Finite and infinite support in nominal algebra and logic: nominal completeness theorems for free

Murdoch J. Gabbay

http://www.gabbay.org.uk

Abstract

By operations on models we show how to relate completeness with respect to permissive-nominal models to completeness with respect to nominal models with finite support. Models with finite support are a special case of permissive-nominal models, so the construction hinges on generating from an instance of the latter, some instance of the former in which sufficiently many inequalities are preserved between elements. We do this using an infinite generalisation of nominal atoms-abstraction.

The results are of interest in their own right, but also, we factor the mathematics so as to maximise the chances that it could be used off-the-shelf for other nominal reasoning systems too. Models with infinite support can be easier to work with, so it is useful to have a semi-automatic theorem to transfer results from classes of infinitely-supported nominal models to the more restricted class of models with finite support.

In conclusion, we consider different permissive-nominal syntaxes and nominal models and discuss how they relate to the results proved here.

Key words: Permissive-nominal techniques, infinite support, finite support, nominal algebra, permissive-nominal logic, completeness, infinite atoms-abstraction.

February 11, 2012

Contents

1 Introduction **2**

2 Permissive-nominal sets **4**
2.1 Atoms, permutations, permission sets 4
2.2 Permissive-nominal sets 5
2.3 Examples 6

3 Permissive-nominal terms syntax and its interpretation **6**
3.1 Signatures 6
3.2 Terms 7
3.3 α-equivalence 8
3.4 Interpretation of signatures and terms 8

4 Reducing support of an interpretation **9**
4.1 Abstraction by atoms and by infinite lists of distinct atoms: $[a]x$ and $[l]x$. 9
4.2 Restricting permutations π/S 11
4.3 Making support smaller 12

5 Three commutation results **13**
5.1 Atoms of a term 13
5.2 First commutation result 13
5.3 Second commutation result 14
5.4 Third commutation result 15

6 Nominal algebra completeness relative to interpretations with finite support **15**

7 Permissive-nominal logic **17**
7.1 Sketch of permissive-nominal logic 17
7.2 Three notions of validity in denotations 17
7.3 Finite support denotations from medium support denotations 18
7.4 Relevance of the theorem 20

8 More permission sets, more permutations **21**
8.1 More permission sets 21
8.2 *shift*-permutations 22

9 Conclusions **24**

1. Introduction

Nominal techniques are an approach to variables in syntax and semantics which give variables denotational reality as *names*. The semantics underlying nominal techniques are *nominal sets* [GP01], which identify variable symbols with names or (for set theorists) *urelemente*. We may call names/urelemente *atoms* and we write the set of all atoms as $\mathbb{A}$.

According to nominal techniques, syntax and semantics both 'contain' atoms, in a sense made formal by a notion of *support* (see Definition 2.7).

The original applications of nominal sets and nominal terms [GP01, UPG04] admitted only finite support (the interested reader can find more applications listed on [Mul10]).

Permissive-nominal terms and models generalise this by allowing infinite support (based on a set of finitely representable but still infinite supporting sets called *permission sets*). Precise definitions will come later. For the benefit of the reader already familiar with nominal techniques we give a simple schematic for how this fits together:

$$\begin{array}{ccc} \text{nominal sets} & \leftrightarrow & \text{nominal terms} \\ \subseteq & & \subseteq \\ \text{permissive-nominal sets} & \leftrightarrow & \text{permissive-nominal terms} \end{array}$$

Both models and syntax seem better-behaved in the permissive case: we avoid the conditional reasoning typical of more traditional finitely-supported nominal techniques.[1] This makes it possible to unify the semantic and syntactic notions of α-equivalence and freshness, to 'just quotient' terms by α-equivalence, and to cleanly add universal quantification. Some complex mathematical proofs become dramatically simpler. Precise examples are cited in the Conclusions of this paper.

So permissive-nominal techniques are arguably nicer to work with, but 'ordinary' nominal techniques are arguably more elementary (no infinities to confuse the reader)—and they are sufficient for many applications.

We indicate subset inclusions in the schematic above because models with finite support are special cases of models with infinite support, and it has been shown by arguments on syntax how to map from 'ordinary' nominal syntax to permissive-nominal syntax [DGM10, Section 4].

But what about the other way around?

In this paper, we explore models with differently-sized sets of atoms, give constructions to move from 'larger' to 'smaller' support, and test when these size transformations can and cannot be internally detected by the logics concerned. The main two results are Theorems 6.8 and 7.15—these follow from two technical results, Theorem 5.2 and Lemma 5.4.

Because our arguments are based on models, it is fairly easy to apply them to different syntaxes. In this paper we use the two examples studied in previous work by the author and others: nominal algebra [GM09] (an equality reasoning system whose term language is nominal terms) and permissive-nominal logic (ditto, for first-order logic) [DG12]. See also a recent survey paper, which covers both of these in a uniform presentation [Gab12b].

Structure of the paper

- In Section 2 we briefly introduce permissive-nominal sets, with examples. These will be our semantic universe in this paper; nominal sets from [GP01] are a special case.
- In Section 3 we introduce permissive-nominal terms: signatures, terms, α-equivalence, and their interpretation in permissive-nominal sets. The critical definition is Definition 3.13, which maps from syntax to semantics.

[1] For instance, 'nominal algebra' uses equations conditional on freshness constraints saying that 'a is fresh for X' [GM09], whereas 'permissive-nominal algebra' uses just equations [Gab12b].

- Section 4 shows how to reduce the size of the support of a interpretation with 'large' support, to obtain a interpretation with 'smaller' support. This requires some interesting technical constructions. Notably, we consider atoms-abstraction by a list of atoms $[l]x$ (Definition 4.3), and a permutative notion of restricting a permutation π/S (Definition 4.11).
- In Section 5 are three technical commutation results: the common theme is that reducing the size of the support of a interpretation commutes with the structure of that interpretation.
- Section 6 proves our first main theorem, that permissive-nominal algebra is complete over finitely-supported interpretations (Theorem 6.8).
- Section 7 introduces a novel notion of 'medium support' (Definition 7.4) and proves our second main theorem, that permissive-nominal logic over interpretations with medium support has the same validity as over interpretations with finite support (Theorem 7.15). We discuss what this means in Subsection 7.4.
- Section 8 discusses how the precise design of permission sets and permutations affects the proofs of this paper. We find that the results are delicate: even quite small changes can break the proofs (Propositions 8.2 and 8.5).
- We conclude with a technical discussion of our results, related work, and future work.

2. Permissive-nominal sets

We start with the basic definitions of permission sets, permissive-nominal sets, and then we give some examples.

2.1. Atoms, permutations, permission sets

Definition 2.1. Write $\mathbb{N} = \{0, 1, 2, 3, \ldots\}$ for the natural numbers. and $\mathbb{Z} = \{0, \text{-}1, 1, \text{-}2, 2, \ldots\}$ for the integers.

Definition 2.2. For each $i \in \mathbb{N}$ fix a pair of disjoint countably infinite sets of **atoms** $\mathbb{A}^<_i$ and $\mathbb{A}^>_i$. Write

$$\mathbb{A}^< = \bigcup \mathbb{A}^<_i, \quad \mathbb{A}^> = \bigcup \mathbb{A}^>_i, \quad \mathbb{A}_i = \mathbb{A}^<_i \cup \mathbb{A}^>_i, \quad \text{and} \quad \mathbb{A} = \mathbb{A}^< \cup \mathbb{A}^>.$$

$a, b, c, \ldots$ will range over *distinct* atoms: we call this the **permutative** convention.

Definition 2.3. Given $a, b \in \mathbb{A}_i$ for some $i \in \mathbb{N}$ write $(a\ b)$ for the **swapping** bijection on atoms mapping a to b, b to a, and any other $c \in \mathbb{A} \setminus \{a, b\}$ to c.

If π is a bijection on atoms define $\mathit{nontriv}(\pi) = \{a \mid \pi(a) \neq a\}$.

Write $\mathbb{P}_{\text{fin}}$ for the group of bijections (finitely) generated by swappings, and call these bijections **permutations**.

Write $\pi \circ \pi'$ for the **composition** of π and π' (so $(\pi \circ \pi')(a) = \pi(\pi'(a))$). Write id for the **identity** permutation (so $\mathit{id}(a) = a$ always).

