{atm}(a), x)$ always. Choose some $x \in |\mathcal{I}|$ and choose any fresh b (so $b \#_{\text{sem}} x$). By assumption $\mathcal{I}_{abs}(\mathcal{I}_{atm}(b), (b\ a) \cdot x) = \mathcal{I}_{abs}(\mathcal{I}_{atm}(a), x)$. Also, by Lemma 4.10 we have $b \#_{\text{sem}} \mathcal{I}_{abs}(\mathcal{I}_{atm}(b), (b\ a) \cdot x)$. The result follows. □

Corollary 5.8. $\Delta \vdash^{N\text{-}abs}_{\mathsf{T}'} t' = u'$ *if and only if* $\Delta \vdash^{NA}_{\mathsf{T}} t = u$.

Proof. We reason as follows:

$\Delta \vdash^{\text{NA}}_{\mathsf{T}} t = u$	if and only if	$\Delta \models^{\text{NA}}_{\mathsf{T}} t = u$	Theorem 4.24
	if and only if	$\Delta \models^{\text{N-abs}}_{\mathsf{T}'} t' = u'$	Theorem 5.7
	if and only if	$\Delta \vdash^{\text{N-abs}}_{\mathsf{T}'} t' = u'$	Theorem 5.1

□

A proof of Corollary 5.8 is also possible by transforming nominal algebra derivations in T into N-abs derivations in T'. This is not hard, though it is longer to write out.

Remark 5.9. The constructions above, ending with Corollary 5.8, give a precise sense in which atoms-abstraction in nominal algebra is redundant; modulo a trivial translation of syntax we can characterise the same structures, and derive the same judgements, as if our notion of nominal algebra had admitted only atoms a, moderated unknowns $\pi \cdot X$, and term-formers applied to terms $\mathsf{f}(t_1, \ldots, t_n)$.

We included atoms-abstraction $[a]t$ nonetheless. Atoms-abstraction is a conspicuous feature of nominal terms and we expect readers will want to see it given primitive support. However, in

view of the results above we should be aware that it is a *derived* behaviour within what we can consider a minimal nominal algebraic system.

We imagine an analogy may exist here with the status of implication $\supset$, conjunction $\wedge$, and negation $\neg$ in classical and intuitionistic logic. In the classical case $\supset$ can be expressed using $\wedge$ and $\neg$, but not in the intuitionistic case. It may be that in some interesting weakenings of nominal algebra, which have yet to be created, atoms-abstraction might not be expressible using the rest of the system, just as $\supset$ is not expressible using $\wedge$ and $\neg$ in an intuitionistic context. In that case, the proofs in this paper concerning atoms-abstraction will become mathematically independent.

5.2 N+feq: nominal algebra with stronger freshness derivation rules

In Subsection 4.5 we showed how to express freshness in the models using nominal algebra. In this subsection we investigate what happens if we augment nominal algebra with two extra rules (the two rules are (#=) and ($\mathbf{ax}_{\nabla\vdash\mathbf{a\#t}}$) below) designed to make the syntactic freshness # complete for the semantic freshness $\#_{\text{sem}}$; see Theorems 5.17 and 5.20, and Remark 5.18.

As it turns out, the result is a system with *less* expressive power, though some derivable judgements can be derived more succinctly. See Remarks 5.18 and 5.19.

Definition 5.10. Let N+feq have the syntax and judgements of nominal algebra (Definitions 2.4 and 2.13).

We then make the following changes:

- The notion of a *theory* (Definition 2.14) is augmented; we allow equality *and freshness* axioms.

  That is, we take $\mathsf{T} = (\Sigma, Ax)$ where Σ is a signature and Ax is a possibly infinite set of equality or freshness judgements.

- We augment the derivation rules for freshness (Figure 1) with the following two rules:

$$\frac{a\#t \quad t=u}{a\#u}\,(\#=) \qquad\qquad \frac{\nabla^{\pi}\sigma}{\pi(a)\#t^{\pi}\sigma}\,(\mathbf{ax}_{\nabla\vdash\mathbf{a\#t}})$$

- In our notion of derivability (Definition 3.10) we write $\Delta \vdash_{\mathsf{T}} a\#t$ instead of $\Delta \vdash a\#t$ (because now, the derivability of a freshness can depend on axioms and derivable equalities). We insist that the derivation of $\Delta \vdash_{\mathsf{T}} a\#t$ should mention only terms in the signature of T, and use only instances of ($\mathbf{ax}_{\nabla\vdash\mathbf{a\#t}}$) or ($\mathbf{ax}_{\nabla\vdash\mathbf{t=u}}$) where $\nabla \vdash a\#t$ or $\nabla \vdash t=u$ respectively is an axiom in T.

It is routine to extend the proof-theoretical results of Subsection 3.3 to N+feq: Lemma 3.20 acquires a second case: If $\Delta \vdash_{\mathsf{T}} a\#t$ then $\Delta \vdash_{\mathsf{T}^{\pi}} a\#t$. In Theorem 3.21 we write 'if $\Delta \vdash_{\mathsf{T}} a\#t$ then $\Delta^{\pi} \vdash_{\mathsf{T}} \pi(a)\#t^{\pi}$' (note the $\vdash_{\mathsf{T}}$ instead of $\vdash$). In Theorem 3.22 we write 'if $\Delta \vdash_{\mathsf{T}} a\#t$ then $\Delta \vdash_{\mathsf{T}} \pi(a)\#\pi\cdot t$'. In Theorem 3.23 we write 'if $\Delta \vdash_{\mathsf{T}} a\#t$ then $\Delta' \vdash_{\mathsf{T}} a\#t\sigma$'. Corollary 3.24 acquires a corresponding case for $\vdash_{\mathsf{T}} a\#t$. In Lemma 3.25 $\vdash$ becomes $\vdash_{\mathsf{T}}$. The results of Subsection 3.3.3 about the theory CORE are not affected, since CORE has no axioms and Corollary 3.33 proves that (#=) is admissible in that theory.

The syntax and judgements of N+feq are identical to those of nominal algebra, so the notions of of interpretation, valuation, model, and validity are not greatly affected: The notions of interpretation and valuation (Definitions 4.13 and 4.14) are unchanged. The notion of model (Definition 4.20) acquires an extra clause, because theories can mention freshness axioms:

$$[\![\nabla \vdash a\#t]\!]^{\mathcal{I}} \quad \text{for all axioms } \nabla \vdash a\#t \text{ of } \mathsf{T}.$$

The notion of validity is unchanged from Definition 4.18.