Lemma 2.4. *A bijection π on atoms is a permutation if and only if*

- *$a \in \mathbb{A}_i$ if and only if $\pi(a) \in \mathbb{A}_i$.*
- *$\mathit{nontriv}(\pi) = \{a \mid \pi(a) \neq a\}$ is finite.*

Definition 2.5. If $A \subseteq \mathbb{A}$ define the **pointwise** action by $\pi{\cdot}A = \{\pi(a) \mid a \in A\}$.

A **permission set** S is a set of the form $\pi{\cdot}\mathbb{A}^{\leq}$. S, T will range over permission sets.

The choices made in Definitions 2.3 and 2.5 make Theorems 6.8 and 7.15 work. These choices are possible within the framework of [Gab12b].

2.2. Permissive-nominal sets

Definition 2.6. A **set with a permutation action** X is a pair $(|\mathsf{X}|, \cdot)$ of a **carrier set** $|\mathsf{X}|$ and a group action on the carrier set $(\mathbb{P}_{\mathrm{fin}} \times |\mathsf{X}|) \to |\mathsf{X}|$, written infix as $\pi{\cdot}x$.[2]

Say $A \subseteq \mathbb{A}$ **supports** $x \in |\mathsf{X}|$ when for every (finite) permutation $\pi \in \mathbb{P}_{\mathrm{fin}}$, if $\pi(a) = a$ for all $a \in A$ then $\pi{\cdot}x = x$.

Definition 2.7. A **permissive-nominal set** is a set with a permutation action such that every element has a unique least supporting set $\mathit{supp}(x)$ such that $\mathit{supp}(x) \subseteq S$ for some permission set S. We call this the **support** of x.
X, Y will range over permissive-nominal sets.

In fact, if $x \in |\mathsf{X}|$ has *some* supporting set $A \subseteq S$, then it has a *least* one; see e.g. [DG10, Theorem 4.3].

Definition 2.8. If π is a permutation and $A \subseteq \mathbb{A}$ write $\pi|_A$ for the **restriction** of π to A. This is the partial function such that $\pi|_A(a) = \pi(a)$ when $a \in A$, and is undefined otherwise.

Lemma 2.9. *Suppose* X *is a nominal set. Suppose* $x \in |\mathsf{X}|$ *and* $A \subseteq \mathbb{A}$ *supports* x.
Then $\pi|_A = \pi'|_A$ *implies* $\pi{\cdot}x = \pi'{\cdot}x$.

Proof. From the definition of support, considering $\pi^{-1}{\circ}\pi'$. □

Lemma 2.10. *Suppose* X *is a permissive-nominal set and* $x \in |\mathsf{X}|$. *Then* $\mathit{supp}(\pi{\cdot}x) = \pi{\cdot}\mathit{supp}(x)$.

Proof. By a routine calculation using the group action. □

We conclude with a useful condition for checking whether $a \in \mathit{supp}(x)$:

Corollary 2.11. *Suppose* X *is a permissive-nominal set and* $x \in |\mathsf{X}|$. *Suppose* $b \notin \mathit{supp}(x)$. *Then* $(b\ a){\cdot}x = x$ *if and only if* $a \notin \mathit{supp}(x)$.

Proof. Suppose $b \notin \mathit{supp}(x)$. The right-to-left implication is by the definition of support. For the left-to-right implication, we prove the contrapositive. Suppose $a \in \mathit{supp}(x)$. By Lemma 2.10 $\mathit{supp}((b\ a){\cdot}x) = (b\ a){\cdot}\mathit{supp}(x)$. By our suppositions, $(b\ a){\cdot}\mathit{supp}(x) \neq \mathit{supp}(x)$. It follows that $(b\ a){\cdot}x \neq x$. □

[2] So, $\mathit{id}{\cdot}x = x$ and $\pi{\cdot}(\pi'{\cdot}x) = (\pi{\circ}\pi'){\cdot}x$ for every π and π' and every $x \in |\mathsf{X}|$.

2.3. Examples

We briefly consider examples of permissive-nominal sets, which will be useful shortly.

Definition 2.12. $\mathbb{A}$ the set of atoms can be considered a permissive-nominal set with a natural permutation action $\pi{\cdot}a = \pi(a)$.

In the case of $\mathbb{A}$ only, we will be lax about the distinction between the set, and the permissive-nominal set with its natural permutation action.

Definition 2.13. Suppose X is a permissive-nominal set and $\mathbb{A}_\nu$ is a set of atoms. Suppose $x \in |\mathsf{X}|$ and $a \in \mathbb{A}_\nu$. Define **atoms-abstraction** $[a]x$ and $[\mathbb{A}_\nu]\mathsf{X}$ by:

$$\begin{array}{rl} [a]x & = \{(a,x)\} \cup \{(b,(b\ a){\cdot}x) \mid b \in \mathbb{A}_\nu \backslash \mathit{supp}(x)\} \\ |[\mathbb{A}_\nu]\mathsf{X}| & = \{[a]x \mid a \in \mathbb{A}_\nu,\ x \in |\mathsf{X}|\} \\ \pi{\cdot}[a]x & = [\pi(a)]\pi{\cdot}x \end{array}$$

(Compare Definition 2.13 with Definition 4.3.)

Remark 2.14. In the definition of $[a]x$ in Definition 2.13 recall that by our permutative convention $b \neq a$. An equivalent and more compact way of writing this is

$$[a]x = \{(\pi(a), \pi{\cdot}x) \mid \pi \in \mathit{fix}(\mathit{supp}(x)\backslash\{a\})\}$$

where $\mathit{fix}(A) = \{\pi \mid \forall a{\in}A.\pi(a) = a\}$ [Gab11a, Definition 3.8].

Lemma 2.15.
1. $[\mathbb{A}_\nu]\mathsf{X}$ *is a permissive-nominal set.*
2. $[a]x{=}[a]x'$ *if and only if* $x{=}x'$*, for* $a{\in}\mathbb{A}_\nu$ *and* $x{\in}|\mathsf{X}|$.
3. $[a]x{=}[a']x'$ *if and only if* $a'{\notin}\mathit{supp}(x)$ *and* $(a'\ a){\cdot}x{=}x'$*, for* $a, a'{\in}\mathbb{A}_\nu$ *and* $x, x'{\in}|\mathsf{X}|$.

Definition 2.16. If X_i are permissive-nominal sets for $1 \leq i \leq n$ then define $\mathsf{X}_1 \times \ldots \times \mathsf{X}_n$ by:

$$\begin{array}{rl} |\mathsf{X}_1 \times \ldots \times \mathsf{X}_n| & = |\mathsf{X}_1| \times \ldots \times |\mathsf{X}_n| \\ \pi{\cdot}(x_1, \ldots, x_n) & = (\pi{\cdot}x_1, \ldots, \pi{\cdot}x_n) \end{array}$$

Lemma 2.17.
- $\mathit{supp}(a) = \{a\}$.
- $\mathit{supp}([a]x) = \mathit{supp}(x) \setminus \{a\}$.
- $\mathit{supp}((x_1, \ldots, x_n)) = \bigcup\{\mathit{supp}(x_i) \mid 1 \leq i \leq n\}$.

Proof. Proofs are as in [GP01] or [Gab11a]. □

3. Permissive-nominal terms syntax and its interpretation

3.1. Signatures

Definition 3.1. A **sort-signature** is a tuple $(\mathcal{A}, \mathcal{B})$ of **name** and **base** sorts $\mathcal{A} \subseteq \mathbb{N}$ and $\mathcal{B}$.

ν will range over name sorts; τ will range over base sorts.

A **sort language** is defined by

$$\alpha ::= \nu \mid \tau \mid (\alpha, \ldots, \alpha) \mid [\nu]\alpha.$$

Definition 3.2. A **term-signature** over a sort-signature $(\mathcal{A}, \mathcal{B})$ is a tuple $(\mathcal{C}, \mathcal{X}, \mathcal{F}, ar, pmss)$ where:

- $\mathcal{C}$ is a set of **constants**.
- $\mathcal{X}$ is a set of **unknowns**.
- $\mathcal{F}$ is a set of **term-formers**.
- ar assigns
  - to each constant $C \in \mathcal{C}$ a base sort τ which we may write $sort(C)$,
  - to each unknown $X \in \mathcal{X}$ a sort α which we write may $sort(X)$, and
  - to each $\mathsf{f} \in \mathcal{F}$ a **term-former arity** $(\alpha)\tau$, where α and τ are in the sort-language determined by $(\mathcal{A}, \mathcal{B})$.
- $pmss$ assigns to each constant a set $pmss(C) \subseteq \mathbb{A}^<$.