We can then prove Theorem 5.11; derivable freshness and equality in N+feq is sound for the nominal sets semantics:

Theorem 5.11 (Soundness)**.** *Suppose* T *is an N+feq theory.*

1. *If* $\Delta \vdash^{N+feq}_{\mathsf{T}} a\#t$ *then* $\Delta \models^{N+feq}_{\mathsf{T}} a\#t$.
2. *If* $\Delta \vdash^{N+feq}_{\mathsf{T}} t = u$ *then* $\Delta \models^{N+feq}_{\mathsf{T}} t = u$.

Proof. The proof used in Theorem 4.24 carries through without any changes except that we add cases for (#=) and ($\mathbf{ax}_{\nabla\vdash \mathbf{a\#t}}$):

- (#=). $a\#_{\mathsf{sem}}[\![t]\!]^{\mathcal{I}}_{\varsigma}$ and $[\![t]\!]^{\mathcal{I}}_{\varsigma} = [\![u]\!]^{\mathcal{I}}_{\varsigma}$ imply that $a\#_{\mathsf{sem}}[\![u]\!]^{\mathcal{I}}_{\varsigma}$.
- ($\mathbf{ax}_{\nabla\vdash \mathbf{a\#t}}$). Suppose $[\![\nabla^{\pi}\sigma]\!]^{\mathcal{I}}_{\varsigma}$. Then $\pi(a)\#_{\mathsf{sem}}[\![\sigma(X)]\!]^{\mathcal{I}}_{\varsigma}$ holds for all $a\#X \in \nabla$. By part 3 of Lemma 4.3 also $a\#_{\mathsf{sem}}\pi^{-1}\cdot[\![\sigma(X)]\!]^{\mathcal{I}}_{\varsigma}$ for all $a\#X \in \nabla$. Let ς' be defined as

  $$\varsigma'(X) = \pi^{-1}\cdot[\![\sigma(X)]\!]^{\mathcal{I}}_{\varsigma} \quad \text{for every } X.$$

  Then $a\#_{\mathsf{sem}}\varsigma'(X)$ for all $a\#X \in \nabla$, so $[\![\nabla]\!]^{\mathcal{I}}_{\varsigma'}$ holds. Since $\nabla \vdash a\#t$ is an axiom of T, we know $a\#_{\mathsf{sem}}[\![t]\!]^{\mathcal{I}}_{\varsigma'}$. Then by part 3 of Lemma 4.3 also $\pi(a)\#_{\mathsf{sem}}\pi\cdot[\![t]\!]^{\mathcal{I}}_{\varsigma'}$, and by Lemma 4.16 we obtain $\pi(a)\#_{\mathsf{sem}}[\![\pi\cdot t]\!]^{\mathcal{I}}_{\varsigma'}$. By a straightforward induction on syntax we can verify that $[\![\pi\cdot t]\!]^{\mathcal{I}}_{\varsigma'} = [\![t^{\pi}\sigma]\!]^{\mathcal{I}}_{\varsigma}$, so $\pi(a)\#_{\mathsf{sem}}[\![t^{\pi}\sigma]\!]^{\mathcal{I}}_{\varsigma}$ as required.

□

Remark 5.12. Recall LAM from Example 2.15 and recall the discussion and examples from Remark 4.49. In N+feq we derive

$$\vdash^{\mathrm{N+feq}}_{\mathsf{LAM}} a\#\mathsf{app}(\mathsf{lam}([a]b), a) \quad \text{as follows:} \quad \dfrac{\dfrac{\dfrac{}{b[a\mapsto a]=b}\,(\mathbf{id}{\mapsto})}{(\lambda[a]b)a=b}\,(\beta) \qquad \dfrac{}{a\#b}\,(\#\mathbf{ab})}{a\#(\lambda[a]b)a}\,(\#=).$$

Of course, x is free in the λ-calculus expression $(\lambda x.y)x$; intuitively the derivation above works by reducing it to y and then proving that x is not free in the syntax y.

Thus, $\Delta \vdash^{\mathrm{N+feq}}_{\mathsf{T}} a\#t$ is unsound for the informal interpretation 'is not a free variable symbol in the syntax', because this is not in general respected by derivable equality. Instead, it is sound for $\#_{\mathsf{sem}}$, as we formally observed in Theorem 5.11. In Theorem 5.20 we shall also prove it complete. So # in N+feq corresponds with the intuition 'is not in the support of the denotation of the syntax' or perhaps (trying to relate this back to informal practice) 'is not free in the syntax of a derivably equal term'; this is respected by derivable equality by construction.

Recall from Definition 4.50 the specification of $\Delta^{+} \vdash (b\ a)\cdot t = t$ from $\Delta \vdash a\#t$.

Definition 5.13. We define a map -′ from N+feq judgements to nominal algebra judgements as follows:

- $\Delta \vdash t = u$ maps to $\Delta \vdash t = u$.
- $\Delta \vdash a\#t$ maps to $\Delta^{+} \vdash (b\ a)\cdot t = t$.

We map an N+feq theory $\mathsf{T} = (\Sigma, Ax)$ to a nominal algebra theory $\mathsf{T}' = (\Sigma, Ax')$ such that

$$Ax' = \{A' \mid A \in Ax\}.$$

(Recall that an N+feq theory is allowed to mention freshness axioms $\nabla \vdash a\#t$. A nominal algebra theory is not, and note that $(\nabla \vdash a\#t)' = (\nabla^{+} \vdash (b\ a)\cdot t = t)$.)

Definition 5.13 is correct in the following sense:

Lemma 5.14. *Suppose* $\mathsf{T} = (\Sigma, Ax)$ *is a theory and* $\mathcal{I}$ *is an interpretation of* Σ*. Then* $\mathcal{I}$ *is an N+feq model of* T *if and only if* $\mathcal{I}$ *is a nominal algebra model of* T'*.*

Proof. Suppose that $\mathcal{I}$ is an N+feq model of T. We must show that

- for every $(\nabla \vdash a\#t) \in Ax$, $[\![\nabla^+ \vdash (b\ a)\cdot t = t]\!]^{\mathcal{I}}$, and
- for every $(\nabla \vdash t = u) \in Ax$, $[\![\nabla \vdash t = u]\!]^{\mathcal{I}}$.