A **(nominal terms) signature** Σ is then a tuple $(\mathcal{A}, \mathcal{B}, \mathcal{C}, \mathcal{X}, \mathcal{F}, ar, pmss)$.

We may write $((\alpha_1, \ldots, \alpha_n))\tau$ just as $(\alpha_1, \ldots, \alpha_n)\tau$.

3.2. *Terms*

Definition 3.3. For each signature $\Sigma = (\mathcal{A}, \mathcal{B}, \mathcal{C}, \mathcal{X}, \mathcal{F}, ar, pmss)$, define **terms** over Σ by:

$$\frac{(a \in \mathbb{A}_\nu,\ \nu \in \mathcal{A})}{a : \nu} \qquad \frac{(sort(C) = \tau)}{\pi{\cdot}C : \tau} \qquad \frac{(sort(X) = \alpha)}{\pi{\cdot}X : \alpha}$$

$$\frac{r : \alpha \quad (ar(\mathsf{f}) = (\alpha)\tau)}{\mathsf{f}(r) : \tau} \qquad \frac{r_1 : \alpha_1 \ \ldots\ r_n : \alpha_n}{(r_1, \ldots, r_n) : (\alpha_1, \ldots, \alpha_n)} \qquad \frac{r : \alpha \quad (a \in \mathbb{A}_\nu,\ \nu \in \mathcal{A})}{[a]r : [\nu]\alpha}$$

We may write $\mathsf{f}((r_1, \ldots, r_n))$ as $\mathsf{f}(r_1, \ldots, r_n)$.

Definition 3.4. Define **free atoms** and the **permutation action**, and **free variables** on terms r as follows:

$$\begin{array}{rlrl} fa(a) &= \{a\} & fa(\mathsf{f}(r)) &= fa(r) \\ fa(\pi{\cdot}C) &= \pi{\cdot}pmss(C) & fa((r_1, \ldots, r_n)) &= \bigcup_{1 \le i \le n} fa(r_i) \\ fa(\pi{\cdot}X) &= \pi{\cdot}\mathbb{A}^< & fa([a]r) &= fa(r) \backslash \{a\} \end{array}$$

$$\begin{array}{rlrl} \pi{\cdot}a &= \pi(a) & \pi{\cdot}\mathsf{f}(r) &= \mathsf{f}(\pi{\cdot}r) \\ \pi{\cdot}(\pi'{\cdot}C) &= (\pi \circ \pi'){\cdot}C & \pi{\cdot}(r_1, \ldots, r_n) &= (\pi{\cdot}r_1, \ldots, \pi{\cdot}r_n) \\ \pi{\cdot}(\pi'{\cdot}X) &= (\pi \circ \pi'){\cdot}X & \pi{\cdot}[a]r &= [\pi(a)]\pi{\cdot}r \end{array}$$

$$\begin{array}{rlrl} fv(a) &= \varnothing & fv(\mathsf{f}(r)) &= fv(r) \\ fv(\pi{\cdot}C) &= \varnothing & fv((r_1, \ldots, r_n)) &= \bigcup_{1 \le i \le n} fv(r_i) \\ fv(\pi{\cdot}X) &= \{X\} & fv([a]r) &= fv(r) \end{array}$$

Remark 3.5. In Definition 3.4 we in effect give every unknown permission set $\mathbb{A}^<$ (so that $fa(\pi{\cdot}X) = \pi{\cdot}\mathbb{A}^<$). We obtain the effect of an unknown with permission set $\pi{\cdot}\mathbb{A}^<$ just by writing $\pi{\cdot}X$. This simplified design makes Proposition 5.6 easier to express. It corresponds roughly to [Gab12b, Example 3.1.7(2)].

Lemma 3.6. $fa(\pi{\cdot}r) = \pi{\cdot}fa(r)$.

Lemma 3.7. *If* $\pi(a) = \pi'(a)$ *for all* $a \in fa(r)$ *then* $\pi{\cdot}r = \pi'{\cdot}r$.

3.3. α-equivalence

Definition 3.8. A **congruence** is an equivalence relation R such that if $r\ R\ s$ then $\mathsf{f}(r)\ R\ \mathsf{f}(s)$ and $(t_1,\ldots,r,\ldots,t_n)\ R\ (t_1,\ldots,s,\ldots,t_n)$ and $[a]r\ R\ [a]s$.

α-equivalence is then the least congruence such that if $a, b \notin fa(r)$ then $(b\,a){\cdot}r =_\alpha r$.[3]

We do not quotient terms by α-equivalence. The syntax $[a]r$ is a formal pair of a and r. So for example, $[a]X$ and $[b](b\,a){\cdot}X$ for $b \notin \mathbb{A}^<$ are different concrete terms.

In fact, we never use α-equivalence $=_\alpha$ directly in this paper (it would be needed if we proved soundness and completeness, but these proofs are in other papers and are not included here). However $=_\alpha$ lurks in the background, hard-wired into the denotation: it can be proved that if $r =_\alpha s$ then r and s will always denote the same element in Definition 3.13.

3.4. Interpretation of signatures and terms

Definition 3.9. Suppose X and Y are permissive-nominal sets and $F \in |\mathsf{X}| \to |\mathsf{Y}|$ is a function. Call F **equivariant** when $F(\pi{\cdot}x) = \pi{\cdot}F(x)$ for all permutations $\pi \in \mathbb{P}_{\mathrm{fin}}$ and $x \in |\mathsf{X}|$.

Definition 3.10. Suppose $(\mathcal{A},\mathcal{B})$ is a sort-signature (Definition 3.1).

A **interpretation** $\mathscr{I}$ for $(\mathcal{A},\mathcal{B})$ consists of an assignment of a permissive-nominal set $[\![\alpha]\!]^{\mathscr{I}}$ to each sort α in $(\mathcal{A},\mathcal{B})$, along with equivariant maps

- for each $\nu \in \mathcal{A}$ an equivariant and injective map $\mathbb{A}_\nu \to [\![\nu]\!]^{\mathscr{I}}$ which we write $a^{\mathscr{I}}$,
- for each $\nu \in \mathcal{A}$ and α an equivariant and injective map $[\mathbb{A}_\nu][\![\alpha]\!]^{\mathscr{I}} \to [\![[\nu]\alpha]\!]^{\mathscr{I}}$ which we write $[a]^{\mathscr{I}}x$, and
- for each α_i for $1 \le i \le n$ an equivariant and injective map $\Pi_i[\![\alpha_i]\!]^{\mathscr{I}} \to [\![(\alpha_1,\ldots,\alpha_n)]\!]^{\mathscr{I}}$ which we write $(x_1,\ldots,x_n)^{\mathscr{I}}$.

Definition 3.11. Suppose $\Sigma = (\mathcal{A},\mathcal{B},\mathcal{C},\mathcal{F},ar,pmss)$ is a signature (Definition 3.2).

A **(Σ-)interpretation** $\mathscr{I}$ for Σ, or **Σ-algebra**, consists of the following data:

- An interpretation for the sort-signature $(\mathcal{A},\mathcal{B})$ (Definition 3.10).
- For every $\mathsf{f} \in \mathcal{F}$ with $ar(\mathsf{f}) = (\alpha)\tau$ an equivariant function $\mathsf{f}^{\mathscr{I}}$ from $[\![\alpha]\!]^{\mathscr{I}}$ to $[\![\tau]\!]^{\mathscr{I}}$.
- An assignment of a $C^{\mathscr{I}} \in [\![sort(C)]\!]^{\mathscr{I}}$ to $C \in \mathcal{C}$, such that $supp(C^{\mathscr{I}}) \subseteq pmss(C)$.

Definition 3.12. Suppose $\mathscr{I}$ is a Σ-algebra. A **valuation** ς to $\mathscr{I}$ is an equivariant function on unknowns $\mathcal{X}$ such that for each unknown X, $\varsigma(X) \in [\![sort(X)]\!]^{\mathscr{I}}$.

ς will range over valuations.

[3] This characterisation, which follows [GM07] (see Lemma 3.2 and the discussion preceding it in [GM07]) captures in slightly abstract form three more syntax-directed rules: $b \notin fa(r)$ then $[b](b\,a){\cdot}r =_\alpha [a]r$, and if $\pi|_{\mathbb{A}^<} = \pi'|_{\mathbb{A}^<}$ then $\pi{\cdot}X =_\alpha \pi'{\cdot}X$, and if $\pi|_{pmss(C)} = \pi'|_{pmss(C)}$ then $\pi{\cdot}C =_\alpha \pi'{\cdot}C$.

Definition 3.13. Suppose $\mathscr{I}$ is a Σ-algebra. Suppose ς is a valuation to $\mathscr{I}$.

Extend $\mathscr{I}$ to an **interpretation** on terms $[\![r]\!]^{\mathscr{I}}_{\varsigma}$ (where of course r is a term in the signature Σ) by:

$$\begin{array}{cc} [\![a]\!]^{\mathscr{I}}_{\varsigma} = a^{\mathscr{I}} & [\![\mathsf{f}(r)]\!]^{\mathscr{I}}_{\varsigma} = \mathsf{f}^{\mathscr{I}}([\![r]\!]^{\mathscr{I}}_{\varsigma}) \\ [\![C]\!]^{\mathscr{I}}_{\varsigma} = C^{\mathscr{I}} & [\![(r_1,\ldots,r_n)]\!]^{\mathscr{I}}_{\varsigma} = ([\![r_1]\!]^{\mathscr{I}}_{\varsigma},\ldots,[\![r_n]\!]^{\mathscr{I}}_{\varsigma})^{\mathscr{I}} \\ [\![\pi{\cdot}X]\!]^{\mathscr{I}}_{\varsigma} = \pi{\cdot}\varsigma(X) & [\![[a]r]\!]^{\mathscr{I}}_{\varsigma} = [a]^{\mathscr{I}}[\![r]\!]^{\mathscr{I}}_{\varsigma} \end{array}$$

Lemmas 3.14 to 3.17 are proved by routine inductions:

Lemma 3.14. *If $r:\alpha$ then $[\![r]\!]^{\mathscr{I}}_{\varsigma} \in [\![\alpha]\!]^{\mathscr{I}}$.*

Lemma 3.15. *If $\varsigma(X) = \varsigma'(X)$ for every $X \in fv(r)$ then $[\![r]\!]^{\mathscr{I}}_{\varsigma} = [\![r]\!]^{\mathscr{I}}_{\varsigma'}$.*

Lemma 3.16. $\pi{\cdot}[\![r]\!]^{\mathscr{I}}_{\varsigma} = [\![\pi{\cdot}r]\!]^{\mathscr{I}}_{\varsigma}$.

Lemma 3.17. $supp([\![r]\!]^{\mathscr{I}}_{\varsigma}) \subseteq fa(r)$.

Looking ahead, later on in Section 6, we use interpretations to define a notion of validity with respect to a model or a collection of models, written $\mathscr{H} \models r = s$ and $\mathsf{T} \models r = s$.

4. Reducing support of an interpretation

In this section we show how, given an interpretation $\mathscr{H}$, to build an interpretation $[m]\mathscr{H}$ with 'smaller' support.

$[m]\mathscr{H}$ will have 'almost the same structure' as $\mathscr{H}$. If two terms have a distinct denotation in $\mathscr{H}$ then their interpretation in $[m]\mathscr{H}$ is also distinct (Proposition 6.3, which is essentially Theorem 5.2 combined with Lemma 4.5).

As we shall see in Section 6, this result can be leveraged to proofs of completeness with respect to interpretations with finite support, assuming completeness with respect to all interpretations.

The idea of the construction is simple: in Definition 4.3 we take $\mathscr{H}$ and abstract all but finitely many atoms in its elements—in Definition 4.15 we show how to combine this with the interpretation of the term-formers of $\mathscr{H}$.

One way to think of this, is that we replace atoms by numerical indexes (where a is identified with its position in the infinite list of abstractions which we impose). We can think of $[m]\mathscr{H}$ as a version of $\mathscr{H}$ with *abstract de Bruijn indexes*, where we recall that de Bruijn indexes are a method of representing object-level variables as numerical indexes [dB72] typically applied concretely to formal syntax rather than to models. More on this in the Conclusions.

4.1. Abstraction by atoms and by infinite lists of distinct atoms: $[a]x$ and $[l]x$

Definition 4.1. Choose a fixed but arbitrary enumeration $a_{\text{-}1}, a_{\text{-}2}, a_{\text{-}3}, \ldots$ of some subset of $\mathbb{A}^<$—since atoms are countable, this can be done. Write this enumeration as a list, $l_* = [a_{\text{-}1}, a_{\text{-}2}, a_{\text{-}3}, \ldots]$.[4]

[4]We use negative indexes because we wrote $\mathbb{A}^<$ with a $<$. Of course this does not matter, but it does allow the diagram in Section 8 to make geometric sense.

Define a permissive-nominal set $\mathbb{L}$ (parameterised by l_*) by:

$$\begin{array}{l}\pi{\cdot}l_* = [\pi(a_{\text{-}1}), \pi(a_{\text{-}2}), \pi(a_{\text{-}3}), \ldots] \\ |\mathbb{L}| = \{\pi{\cdot}l_* \mid \text{all } \pi\}\end{array}$$

l will range over elements of $|\mathbb{L}|$.

It is very easy to check that $\mathbb{L}$ is indeed a permissive-nominal set, and that $\mathit{supp}(l)$ is equal to the atoms in l.

We will be most interested in the cases of Definition 4.1 when l_* enumerates all of $\mathbb{A}^<$ (Section 6) and when l_* enumerates 'half' of $\mathbb{A}^<$ (Section 7). However, nothing in the mathematics below will depend on this.

Definition 4.2. If $A \subseteq \mathbb{A}$ define $\mathit{fix}(A)$ by:

$$\mathit{fix}(A) = \{\pi \mid \forall a{\in}A.\pi(a) = a\}$$

Definition 4.3. Suppose X is a permissive-nominal set and $x \in |\mathsf{X}|$. Suppose $l \in |\mathbb{L}|$. Define $[l]x$ and $[\mathbb{L}]\mathsf{X}$ as follows:

$$\begin{array}{l}[l]x = \{(\pi{\cdot}l, \pi{\cdot}x) \mid \pi \in \mathit{fix}(\mathit{supp}(x)\backslash\mathit{supp}(l))\} \\ |[\mathbb{L}]\mathsf{X}| = \{[l]x \mid x \in |\mathsf{X}|,\ l \in |\mathbb{L}|\} \\ \pi{\cdot}[l]x = [\pi{\cdot}l]\pi{\cdot}x\end{array}$$

Remark 4.4. $[l]x$ and $[\mathbb{L}]\mathsf{X}$ mirror $[a]x$ and $[\mathbb{A}]\mathsf{X}$ from Definition 2.13, and have broadly similar properties. The idea of abstracting over infinitely many atoms was investigated in [Gab07] (see equation (2) in Subsection 2.1).

Lemma 4.5. *Suppose* X *is a permissive-nominal set and* $x, y \in |\mathsf{X}|$. *Suppose* $l \in |\mathbb{L}|$.
Then $[l]x = [l]y$ *if and only if* $x = y$.

Proof. Clearly if $x = y$ then $[l]x = [l]y$. Suppose $[l]x = [l]y$. By construction $(l, x) \in [l]x$, so also $(l, x) \in [l]y$. It follows that there exists π such that $\pi{\cdot}l = l$ and $\pi{\cdot}y = x$, and $\pi \in \mathit{fix}(\mathit{supp}(y) \setminus \mathit{supp}(l))$. From $\pi{\cdot}l = l$ follows that $\pi \in \mathit{fix}(\mathit{supp}(l))$. It follows that $\pi \in \mathit{fix}(\mathit{supp}(y))$ and so by Lemma 2.9 that $\pi{\cdot}y = y$. □

Lemma 4.6. *Suppose* X *is a permissive-nominal set and* $x \in |\mathsf{X}|$. *Suppose* $l \in |\mathbb{L}|$.
Then $\mathit{supp}([l]x) = \mathit{supp}(x)\backslash\mathit{supp}(l)$.