The first part is by Lemma 4.51. The second part is immediate. The reverse implication is similar. □

Lemma 5.15. *Suppose* T *is an N+feq theory.*

1. *If* $\Delta \vdash^{NA}_{\mathsf{T}'} t = u$ *is derivable then so is* $\Delta \vdash^{N+feq}_{\mathsf{T}} t = u$.
2. *If* $\Delta^+ \vdash^{NA}_{\mathsf{T}'} (b\ a)\cdot t = t$ *is derivable then so is* $\Delta \vdash^{N+feq}_{\mathsf{T}} a\#t$.

Proof. The first part is by a routine induction on derivations. The only interesting case is when the derivation uses $(\mathbf{ax}_{\nabla^+\vdash(\mathbf{b\ a})\cdot\mathbf{t}=\mathbf{t}})$, for $(\nabla^+ \vdash (b\ a)\cdot t = t) \in Ax'$ because $(\nabla \vdash a\#t) \in Ax$. We translate this into N+feq according to the following sketch:

$$\dfrac{\dfrac{\nabla^{+\pi}\sigma}{\vdots}}{\pi(b)\#t^\pi\sigma} \qquad \dfrac{\dfrac{\vdots}{\nabla^\pi\sigma}}{\pi(a)\#t^\pi\sigma}\,(\mathbf{ax}_{\nabla\vdash\mathbf{a}\#\mathbf{t}})}{(\pi(b)\ \pi(a))\cdot t^\pi\sigma = t^\pi\sigma}\,(\mathbf{perm})$$

For the second part, we observe that by the first part $\Delta^+ \vdash^{\mathrm{N+feq}}_{\mathsf{T}} (b\ a)\cdot t = t$. The result then follows using freshness derivation rules, (#=), and (**fr**). □

Lemma 5.16 proves the reverse implication to Lemma 5.15. We could also prove Lemma 5.16 by manipulating derivations using an argument similar to, but more complex than, that used to prove Lemma 5.15.[7] With the results we have already proved, an argument on models is shorter and sweeter:

Lemma 5.16. *Suppose* T *is an N+feq theory.*

- *If* $\Delta \vdash^{N+feq}_{\mathsf{T}} t = u$ *is derivable then so is* $\Delta \vdash^{NA}_{\mathsf{T}'} t = u$.
- *If* $\Delta \vdash^{N+feq}_{\mathsf{T}} a\#t$ *is derivable then so is* $\Delta^+ \vdash^{NA}_{\mathsf{T}'} (b\ a)\cdot t = t$.

Proof. Suppose $\Delta \vdash^{\mathrm{N+feq}}_{\mathsf{T}} t = u$ is derivable. By Theorem 5.11 also $\Delta \models^{\mathrm{N+feq}}_{\mathsf{T}} t = u$. By Lemma 5.14 this is equivalent to $\Delta \models^{\mathrm{NA}}_{\mathsf{T}'} t = u$. By Theorem 4.39 $\Delta \vdash^{\mathrm{NA}}_{\mathsf{T}'} t = u$.

Suppose $\Delta \vdash^{\mathrm{N+feq}}_{\mathsf{T}} a\#t$. By Theorem 5.11 $\Delta \models^{\mathrm{N+feq}}_{\mathsf{T}} a\#t$. By Lemma 5.14 this is equivalent to $\Delta^+ \models^{\mathrm{NA}}_{\mathsf{T}'} a\#t$. By Theorem 4.52 $\Delta^+ \vdash^{\mathrm{NA}}_{\mathsf{T}'} (b\ a)\cdot t = t$. □

Theorem 5.17. *Suppose* T *is an N+feq theory.*

- $\Delta \vdash^{N+feq}_{\mathsf{T}} t = u$ *if and only if* $\Delta \vdash^{NA}_{\mathsf{T}'} t = u$.
- $\Delta \vdash^{N+feq}_{\mathsf{T}} a\#t$ *if and only if* $\Delta^+ \vdash^{NA}_{\mathsf{T}'} (b\ a)\cdot t = t$.

Proof. From Lemmas 5.16 and 5.15. □

Remark 5.18. A word on what Theorem 5.17 means relative to Theorem 4.52.

- Theorem 4.52 gives a precise sense in which an explicit semantic freshness judgement form is redundant in nominal algebra; it is already captured in equality.
- Theorem 5.17 gives a precise sense in which extra derivation rules (#=) and $(\mathbf{ax}_{\nabla\vdash\mathbf{a}\#\mathbf{t}})$ are also redundant.

[7] We transform instances of $(\mathbf{ax}_{\nabla\vdash\mathbf{a}\#\mathbf{t}})$ into instances of $(\mathbf{ax}_{\nabla^+\vdash(\mathbf{b\ a})\cdot\mathbf{t}=\mathbf{t}})$ followed by (#=) and (**fr**), and we then commute instances of (#=) and (**fr**) down the derivation.

In fact we can say a little more: the addition of (#=) loses some expressive power; we noted in Remark 5.12 in N+feq the intuition of # meaning 'is not a free variable symbol in the syntax' is unsound, and this model of meaning for #, corresponding with freshness side-conditions in informal practice, is destroyed by (#=) and cannot be recovered.

Remark 5.19. Note that a logic 'with more rules' does not necessarily mean a logic 'with more expressivity': Consider a first-order predicate logic with at least one constant c, and an extra derivation rule $\dfrac{}{t = u}\,(\mathbf{AllEqual})$. We can derive more entailments in this new logic, but it is a fact that up to a simple syntactic translation the resulting derivation system is equivalent to propositional logic.

It is easy to generate other examples of this. An extreme case is the rule $\dfrac{}{\bot}\,(\mathbf{Triv})$ which certainly lets us derive more entailments, but up to a simple syntactic translation the resulting derivation system is equivalent to a 'logic' with one predicate $\top$, and no logical connectives or derivation rules!

In a similar way, adding (#=) lets us derive more sequents but in the sense we have made formal, the resulting logic is strictly less expressive than what we started with.

Given our results so far, it is easy to leverage nominal algebra completeness to a nice corollary:

Theorem 5.20 (Completeness)**.** *Suppose* T *is an N+feq theory.*

- *If* $\Delta \models_{\mathsf{T}}^{N+feq} t = u$ *then* $\Delta \vdash_{\mathsf{T}}^{N+feq} t = u$.
- *If* $\Delta \models_{\mathsf{T}}^{N+feq} a\#t$ *then* $\Delta \vdash_{\mathsf{T}}^{N+feq} a\#t$.

Proof. From Theorem 5.17 and nominal algebra completeness (Theorem 4.39). □

6 Conclusions

Nominal terms embrace the difference between the object-level and the meta-level. There are two classes of variables, atoms a and unknowns X. Substitution for X can capture abstractions by a. For instance the syntactic identity

$$(\lambda[a]X)\sigma \equiv \lambda[a]a$$

where $\sigma(X) \equiv a$ formally reflects the informal sentence

$$\text{instantiate } e \text{ to } x \text{ in } \lambda x.e\text{; obtain } \lambda x.x.$$

Freshnesses like $a\#X$ capture the habitual side-conditions which come with such statements, and correspond with 'x is not a free variable symbol in the syntax t'.

We present nominal algebra as a logical model which is 'ϵ away from' the informal meta-level of equality *with binding*, and also 'ϵ away from' universal algebra. We have seen how the flavour of both universal algebra and of informal practice is maintained in the nominal algebra setting, while at the same time being completely rigorous.

As discussed in the Introduction, 'algebra' is used in several senses in the literature. For us, 'universal algebra' means 'the logic of equality and nothing else', as presented for example in [BS81].

From a wider perspective, this paper pursues a programme by the authors arguing that *names* can be studied as mathematical entities in both new logics and in new denotations. From that point of view, nominal algebra is a rigorous universal algebraic system in which term-formers can be given properties (just as in universal algebra) — and so can atoms. For example, SUB from Example 2.15 can be read as 'names, with a capture-avoiding substitution action'.

6.1 Related work

6.1.1 Previous work on nominal algebra

The first work to consider arbitrary theories of equality on nominal terms was *nominal rewriting* [FGM04, FG07] (we can think of a rewrite as a directed equality).

The outline of nominal algebra itself was first presented in a workshop [GM06b] and accompanying technical report [GM06c]; this was followed by a conference paper [GM07] and by the second author's thesis [Mat07]. Nominal algebra has also been used in the following applications:

- We axiomatise and study capture-avoiding substitution [GM06a, GM08a].
- We axiomatise the λ-calculus [GM09a, GM08b, GM09b].
- We axiomatise first-order logic [GM06d, GM08c] (and we use this axiomatisation to develop a sequent-style proof-theory for a variant of first-order logic with explicit meta-variables standing for 'unknown predicates').

Also, and most technically challenging, is [Gab09]. This proves that nominal algebra satisfies a form of the HSP theorem (also called Birkhoff's theorem). The HSP theorem is a fundamental theorem of universal algebra. Proving a version of it for nominal algebra gives a precise and powerful sense in which nominal algebra *is* a universal algebra system in the classic sense of the term.

This paper is a journal version of the conference paper [GM07], and extends and improves on the relevant parts of [Mat07]. Full proofs are included, the presentation is revised and extended, and we give several new mathematical results which indicate the design space within which nominal algebra exists, and stating formally in what senses these are all equivalent.

6.1.2 Nominal equational logic and nominal logic

Since the conception of nominal algebra, Clouston and Pitts presented *nominal equational logic* (**NEL**) [CP07]. This corresponds roughly with N+feq from Subsection 5.2 plus a sorting system similar to that used in [UPG04, FG07], though there are also differences at the level of syntax (notably, nominal equational logic does not use nominal terms, and atoms and abstractions are modelled using families of term-formers indexed by an infinite collection of atoms). Note also that NEL conflates freshness and equality judgements into a single judgement form $\Delta \vdash a\#t = u$, but this difference is inessential. Using the notation in this paper, we can draw up a small table of what meanings are given to the different judgements in the two systems:

- Equality judgements $\Delta \vdash t = u$.
  - Nominal algebra. Derivable equality $=$ corresponds with semantic equality, also written $=$.
  - NEL. Derivable equality $=$ corresponds with semantic equality, also written $=$.
- Freshness judgements $\Delta \vdash a\#t$.
  - Nominal algebra. Derivable freshness $\#$ corresponds with 'not free in', written $\notin fv$.
  - NEL. Derivable freshness $\#$ corresponds with semantic freshness, written $\#_{\text{sem}}$.

In the conclusions of their paper [CP07] Clouston and Pitts observe that nominal algebra freshness is not complete for freshness in the model $\#_{\text{sem}}$. This observation is correct as far as it goes, but it leaves several things unsaid:

- Conversely, NEL freshness judgements are unsound for 'not free in' (see Remarks 4.49, 5.12, and 5.18).
- In Subsection 4.5 we demonstrate how to soundly and completely express semantic freshness $\#_{\text{sem}}$ using equality in nominal algebra (Theorem 4.52). Thus, semantic freshness is redundant in nominal algebra.

  This observation is not new. The original equation for defining semantic freshness [GP01, equation 13, page 8] uses semantic equality plus the Gabbay-Pitts И quantifier which, from the point of view of this paper, looks just like the 'fresh b' in Δ^+ in Definition 4.50. (This expands on Theorem 5.5 of the conference version [GM07].)

- In Subsection 5.2 we demonstrate how adding extra deductive power to make derivable freshness # complete for semantic freshness $\#_{\text{sem}}$, is also redundant (Theorems 5.17 and 5.20).

  Indeed, this actually loses some expressivity (Remark 5.18). Put another way, it is not the case that versions of Theorems 4.52 and 5.17 hold in the 'reverse direction'; N+feq is expressible in terms of nominal algebra, but not vice versa.

- It may also be worth noting that NEL freshness is in general undecidable, whereas nominal algebra freshness is always decidable, because the rules, presented in Figure 1, are syntax-directed.

  Related to this is an observation by theoreticians in process algebras, that the set of atoms which can take part in the behaviour of a process (in our terminology; the atoms which are not semantically fresh for their particular notion of semantics), is undecidable [BW90].

We should mention the caveat that N+feq is not NEL[8] but to the level of detail of the comment '# is not complete for $\#_{\text{sem}}$', comparing N+feq with nominal algebra is adequate to argue the point. Modulo this caveat it seems to us that nominal algebra, compared with nominal equational logic, is simpler and more compact, and that it may offer easier syntax, semantics, and proofs for mathematical study[9] — or perhaps we should say this rather of N-abs, since abs is also redundant given the rest of the nominal algebra framework (as discussed in Subsection 5.1).