Proof. By properties of the group action if $\pi \in \mathit{fix}(\mathit{supp}(x)\backslash\mathit{supp}(l))$ then $\pi{\cdot}[l]x = [\pi{\cdot}l]\pi{\cdot}x$.

Now suppose $a \in \mathit{supp}(x)\backslash\mathit{supp}(l)$ and choose any b fresh (so $b \notin \mathit{supp}(x) \cup \mathit{supp}(l)$). It is easy to use Lemma 2.10 to verify that every $(l', x') \in [l]x$ satisfies $a \in \mathit{supp}(x')$ whereas every $(l', x') \in (b\ a){\cdot}[l]x$ satisfies $a \notin \mathit{supp}(x')$. It follows that $(b\ a){\cdot}[l]x \neq [l]x$ and so by Corollary 2.11 $a \in \mathit{supp}([l]x)$. □

Corollary 4.7. $[\mathbb{L}]\mathsf{X}$ *from Definition 4.3 is a permissive-nominal set.*

Proof. That it is a set with a permutation action is clear. That every element has a supporting permission set follows from Lemma 4.6. □

Lemma 4.8. *Suppose* X *is a permissive-nominal set. Suppose* $\hat{x} \in |[\mathbb{L}]\mathsf{X}|$ *and* $l \in |\mathbb{L}|$ *is such that* $supp(\hat{x}) \cap supp(l) = \varnothing$*. Then there exists a unique element, write it* $\hat{x}@l \in |\mathsf{X}|$*, such that* $\hat{x} = [l](\hat{x}@l)$*.*

Proof. By Lemma 4.5 $\hat{x}@l$ is unique if it exists.

Suppose $supp(l) \cap supp(\hat{x}) = \varnothing$. By construction (Definition 4.3) $\hat{x} = [l']x'$ for some $l' \in \mathbb{L}$ and $x' \in |\mathsf{X}|$. By construction (Definition 4.1) $l' = \pi{\cdot}l$ for some π.[5] It is also a fact that since $supp(l) \cap supp(\hat{x}) = \varnothing$ and (by Lemma 4.6) $supp(l') \cap supp(\hat{x}) = \varnothing$, we can suppose without loss of generality that $nontriv(\pi) \cap supp(\hat{x}) = \varnothing$. It follows that $\hat{x} = [l]\pi^{-1}{\cdot}x'$ and so $\hat{x}@l$ exists and is equal to $\pi^{-1}{\cdot}x'$. □

Lemma 4.9. *Suppose* $y_1, \ldots, y_n \in |[\mathbb{L}]\mathsf{X}|$*. Then for any* l *such that* $supp(l) \cap \bigcup supp(y_i) = \varnothing$*, there exist* $x_1, \ldots, x_n \in |\mathsf{X}|$ *such that* $y_i = [l]x_i$ *for* $1 \leq i \leq n$*.*

Proof. We use Lemma 4.8 and take $x_i = y_i@l$. □

4.2. Restricting permutations π/S

Intuitively, π/S (Definition 4.11) is the 'smallest' permutation to agree with π on S. π/S is 'trying' to be $\pi|_S$ (Definition 2.8) but π/S is a total function and furthermore is a permutation. The main result is Theorem 4.14, and we use π/S in Theorem 5.2.

As nominal techniques demonstrate, permutations are an attractive way to handle name-binding. Think of π/S as a version of $\pi|_S$ that we can use if we want to stay in the world of permutations.

Example 4.10. Suppose $\pi = (a\ b\ c\ d\ e)(f\ g)$ (so π maps a to b to c to d to e to a, and f to g to f). Then:

$$\pi/\{a\} = (a\ b\ e) \qquad \pi/\{a,b\} = (a\ b\ c\ e)$$
$$\pi/\{a,c\} = (a\ b\ c\ d\ e) \qquad \pi/\{a,f\} = (a\ b\ e)(f\ g)$$

Suppose $\pi = (a\ b\ c\ d\ e\ f)$. Then

$$\pi/\{b,e\} = (a\ b\ c)(d\ e\ f) \qquad \pi/\{b\} = (a\ b\ c)$$
$$\pi/\{b,e,d\} = (a\ b\ c\ d\ e\ f) \qquad \pi/\{a,d\} = (a\ b\ f)(c\ d\ e)$$

Recall the definitions of $nontriv(\pi)$ and π from Definition 2.3.

Definition 4.11. Represent permutations π as cycles; so we write π as a finite set of finite cycles indexed by $i \in I$ where cycle number i has length $\alpha_i > 1$:

$$\pi = \Pi_{i \in I}(a_{i1}\ a_{i2}\ \ldots\ a_{i\alpha_i})$$

Define π/S as that permutation obtained as follows:

- Delete from the cycle representation of π above any atom a such that $\{a, \pi(a), \pi^{-1}(a)\} \cap S = \varnothing$. That is, if there is any part of a cycle of the form '$a_1\ a_2\ a_3$' where $a_1 \notin S$, $a_2 \notin S$, and $a_3 \notin S$, then we replace it with '$a_1\ a_3$'. Repeat, until we cannot proceed.
- If there is any part of a cycle of the form '$a_1\ a_2\ a_3\ a_4$' where $a_1 \in S$ and $a_4 \in S$ but $a_2 \notin S$ and $a_3 \notin S$, break the cycle into two subcycles as follows: '$a_1\ a_2)(a_3\ a_4$'.

[5] This is the crux of the proof: $\mathbb{L}$ is composed of a single orbit under the permutation action.

In words:

> π/S is obtained from π by eliding sequences of three or more consecutive atoms not in S, and then by splitting cycles at any two consecutive atoms not in S.

Lemma 4.12. *π/S is well-defined.*

Proof. At each step the size of $nontriv$ reduces, so the rewrite system is terminating. It is not hard to check that rewrites are locally confluent. The result follows by Newman's Lemma [New42]. □

Definition 4.13. Define $\pi' \leq_S \pi$ when:

- $\pi'|_S = \pi|_S$
- $(\pi')^{-1}|_S = \pi^{-1}|_S$
- For every cycle with atoms C' in π', there is a cycle with atoms C in π such that $C' \subseteq C$.

It is easy to verify that $\leq_S$ is a transitive reflexive relation. $\leq_S$ is not antisymmetric: if $\pi = (a\ b\ c)$ and $\pi' = (a\ c\ b)$ and $S = \varnothing$ then $\pi \leq_S \pi'$ and $\pi' \leq_S \pi$ yet $\pi \neq \pi'$.

Theorem 4.14. 1. *π/S is the unique $\leq_S$-least permutation beneath π.*
2. *As a corollary, $(\pi/S)|_S = \pi|_S$ and if $\pi|_S = \pi'|_S$ and $\pi^{-1}|_S = (\pi')^{-1}|_S$ then $\pi/S = \pi'/S$.*

Proof. By construction π/S contains only those atoms, in the smallest possible cycles, necessary to agree with π and π^{-1} on S. □

4.3. Making support smaller

Given an interpretation $\mathscr{H}$ and a list of atoms m, we are interested in 'subtracting' m from the support of $\mathscr{H}$, in some sense. The main definition is Definition 4.15, which builds an interpretation with smaller support out of an interpretation. For the cases we care about, 'smaller support' will mean finite support; this will come later in Lemmas 6.4 and 7.10, which are then used in Theorems 6.8 and Theorem 7.15 respectively. Here, we give the relevant construction.

Definition 4.15. Given a signature Σ, a Σ-interpretation $\mathscr{H}$, and a list $m \in |\mathbb{L}|$ construct a Σ-interpretation $[m]\mathscr{H}$ as follows:

$$[\![\alpha]\!]^{[m]\mathscr{H}} = \{[l]x \mid l \in |\mathbb{L}|,\ x \in [\![\alpha]\!]^{\mathscr{H}}\}$$

$$\begin{aligned} a^{[m]\mathscr{H}} &= [l](a^{\mathscr{H}})\ \ (supp(l) \not\ni a) & \mathsf{f}^{[m]\mathscr{H}}([l]x) &= [l]\mathsf{f}^{\mathscr{H}}(x) \\ ([l]x_1, \ldots, [l]x_n)^{[m]\mathscr{H}} &= [l](x_1, \ldots, x_n)^{\mathscr{H}} & [a]^{[m]\mathscr{H}}([l]x) &= [l]([a]^{\mathscr{H}}x) \\ C^{[m]\mathscr{H}} &= [m]C^{\mathscr{H}} \end{aligned}$$

Remark 4.16. A couple of comments on Definition 4.15:

The index m of $[m]\mathscr{H}$ is only used to interpret constants C. We have to choose *some* list of atoms to abstract—if our language did not admit non-equivariant constants, as was the case for the original Urban-Pitts-Gabbay syntax from [UPG04] or its permissive variant from e.g. [DGM10], then we could just write $[\mathbb{L}]\mathscr{H}$.