Concerning applications, at the time of writing there is no NEL analogue of our applications for nominal algebra [GM06a, GM08a, GM09a, GM09b, GM06d, GM08c, Gab09]; investigating these results for NEL would be future work. Of course, one way to obtain these results cheaply might be via a translation to nominal algebra based on the development in Subsection 5.2.

Note that nominal algebra is not 'just' the equality fragment of nominal logic [Pit03]. Such a fragment would satisfy (#=) (Definition 5.10), and nominal algebra does not.

Since this paper was written, Fiore and Hur have completed a general study of nominal and other equational logics, using the language of categories. This places nominal algebra in a general context [FH08]. It would also be interesting to express the ideas in this paper in a format directly modelled on Lawvere theories [Law63] (see [HP07] for an interesting discussion with references).

6.1.3 Higher-order techniques

At the heart of nominal algebra is the capturing substitution of unknowns for terms. There is a whole other thread of research devoted to systems based on capture-avoiding substitution and/or $\alpha\beta\eta$-equivalence (though not all are necessarily algebraic ones). The theory of contexts [Mic01] can be used to axiomatise systems with binding. So, differently, can higher-order algebra [Mei92]. So indeed can simply-typed λ-calculus [Bar00]. These systems are different and intended for different purposes but they share a core which is in essence simply-typed λ-calculus expressions up to $\alpha\beta\eta$-equivalence. Just as is the case for nominal terms this richer term-language gives more expressivity, which can be used to give stronger axioms. This inherits the distinctive capture-avoiding substitution which for us is not a direct model of the behaviour seen at the informal meta-level as discussed in the Introduction. See [GP01, Subsection 1.1] and [Pit03, Section 9] for excellent discussions.

We note that in [GJ02, Joj04], Geuvers and Jojgov extend higher-order logic with explicit meta-variables. Although the approach of their oHOL language is similar to ours, there are some fundamental differences: the default notion of instantiation of meta-variables in oHOL is capture-avoiding; capturing instantiation can be achieved by parameterising the meta-variable. Also, meta-variables are equipped with pending substitutions of object-variables in oHOL. In nominal

[8] NEL syntax tends to have 'more term-formers' in a style which we find quite reminiscent of cylindric techniques, discussed below; see [CP07, Remark 3.2] (note that what we call term-formers are in NEL terminology called *operation symbols*). Nominal algebra uses nominal terms' syntax and follows [UPG04, FG07]. Also, we have also not considered a sort system for the nominal terms syntax; we would expect that imposing a sort system like that of [UPG04, FG07] to be routine.

[9] ... albeit one which may need sugar to be palatable to the user of an implemented system, but this is just as true of most other mathematically convenient systems.

algebra meta-variables are equipped with α-renamings of object-variables. Note that we could record these substitutions by using the explicit substitutions from theory SUB (Example 2.15).

6.1.4 Binding algebras

Sun's binding algebras [Sun99] are based on a functional semantics for binding, whereas we work according to the relatively newer nominal semantics which is decidedly non-functional; currently the two strands are essentially independent and it remains to see what ideas might flow between them.

Fiore, Plotkin and Turi's binding algebras [FPT99] use categories of presheaves, whereas we use nominal sets. Categories of presheaves do not have a notion of "least supporting set" like nominal sets do [GP01, Related Work]. So in its current form, the freshness judgement of $a\#x$ cannot be expressed in their framework. For this reason, it is not clear how an easy and direct connection can be made between the two frameworks.

6.1.5 Cylindric techniques and combinatory techniques

A host of 'cylindric' algebraic techniques exist. These embrace meta-variables and reject object-level variables, preferring to encode their expressive power in the term-formers. Examples are lambda-abstraction algebras [Sal00] for the λ-calculus and cylindric algebras [BS81, ANS01] for first-order logic. Combinators [Bar84] reject object-level variables altogether. These systems are effective for their applications, but we do not see that they naturally represent equalities with binding and meta-variables, from the simple fact that there are no object-variables.

6.2 Future work

There is much possible future work.

6.2.1 Axiomatisations of other systems

We have seen examples of three fundamental systems axiomatised in nominal algebra: (capture-avoiding) substitution, the λ-calculus, and first-order logic. In [GM06a, GM08a, GM06d, GM08c] we study the axiomatisations of substitution and first-order logic. We would like to employ the methods of this paper to formalise reasoning on other systems with binding. We speculate on applications to axiomatising substructural logics with quantifiers [Res99]; also to process calculi, some of which feature complex binding side-conditions and for which algebraic reasoning principles are of interest [AG97, Lut02, KD02]; and perhaps even to logics for state such as Bunched Implications and Separation Logic [OP99, Rey02] where atoms (with suitable axioms) could represent locations and freshness might then nicely express separation. To this end an enriched freshness judgement $t'\#t$ could be useful, generalising $a\#t$ and asserting a separation between the atoms not derivably fresh in t and t'.

Calculi of explicit substitution represent the process of substitution and so represent different aspects of name structure explicitly depending on the particular system [Les94]. A discussion exists of how one might use nominal rewriting to express rewrite systems for λ-calculi with explicit substitutions [FG07, Section 9]; enriching freshness contexts with other judgements would allow reduction strategies to be expressed as part of the rewrite rules. It would also be interesting to attempt nominal algebra or nominal rewriting axiomatisations of λ-calculi with non-standard treatments of binding, such as Adbmal [HvO03].

6.2.2 Hierarchies of variables

We are interested in developing logics with *hierarchies* of 'increasingly meta-'variables. Since nominal algebra offers two levels of variable, why not extend this to allow an infinite hierarchy of variables, by analogy with type hierarchies in the λ-calculus [Bar84]? Work has already started in this direction by extending nominal terms with a hierarchy of variables [Gab05, Gab07c, GL08].

6.2.3 Implementation

We are interested in exploring how well nominal algebra could serve as the basis for an implementation of an interactive proof assistent. This may be useful because, as discussed in the Introduction, nominal algebra permits a particularly direct translation between informal mathematical practice and formal syntax. The algorithmic properties of nominal terms remain to be explored; some work in that direction is [Che04].

References

[AG97] Martín Abadi and Andrew D. Gordon. A calculus for cryptographic protocols: the spi calculus. In *CCS '97: Proc. of the 4th ACM conf. on Computer and Communications Security*, pages 36–47. ACM Press, 1997.