In the case of tuples, we know we can write every element in the form $[l]x_i$ for $1 \leq i \leq n$ for some x_i, by Lemma 4.9.

Proposition 4.17. *$[m]\mathscr{H}$ from Definition 4.15 is an interpretation.*

Proof. It is routine to check that every condition in Definitions 3.10 and 3.11 is satisfied. □

The next step is to build valuations to $[m]\mathscr{H}$. This is Definition 4.18 and Proposition 4.19.

Definition 4.18. Suppose ς is a valuation to $\mathscr{H}$ and $l \in |\mathbb{L}|$. Define $[l]\varsigma$ by:

$$([l]\varsigma)(X) = [l](\varsigma(X))$$

Proposition 4.19. *If ς is a valuation to $\mathscr{H}$ then $[l]\varsigma$ is a valuation to $[m]\mathscr{H}$.*

Proof. Consider an unknown X. By assumption $\varsigma(X) \in [\![sort(X)]\!]^{\mathscr{H}}$ and $supp(\varsigma(X)) \subseteq \mathbb{A}^<$. By construction in Definitions 2.5 and 4.3, $\mathbb{A}^< \setminus supp(l)$ is finite so by Lemma 4.6, $supp([l]\varsigma(X))$ is finite. The result follows. □

5. Three commutation results

Theorem 5.2, Lemma 5.4, and Proposition 5.6 are three commutation results. In Sections 6 and 7 we will use these as the technical 'engine' behind main theorems such as Theorems 6.8 and 7.15.

5.1. Atoms of a term

First, we need a technical tool $atoms(r)$. We need this to express the side-condition $atoms(r) \cap supp(l) = \varnothing$ in Theorem 5.2, and the side-condition $atoms(r) \cap nontriv(\pi) = \varnothing$ in Proposition 5.6. Without these side-condition, the results would not hold.

Definition 5.1. Define $atoms(r)$ inductively by:

$$\begin{array}{rr} atoms(a) = \{a\} & atoms(\mathsf{f}(r)) = atoms(r) \\ atoms(\pi{\cdot}C) = nontriv(\pi/pmss(C)) & atoms((r_1,\ldots,r_n)) = \bigcup atoms(r_i) \\ atoms(\pi{\cdot}X) = nontriv(\pi/\mathbb{A}^<) & atoms([a]r) = atoms(r) \cup \{a\} \end{array}$$

$atoms(r)$ collects the atoms 'explicit' in r. Contrast this with 'free atoms of' $fa(r)$ from Definition 3.4 which collects the atoms 'potentially' in r. For instance, $fa(X) = \mathbb{A}^<$ and is infinite, but $atoms(X) = \varnothing$. This is because X mentions no atoms explicitly, but intuitively it could be instantiated for any term with atoms in $\mathbb{A}^<$.

5.2. First commutation result

Recall from Definition 4.1 the construction of $\mathbb{L}$, parameterised over some l_*.

Theorem 5.2. *Suppose $l \in |\mathbb{L}|$ and $atoms(r) \cap supp(l) = \varnothing$. Then $[\![r]\!]^{[l]\mathscr{H}}_{[l]\varsigma} = [l][\![r]\!]^{\mathscr{H}}_{\varsigma}$.*

Proof. By induction on r:

- *The case a.* We reason as follows:

$$\begin{array}{rll}
[\![a]\!]^{[l].\mathscr{H}}_{[l]\varsigma} & = [l]a^{\mathscr{H}} & \text{Defs 3.13, 4.15, } a \notin \mathit{supp}(l) \\
& = [l][\![a]\!]^{\mathscr{H}}_{\varsigma} & \text{Definition 3.13}
\end{array}$$

We know $a \notin \mathit{supp}(l)$ because we assumed $\mathit{atoms}(r) \cap \mathit{supp}(l) = \varnothing$, and $\mathit{atoms}(a) = \{a\}$.

- *The case $\pi \cdot X$.* We reason as follows:

$$\begin{array}{rll}
[\![\pi \cdot X]\!]^{[l].\mathscr{H}}_{[l]\varsigma} & = \pi \cdot [l]\varsigma(X) & \text{Definition 3.13} \\
& = (\pi/\mathbb{A}^{<}) \cdot [l]\varsigma(X) & \text{Lems 2.9 \& 4.6, Thm 4.14} \\
& = [l](\pi/\mathbb{A}^{<}) \cdot \varsigma(X) & \text{Fact} \\
& = [l]\pi \cdot \varsigma(X) & \text{Lems 2.9 \& 4.6, Thm 4.14} \\
& = [l][\![\pi \cdot X]\!]^{\mathscr{H}}_{\varsigma} & \text{Definition 3.13}
\end{array}$$

The fact above follows since we assumed $\mathit{atoms}(\pi \cdot X) \cap \mathit{supp}(l) = \varnothing$.

- *The case $[a]r$, where $a \notin \mathit{supp}(l)$.* We reason as follows:

$$\begin{array}{rll}
[\![[a]r]\!]^{[l].\mathscr{H}}_{[l]\varsigma} & = [a]^{[l].\mathscr{H}}[\![r]\!]^{[l].\mathscr{H}}_{[l]\varsigma} & \text{Definition 3.13} \\
& = [a]^{[l].\mathscr{H}}[l][\![r]\!]^{\mathscr{H}}_{\varsigma} & \text{ind. hyp.} \\
& = [l]([a]^{\mathscr{H}}[\![r]\!]^{\mathscr{H}}_{\varsigma}) & \text{Definition 4.15} \\
& = [l][\![[a]r]\!]^{\mathscr{H}}_{\varsigma} & \text{Definition 3.13}
\end{array}$$

- *The case $\pi \cdot C$.* We reason as follows:

$$\begin{array}{rll}
[\![\pi \cdot C]\!]^{[l].\mathscr{H}}_{[l]\varsigma} & = \pi \cdot [l]C^{\mathscr{H}} & \text{Defs 3.13 \& 4.15} \\
& = (\pi/\mathit{pmss}(C)) \cdot [l]C^{\mathscr{H}} & \text{Lems 2.9 \& 4.6, Thm 4.14} \\
& = [l](\pi/\mathit{pmss}(C)) \cdot C^{\mathscr{H}} & \text{Fact} \\
& = [l]\pi \cdot C^{\mathscr{H}} & \text{Lems 2.9 \& 4.6, Thm 4.14} \\
& = [l][\![\pi \cdot C]\!]^{\mathscr{H}}_{\varsigma} & \text{Definition 3.13}
\end{array}$$

The fact above follows since we assumed $\mathit{atoms}(\pi \cdot C) \cap \mathit{supp}(l) = \varnothing$.