[ANS01] H. Andréka, I. Németi, and I. Sain. Algebraic logic. In D.M. Gabbay and F. Guenthner, editors, *Handbook of Philosophical Logic, 2nd Edition*, volume 2, pages 133–249. Kluwer, 2001.

[Bar75] Jon Barwise. *Admissible Sets and Structures: an approach to definability theory.* Perspectives in mathematical logic. Springer, 1975.

[Bar84] H. P. Barendregt. *The Lambda Calculus: its Syntax and Semantics (revised ed.).* North-Holland, 1984.

[Bar00] H. P. Barendregt. Lambda calculi with types. In S. Abramsky, D. M. Gabbay, and T. S. E. Maibaum, editors, *Handbook of Logic in Computer Science, Volume 2*, pages 117–309. OUP, 2000.

[Bru96] Norbert Brunner. 75 years of independence proofs by Fraenkel-Mostowski permutation models. *Mathematica Japonica*, 43:177–199, 1996.

[BS81] S. Burris and H. Sankappanavar. *A Course in Universal Algebra.* Graduate texts in mathematics. Springer, 1981.

[BW90] J. C. M. Baeten and W. P. Weijland. *Process Algebra*, volume 18 of *Cambridge Tracts in Theoretical Computer Science.* Cambridge University Press, 1990.

[CF58] Haskell B. Curry and R. Feys. *Combinatory Logic*, volume 1. North Holland, 1958.

[Che04] James Cheney. The complexity of equivariant unification. In *Proc. 31st Int'l Colloquium on Automata, Languages and Programming (ICALP 2004)*, volume 3142 of *Lecture Notes in Computer Science*, pages 332–344. Springer, 2004.

[Che06] James Cheney. Completeness and Herbrand theorems for nominal logic. *Journal of Symbolic Logic*, 71:299–320, 2006.

[CP07] Ranald A. Clouston and Andrew M. Pitts. Nominal equational logic. *Electronic Notes in Theoretical Computer Science*, 172:223–257, 2007.

[dB91] N.G. de Bruijn. Checking mathematics with computer assistance. *Notices of the American Mathematical Society (AMS)*, 38(1):8–15, 1991.

[FG07] Maribel Fernández and Murdoch J. Gabbay. Nominal rewriting. *Information and Computation*, 205(6):917–965, 2007.

[FGM04] Maribel Fernández, Murdoch J. Gabbay, and Ian Mackie. Nominal Rewriting Systems. In *Proc. 6th Int. ACM SIGPLAN Conf. on Principles and Practice of Declarative Programming (PPDP'2004)*, pages 108–119. ACM Press, 2004.

[FH08] Marcelo Fiore and Chung-Kil Hur. Term equational systems and logics. *Electronic Notes in Theoretical Computer Science*, 218:171–192, 2008.

[FPT99] Marcelo P. Fiore, Gordon D. Plotkin, and Daniele Turi. Abstract syntax and variable binding. In *LICS '99: 14th Annual Symposium on Logic in Computer Science*, pages 193–202. IEEE, 1999.

[Gab00] Murdoch J. Gabbay. *A Theory of Inductive Definitions with alpha-Equivalence*. PhD thesis, Cambridge, UK, 2000.

[Gab05] Murdoch J. Gabbay. A NEW calculus of contexts. In *PPDP '05: Proc. of the 7th ACM SIGPLAN symposium on Principles and Practice of Declarative Programming*, pages 94–105. ACM, 2005.

[Gab07a] Murdoch J. Gabbay. Fresh Logic. *Journal of Applied Logic*, 5(2):356–387, June 2007.

[Gab07b] Murdoch J. Gabbay. A General Mathematics of Names. *Information and Computation*, 205(7):982–1011, July 2007.

[Gab07c] Murdoch J. Gabbay. Hierarchical Nominal Terms and Their Theory of Rewriting. *Electronic Notes in Theoretical Computer Science*, 174(5):37–52, 2007.

[Gab09] Murdoch J. Gabbay. Nominal algebra and the HSP theorem. *Journal of Logic and Computation*, 19(2):341–367, 2009.

[GJ02] Herman Geuvers and Gueorgui I. Jojgov. Open proofs and open terms: A basis for interactive logic. In *Computer Science Logic: 16th International Workshop*, pages 537–552, 2002.

[GL08] Murdoch J. Gabbay and Stéphane Lengrand. The lambda-context calculus. *Electronic Notes in Theoretical Computer Science*, 196:19–35, 2008.

[GM06a] Murdoch J. Gabbay and Aad Mathijssen. Capture-avoiding Substitution as a Nominal Algebra. In *ICTAC 2006: Theoretical Aspects of Computing*, volume 4281 of *Lecture Notes in Computer Science*, pages 198–212, 2006.

[GM06b] Murdoch J. Gabbay and Aad Mathijssen. Nominal Algebra. In *18th Nordic Workshop on Programming Theory*, 2006.

[GM06c] Murdoch J. Gabbay and Aad Mathijssen. Nominal Algebra. Technical Report HW-MACS-TR-0045, Heriott-Watt, 2006.

[GM06d] Murdoch J. Gabbay and Aad Mathijssen. One-and-a-halfth-order logic. In *PPDP '06: Proc. of the 8th ACM SIGPLAN symposium on Principles and Practice of Declarative Programming*, pages 189–200. ACM, 2006.

[GM07] Murdoch J. Gabbay and Aad Mathijssen. A Formal Calculus for Informal Equality with Binding. In *WoLLIC'07: 14th Workshop on Logic, Language, Information and Computation*, volume 4576 of *Lecture Notes in Computer Science*, pages 162–176. Springer, 2007.

[GM08a] Murdoch J. Gabbay and Aad Mathijssen. Capture-Avoiding Substitution as a Nominal Algebra. *Formal Aspects of Computing*, 20(4-5):451–479, June 2008.

[GM08b] Murdoch J. Gabbay and Aad Mathijssen. A nominal axiomatisation of the lambda calculus. Technical Report 08-18, Technische Universiteit Eindhoven, 2008.

[GM08c] Murdoch J. Gabbay and Aad Mathijssen. One-and-a-halfth-order Logic. *Journal of Logic and Computation*, 18(4):521–562, August 2008.

[GM09a] Murdoch J. Gabbay and Aad Mathijssen. *Festschrift in Honour of Peter B. Andrews on his 70th Birthday*, chapter The lambda-calculus is nominal algebraic. Studies in Logic and the Foundations of Mathematics. IFCoLog, 2009. To appear.

[GM09b] Murdoch J. Gabbay and Aad Mathijssen. A nominal axiomatisation of the lambda-calculus. *Journal of Logic and Computation*, 2009. In press.

[GMR+08] Jan Friso Groote, Aad Mathijssen, Michel A. Reniers, Yaroslav S. Usenko, and Muck van Weerdenburg. Analysis of distributed systems with mCRL2. In Michael Alexander and William Gardner, editors, *Process Algebra for Parallel and Distributed Processing*, pages 99–128. Chapman and Hall, 2008.

[GP01] Murdoch J. Gabbay and A. M. Pitts. A New Approach to Abstract Syntax with Variable Binding. *Formal Aspects of Computing*, 13(3–5):341–363, 2001.

[Gro97] Jan Friso Groote. The syntax and semantics of timed μCRL. Technical Report SEN-R9709, CWI, Amsterdam, 1997.

[HMT85] L. Henkin, J. D. Monk, and A. Tarski. *Cylindric Algebras*. North Holland, 1971 and 1985. Parts I and II.

[HP07] Martin Hyland and John Power. The category theoretic understanding of universal algebra: Lawvere theories and monads. *Electronic Notes in Theoretical Computer Science*, 172:437–458, 2007.

[HvO03] Dimitri Hendriks and Vincent van Oostrom. Adbmal. In *CADE 2003: Conference on Automated Deduction*, pages 136–150, 2003.

[Joh87] P. T. Johnstone. *Notes on logic and set theory*. Cambridge University Press, 1987.

[Joj04] Gueorgui I. Jojgov. *Incomplete Proofs and Terms and Their Use in Interactive Theorem Proving*. PhD thesis, Technische Universiteit Eindhoven, 2004.

[KD02] Joost-Pieter Katoen and Pedro R. D'Argenio. General distributions in process algebra. In *Lectures on formal methods and performance analysis: first EEF/Euro summer school on trends in computer science*, pages 375–429. Springer, 2002.

[KKSdV97] J.R. Kennaway, J.W. Klop, M.R. Sleep, and F.J. de Vries. Infinitary lambda calculus. *Theoretical Computer Science*, 175:93–125, 1997.

[Law63] F. W. Lawvere. *Functorial Semantics of Algebraic Theories*. PhD thesis, Columbia University, 1963. Available with commentary as TAC Reprint 5.

[Les94] Pierre Lescanne. From lambda-sigma to lambda-upsilon: a journey through calculi of explicit substitutions. In *Proc. 21st ACM SIGPLAN-SIGACT Symposium on Principles of Programming Languages (POPL'94)*, pages 60–69. ACM Press, 1994.

[LS04] S. Lusin and A. Salibra. The lattice of lambda theories. *Journal of Logic and Computation*, 14 n.3:373–394, 2004.

[Lut02] Bas Luttik. *Choice Quantification in Process Algebra*. PhD thesis, University of Amsterdam, 2002.

[Mat07] Aad Mathijssen. *Logical Calculi for Reasoning with Binding*. PhD thesis, Technische Universiteit Eindhoven, 2007.

[Mei92] K. Meinke. Universal algebra in higher types. *Theoretical Computer Science*, 100(2):385–417, 1992.

[Mic01] Marino Miculan. Developing (meta)theory of lambda-calculus in the theory of contexts. *Electronic Notes in Theoretical Computer Science*, 1(58), 2001.

[OP99] Peter W. O'Hearn and David J. Pym. The logic of bunched implications. *Bulletin of Symbolic Logic*, 2(5):215–244, 1999.

[Par01] Joachim Parrow. An introduction to the pi-calculus. In Jan Bergstra, Alban Ponse, and Scott Smolka, editors, *Handbook of Process Algebra*, pages 479–543. Elsevier Science, 2001.

[Pit03] A. M. Pitts. Nominal logic, a first order theory of names and binding. *Information and Computation*, 186(2):165–193, 2003.

[Pra65] Dag Prawitz. *Natural Deduction: A Proof Theoretical Study*. Almqvist and Wiksell, Stockholm, 1965.

[Res99] Greg Restall. *An Introduction to Substructural Logics*. Routledge, 1999.

[Rey02] John C. Reynolds. Separation logic: A logic for shared mutable data structures. In *LICS '02: Proceedings of the 17th Annual IEEE Symposium on Logic in Computer Science*, pages 55–74. IEEE Computer Society, 2002.

[Sal00] Antonino Salibra. On the algebraic models of lambda calculus. *Theoretical Computer Science*, 249(1):197–240, 2000.

[Sun99] Yong Sun. An algebraic generalization of Frege structures - binding algebras. *Theoretical Computer Science*, 211:189–232, 1999.

[UPG04] Christian Urban, Andrew M. Pitts, and Murdoch J. Gabbay. Nominal Unification. *Theoretical Computer Science*, 323(1–3):473–497, 2004.

[Urb08] Christian Urban. Nominal reasoning techniques in Isabelle/HOL. *Journal of Automatic Reasoning*, 40(4):327–356, 2008.

[vB01] Johan van Benthem. Higher-order logic. In *Handbook of Philosophical Logic, 2nd Edition*, volume 1, pages 189–244. Kluwer, 2001.

Acknowledgements. Thanks to the anonymous referees. Supported by grant RYC-2006-002131 at the Polytechnic University of Madrid.

A Equivariance

We introduce atoms in this paper, when we write 'Fix a countably infinite collection of **atoms** $a, b, c, \ldots$' in Definition 2.1.

Atoms have a very special property: we can tell them apart — $a \neq b$ is true and $a = b$ is false — but they have no internal structure (set-theorists should think of *urelemente* with good reason; the concrete model we used to develop our treatment of names is based on Fraenkel-Mostowski set theory [Gab00, Bru96]).

This gives atoms a useful meta-mathematical property of *equivariance*; this property refers to the assertions written in English in this paper about nominal algebra, equational logic and so on.

Definition A.1. The language of ZFA set theory is first-order logic with equality with in addition:

- A binary predicate symbol $\in$ called *set membership*.
- A constant term-former $\mathbb{A}$ called *the set of atoms*.

We use standard sugar of classical logic.

Definition A.2. ZFA set theory has the axioms in Figure 4.