- *The case $(r_1, \ldots, r_n)$.* We reason as follows:

$$\begin{array}{rll}
[\![(r_1, \ldots, r_n)]\!]^{[l].\mathscr{H}}_{[l]\varsigma} & = ([\![r_1]\!]^{[l].\mathscr{H}}_{[l]\varsigma}, \ldots, [\![r_n]\!]^{[l].\mathscr{H}}_{[l]\varsigma})^{[l].\mathscr{H}} & \text{Definition 3.13} \\
& = ([l][\![r_1]\!]^{\mathscr{H}}_{\varsigma}, \ldots, [l][\![r_n]\!]^{\mathscr{H}}_{\varsigma})^{[l].\mathscr{H}} & \text{ind. hyp.} \\
& = [l]([\![r_1]\!]^{\mathscr{H}}_{\varsigma}, \ldots, [\![r_n]\!]^{\mathscr{H}}_{\varsigma})^{\mathscr{H}} & \text{Definition 4.15} \\
& = [l][\![(r_1, \ldots, r_n)]\!]^{\mathscr{H}}_{\varsigma} & \text{Definition 3.13}
\end{array}$$

- *The case $\mathsf{f}(r) \ldots$* is routine. □

5.3. Second commutation result

Definition 5.3. Given an interpretation $\mathscr{H}$, a valuation ς to $\mathscr{H}$, and some X and $x \in [\![\mathit{sort}(X)]\!]^{\mathscr{H}}$ with $\mathit{supp}(x) \subseteq \mathbb{A}^{<}$, define $\varsigma[X := x]$ by:

$$\varsigma[X := x](X) = x \qquad \varsigma[X := x](Y) = \varsigma(Y)$$

Lemma 5.4. *Suppose ς, X, and x are as in Definition 5.3. Suppose $l \in |\mathbb{L}|$. Then*

$$([l]\varsigma)[X{:=}[l]x]) = [l](\varsigma[X{:=}x]).$$

Proof. By routine calculations. □

5.4. Third commutation result

Definition 5.5. Suppose ς is a valuation. Suppose π is a permutation such that $nontriv(\pi) \subseteq \mathbb{A}^<$.

Define $\pi \circ \varsigma$ by

$$(\pi \circ \varsigma)(X) = \pi \cdot \varsigma(X).$$

Proposition 5.6. *Suppose* $nontriv(\pi) \subseteq \mathbb{A}^<$ *and* $atoms(r) \cap nontriv(\pi) = \varnothing$. *Then* $[\![r]\!]^{\mathscr{H}}_{\pi \circ \varsigma} = \pi \cdot [\![r]\!]^{\mathscr{H}}_{\varsigma}$.

Proof. By a routine induction on r similar to that in Theorem 5.2:

- *The case* a. By assumption $a \notin nontriv(\pi)$.
- *The case* $\pi' \cdot X$. By assumption $nontriv(\pi) \cap nontriv(\pi'/\mathbb{A}^<) = \varnothing$. Since $nontriv(\pi) \subseteq \mathbb{A}^<$ it is a fact that $nontriv(\pi) \cap nontriv(\pi') = \varnothing$. The result follows.
- *The case* $[a]r$, *where* $a \notin supp(l)$. By assumption $a \notin nontriv(\pi)$.
- *The case* $\pi' \cdot C$. As for $\pi' \cdot X$.
- *The cases* $(r_1, \ldots, r_n)$ *and* $\mathsf{f}(r) \ldots$ are routine. □

6. Nominal algebra completeness relative to interpretations with finite support

We now have everything we need to set up two notions of validity $\vDash$ and $\vDash_{\mathit{fin}}$ (Definition 6.7) and prove our main result, that they are equal (Theorem 6.8).

Definition 6.1. Suppose r and s are terms in Σ, which is the signature of an interpretation $\mathscr{H}$.

- Write $\mathscr{H}, \varsigma \vDash r = s$ when $[\![r]\!]^{\mathscr{H}}_{\varsigma} = [\![s]\!]^{\mathscr{H}}_{\varsigma}$.
- Write $\mathscr{H} \vDash r = s$ when $\mathscr{H}, \varsigma \vDash r = s$ for every valuation ς to $\mathscr{H}$.

Notation 6.2. For the rest of this section, we will take l_* from Definition 4.1 to enumerate all of $\mathbb{A}^<$. We write the $\mathbb{L}$ so generated by Definition 4.1 as $\mathbb{L}^<$.

Recall the construction of $[m]\mathscr{H}$ from Definition 4.15.

Proposition 6.3. *Suppose* r *and* s *are terms in* Σ, *which is the signature of an interpretation* $\mathscr{H}$. *Suppose* $m \in |\mathbb{L}^<|$. *Then:*

1. *If* $\mathscr{H} \not\vDash r = s$ *then* $[m]\mathscr{H} \not\vDash r = s$.
2. *If* $\mathscr{H} \vDash r = s$ *then* $[m]\mathscr{H} \vDash r = s$.

Proof. For the first part, suppose $\mathscr{H} \not\vDash r = s$. So there exists a valuation ς to $\mathscr{H}$ such that $[\![r]\!]^{\mathscr{H}}_{\varsigma} \neq [\![s]\!]^{\mathscr{H}}_{\varsigma}$. Choose some l such that $supp(l) \cap (atoms(r) \cup atoms(s)) = \varnothing$. We can do this, because $atoms(r)$ and $atoms(s)$ are finite. By Theorem 5.2 $[\![r]\!]^{[m]\mathscr{H}}_{[l]\varsigma} = [l][\![r]\!]^{\mathscr{H}}_{\varsigma}$ and $[\![s]\!]^{[m]\mathscr{H}}_{[l]\varsigma} = [l][\![s]\!]^{\mathscr{H}}_{\varsigma}$. By Lemma 4.5 $[l][\![r]\!]^{\mathscr{H}}_{\varsigma} \neq [l][\![s]\!]^{\mathscr{H}}_{\varsigma}$. It follows that $[\![r]\!]^{[m]\mathscr{H}}_{[l]\varsigma} \neq [\![s]\!]^{[m]\mathscr{H}}_{[l]\varsigma}$.

For the second part, suppose that $\mathscr{H} \vDash r = s$ and suppose ς' is a valuation to $[m]\mathscr{H}$. Choose some $l \in |\mathbb{L}^<|$ such that

$$supp(l) \cap \Big(atoms(r) \cup atoms(s) \cup \bigcup \{ supp(\varsigma'(X)) \mid X \in fv(r) \cup fv(s) \} \Big) = \varnothing.$$

We can do this since all the sets on the right-hand side of $\cap$ are finite.

Using Lemmas 4.9 and 3.15 there exists a valuation ς to $\mathscr{H}$ such that $[\![r]\!]^{[m]\mathscr{H}}_{\varsigma'} = [\![r]\!]^{[m]\mathscr{H}}_{[l]\varsigma}$ and $[\![s]\!]^{[m]\mathscr{H}}_{\varsigma'} = [\![s]\!]^{[m]\mathscr{H}}_{[l]\varsigma}$. We now reason using Theorem 5.2 and Lemma 4.5, as in the first part. □

The model $[m]\mathscr{H}$ is composed of ordinary—i.e. finitely-supported—nominal sets, in the sense of [GP01]:

Lemma 6.4. *Every $[l]x \in [\![\alpha]\!]^{[m]\mathscr{H}}$ has finite support.*

Proof. It suffices to observe Lemma 4.6 and note that by assumption $supp(x)$ is contained in a permission set, and by assumption in Notation 6.2 $supp(l)$ is a permission set, and by construction permission sets differ finitely from one another. □

Definition 6.5 is standard, e.g. from [GM09] (nominal) or [Gab12b] (permissive-nominal):

Definition 6.5. A **(permissive-)nominal algebra theory** $\mathsf{T} = (\Sigma, Ax)$ is a pair of a signature Σ and a set of equality axioms Ax. (So elements of Ax are pairs $r = s$.)

Suppose $\mathscr{H}$ is a Σ-interpretation (Definition 3.11). Write $\mathscr{H} \vDash \mathsf{T}$ to mean that for every valuation ς to $\mathscr{H}$ and every $(r = s) \in Ax$, $[\![r]\!]^{\mathscr{H}}_{\varsigma} = [\![s]\!]^{\mathscr{H}}_{\varsigma}$.

Definition 6.6. Suppose Σ is a signature and $\mathscr{F}$ is a Σ-interpretation. Say that $\mathscr{F}$ has **finite support** when for every sort α in Σ and every $x \in |[\![\alpha]\!]^{\mathscr{F}}|$, it is the case that $supp(x)$ is finite.

Definition 6.7. Suppose $\mathsf{T} = (\Sigma, Ax)$ is a theory. Then:

- Define $\mathsf{T} \vDash_{\mathit{fin}} r = s$ to mean that $\mathscr{F} \vDash \mathsf{T}$ implies $\mathscr{F} \vDash r = s$, for every Σ-interpretation $\mathscr{F}$ with finite support.
- Define $\mathsf{T} \vDash r = s$ to mean that $\mathscr{H} \vDash \mathsf{T}$ implies $\mathscr{H} \vDash r = s$, for every Σ-interpretation $\mathscr{H}$.