In this figure, ϕ ranges over all predicates, $\phi[y/x]$ denotes the predicate obtained by capture-avoiding substitution of x by y, and $F(y)$ represents any function which can be expressed in the language of ZFA sets. We also use the following sugar:

$x = \{z \mid z \in x\}$	is sugar for	$\forall y.(\forall z.(z \in x \Leftrightarrow z \in y) \supset x = y)$
$y = \{z \in x \mid \phi\}$	is sugar for	$\forall z.(z \in y \Leftrightarrow (z \in x \wedge \phi))$
$z = \{F(y) \mid y \in x\}$	is sugar for	$\forall u.(u \in z \Leftrightarrow \exists y.(F(y) = u \wedge y \in x))$
$z = \{x, y\}$	is sugar for	$\forall u.(u \in z \Leftrightarrow (u = x \vee u = y))$
$z = \{y \mid \exists y'.(y \in y' \wedge y' \in x)\}$	is sugar for	$\forall y.(y \in z \Leftrightarrow \exists y'.(y \in y' \wedge y' \in x))$
$z = \{y \mid y \subseteq x\}$	is sugar for	$\forall y.(y \in z \Leftrightarrow \forall y'.(y' \in y \supset y' \in x))$
$\emptyset \in x$	is sugar for	$\exists z.(z \in x \wedge \forall z'.z' \notin z)$
$y \cup \{z\} \in x$	is sugar for	$\exists u.(u \in x \wedge \forall u'.(u' \in u \Leftrightarrow u \in y \vee u = z))$

The syntactic sugar used in set theory is very rich; further details can be found elsewhere [Joh87]. The only property we care about is that set theory is a foundational theory and is rich enough to express, in principle at least, all the mathematics in the rest of this paper.

(**Sets**)	$\forall x.((\exists y.y \in x) \supset x \notin \mathbb{A})$	
(**Extensionality**)	$\forall x.(x \notin \mathbb{A} \supset x = \{z \mid z \in x\})$	
(**Comprehension**)	$\forall x.\exists y.(y \notin \mathbb{A} \wedge y = \{z \in x \mid \phi\})$	(y not free in ϕ)
($\in$-**Induction**)	$\forall x.(\forall y.(y \in x \supset \phi[y/x]) \supset \phi) \supset \forall x.\phi$	
(**Replacement**)	$\forall x.\exists z.(z \notin \mathbb{A} \wedge z = \{F(y) \mid y \in x\})$	
(**Pairset**)	$\forall x.\forall y.\exists z.(z = \{x, y\})$	
(**Union**)	$\forall x.\exists z.(z \notin \mathbb{A} \wedge z = \{y \mid \exists y'.(y \in y' \wedge y' \in x)\})$	
(**Powerset**)	$\forall x.\exists z.(z = \{y \mid y \subseteq x\})$	
(**Infinity**)	$\exists x.(\emptyset \in x \wedge \forall y.(y \in x \supset y \cup \{y\} \in x))$	

Figure 4: Axioms of ZFA set theory

Definition A.3. We define a **permutation action** on ZFA sets by:

$$\pi \cdot a = \pi(a) \qquad \pi \cdot X = \{\pi \cdot x \mid x \in X\} \quad (X \notin \mathbb{A})$$

This definition is by ϵ-induction, a standard method in set theory [Joh87] which relies on a well-foundedness property implied by ($\in$-**Induction**).

Recall that ϕ ranges over predicates of ZFA. Write $\phi(x_1, \ldots, x_n)$ to range over predicates which mention at most $x_1, \ldots, x_n$ as free variable symbols.

Theorem A.4 (ZFA equivariance)**.** *If $\phi(x_1, \ldots, x_n)$ is a predicate of ZFA set theory then*

$$\phi(x_1, \ldots, x_n) \Leftrightarrow \phi(\pi \cdot x_1, \ldots, \pi \cdot x_n)$$

is always provable.

As a corollary, $\phi(x_1, \ldots, x_n)$ and $\phi(\pi \cdot x_1, \ldots, \pi \cdot x_n)$ are interchangeable in proof and in validity on models.

Proof. We work by induction on the syntax of ϕ.

- By definition, $x \in y$ implies $\pi \cdot x \in \pi \cdot y$ follows directly from the fact that $\pi \cdot y = \{\pi \cdot y' \mid y' \in y\}$. The reverse implication is easy using π^{-1}.
- Similarly, $x = y$ if and only if $\pi \cdot x = \pi \cdot y$.
- The case of $\bot$ is trivial, and the cases of $\phi_1 \supset \phi_2$ and $\forall z.\phi'$ follow using the inductive hypothesis.

- $\pi \cdot \mathbb{A} = \mathbb{A}$ is provable, so $x \in \mathbb{A}$ if and only if $\pi \cdot x \in \mathbb{A}$, and $\mathbb{A} \in y$ if and only if $\mathbb{A} \in \pi \cdot y$, and similarly $x = \mathbb{A}$ if and only if $\pi \cdot x = \mathbb{A}$ and $\mathbb{A} = y$ if and only if $\mathbb{A} = \pi \cdot y$.

The result follows. □

Equivariance was first observed by Fraenkel and Mostowski and used to prove the independence of the axiom of choice from the other axioms of set theory [Bru96]. Atoms are atomic objects with no internal structure, so it is natural to use these to model variable symbols. This idea appears already in [Bar75]. To our knowledge the first author's PhD thesis [Gab00] observed the application of equivariance to practical reasoning on variables in computer science as we use it in this paper.

We have used equivariance in structural inductive proofs in this paper to rename atoms in inductive hypotheses while remaining fully formal. We have found this to be a very useful technique. To our knowledge the proof of Lemma 8.3 in [Gab07a] is the first use of equivariance to rename variable symbols in a discoursive inductive proof on abstract syntax; it is used also, for example, in the proofs of Theorems 5.1, 5.2, 5.3, and Lemma 5.7 in [GM08c]. If we wish to be fully formal but ignore equivariance then we must work by induction on measures such as term length or derivation depth. Such proofs tend to be longer and they are rarely given in full detail outside of a theorem-prover; we do not see the point of this effort, given that equivariance is available. Integrating the equivariance reasoning principle into a theorem-prover is a problem which remains to be completely solved. The first author comments on this for example in Subsection 18.3, Remark 19.3.1, and Section 20 of his thesis [Gab00]. Since then much progress has been made [Urb08]. In any case, outside a theorem-prover at the informal meta-level which is the discourse of this paper, equivariance serves us well.