Theorem 6.8. *Suppose that T is a Σ-theory.*

Then $\mathsf{T} \vDash_{\mathit{fin}} r = s$ if and only if $\mathsf{T} \vDash r = s$.

Proof. The right-to-left implication is immediate since an interpretation with finite support is an interpretation.

For the left-to-right implication we prove the contrapositive. Suppose $\mathsf{T} \nvDash r = s$. So there is an interpretation $\mathscr{H}$ such that $\mathscr{H} \vDash \mathsf{T}$ and a valuation ς to $\mathscr{H}$ such that $[\![r]\!]^{\mathscr{H}}_{\varsigma} \neq [\![s]\!]^{\mathscr{H}}_{\varsigma}$.

Choose any $m \in |\mathbb{L}^{\prec}|$. By part 2 of Proposition 6.3 $[m]\mathscr{H} \vDash \mathsf{T}$. By part 1 of Proposition 6.3 $[m]\mathscr{H} \nvDash r = s$, and by Lemma 6.4 we are done. □

Permissive-nominal algebra is sound and complete with respect to permissive-nominal models (the proof is by a Herbrand construction; see [Gab12b, Subsection 7.5]). So the relevance of Theorem 6.8 is to give completeness also with respect to interpretations with finite support.

7. Permissive-nominal logic

Permissive-nominal logic (**PNL**) extends signatures with **proposition-formers** P with arity α. It is 'first-order logic over (permissive-)nominal terms'.

Full details can be found in [DG10, DG12] or [Gab12b, Section 9]. Here, we only give the necessary outline.

7.1. Sketch of permissive-nominal logic

Definition 7.1. PNL **propositions** are defined by

$$\phi, \psi ::= \bot \mid \phi \Rightarrow \phi \mid \forall X.\phi \mid \mathsf{P}(r)$$

where we insist that $r : \alpha$ (where α is the arity of P).

Definition 7.2. if X is a nominal set and $U \subseteq |\mathsf{X}|$ call U **equivariant** when $x \in U \Leftrightarrow \pi{\cdot}x \in U$ for all $x \in |\mathsf{X}|$ and all permutations π.[6]

Definition 7.3 corresponds to e.g. [DG10, Definition 5.11].

Definition 7.3. An interpretation $\mathscr{H}$ maps a term to an element of a permissive-nominal set as in Definition 3.13, and maps each P to an equivariant subset $\mathsf{P}^{\mathscr{H}} \subseteq [\![\alpha]\!]^{\mathscr{H}}$.

This extends to propositions ϕ just as in first-order logic where $[\![\phi]\!]^{\mathscr{H}}_{\varsigma}$ is a truth-value $\top$ or $\bot$, as follows:

- $[\![\bot]\!]^{\mathscr{H}}_{\varsigma}$ (the syntax) is equal to $\bot$ (the truth-value).
- The PNL of [DG10, DG12, Gab12b] is classical, so $[\![\phi \Rightarrow \psi]\!]^{\mathscr{H}}_{\varsigma}$ is interpreted as 'not $[\![\phi]\!]^{\mathscr{H}}_{\varsigma}$ or $[\![\psi]\!]^{\mathscr{H}}_{\varsigma}$'.
- $[\![\mathsf{P}(r)]\!]^{\mathscr{H}}_{\varsigma}$ is equal to '$[\![r]\!]^{\mathscr{H}}_{\varsigma}$ is an element of $\mathsf{P}^{\mathscr{H}}$'.
- The only non-obvious case is the universal quantifier, which gets a denotation as follows:

  $$[\![\forall X.\phi]\!]^{\mathscr{H}}_{\varsigma} = \bigwedge \{ [\![\phi]\!]^{\mathscr{H}}_{\varsigma[X\,:=\,x]} \mid x \in [\![sort(X)]\!]^{\mathscr{H}},\ supp(x) \subseteq \mathbb{A}^{<} \}$$

  This is non-obvious because the $\forall X$ in $\forall X.\phi$ quantifies only over x with support in $\mathbb{A}^{<}$. More discussion on this in the Conclusions.

7.2. Three notions of validity in denotations

Three distinct notions of validity will interest us. They are parameterised by 'how many atoms' they allow in support. This is Definition 7.6; to express it, we need Definition 7.4.

Definition 7.4. For each $i \in \mathbb{N}$ fix some set $\mathbb{A}^{\ll}_i \subseteq \mathbb{A}^{<}_i$ such that $\mathbb{A}^{\ll}_i$ and $\mathbb{A}^{<}_i \setminus \mathbb{A}^{\ll}_i$ are both infinite. Write $\mathbb{A}^{\ll} = \bigcup_i \mathbb{A}^{\ll}_i$ and:

- Say that $x \in [\![\alpha]\!]^{\mathscr{H}}$ has **medium** support when $supp(x) \subseteq \pi{\cdot}\mathbb{A}^{\ll}$ for some π.
- Say that $\mathscr{H}$ has **medium** support when for every sort α and every $x \in [\![\alpha]\!]^{\mathscr{H}}$, x has medium support.

[6] This notion of equivariance coincides with that of Definition 3.9, if we consider U as a function to truth-values $\{\top, \bot\}$, such that $\pi{\cdot}\top = \top$ and $\pi{\cdot}\bot = \bot$ for all π.

Remark 7.5. The point of Definition 7.4 is that x with medium support may have infinite support, but this support cannot exhaust the atoms in $\mathbb{A}^<$. But did we not see this already in Definition 2.2 when we split $\mathbb{A}$ into $\mathbb{A}^<$ and $\mathbb{A}^>$? Yes, but PNL has a $\forall X$, so that now (and unlike was the case in permissive-nominal algebra) we have to worry about exhausting all the atoms in $\mathbb{A}^<$ within nested quantifiers. To see this idea made concrete, consider the proof of Proposition 7.7.

Definition 7.6.
- Write $\models \phi$ to mean that $\mathscr{H}, \varsigma \models \phi$ for every interpretation $\mathscr{H}$ and valuation ς to $\mathscr{H}$.
- Write $\models_{\ll} \phi$ to mean that $\mathscr{H}, \varsigma \models_{\ll} \phi$ for every interpretation $\mathscr{H}$ with medium support and valuation ς to $\mathscr{H}$.
- Write $\models_{\mathit{fin}} \phi$ to mean that $\mathscr{F}, \varsigma \models \phi$ for every interpretation $\mathscr{F}$ with finite support and valuation ς to $\mathscr{F}$.

Proposition 7.7. *$\models \phi$ implies $\models_{\ll} \phi$. The reverse implication does not necessarily hold.*

Proof. The first part is immediate since an interpretation with medium support is also an interpretation.

For the second part it suffices to provide a counterexample. Suppose a base sort τ and name sort ν and variables $X : \tau$ and $Y : \nu$. Suppose a predicate $\# : (\nu, \tau)$ with intended meaning 'is fresh for'/'is not in the support of'. Consider the formula $\phi = \forall X.\exists Y.Y\#X$. Then $\models_{\mathit{fin}} \phi$ and $\models_{\ll} \phi$, but not $\models \phi$; it might be that $\varsigma(X) = l$ where l lists all atoms in $\mathbb{A}^<$, so there exists no atom in $\mathbb{A}^<$ (by Definition 7.3, Y ranges over atoms in $\mathbb{A}^<$) that is not in $\mathit{supp}(l)$. □

The rest of this section is devoted to proving that $\models_{\ll} \phi$ if and only if $\models_{\mathit{fin}} \phi$ (Theorem 7.15). We discuss the relevance of these results in Subsection 7.4.

7.3. Finite support denotations from medium support denotations

Notation 7.8. For the rest of this section, we will take l_* from Definition 4.1 to enumerate $\mathbb{A}^<$. We write the $\mathbb{L}$ generated by Definition 4.1 as $\mathbb{L}^<$.

Definition 7.9. Given a PNL interpretation $\mathscr{H}$ with medium support and a list $m \in |\mathbb{L}^<|$, generate a PNL interpretation $[m]\mathscr{H}$ by extending Definition 4.15 such that

$$\mathsf{P}^{[m]\mathscr{H}} = \{[l]x \mid x \in \mathsf{P}^{\mathscr{H}},\ l \in |\mathbb{L}^<|\}.$$

Where does the m in $[m]\mathscr{H}$ appear on the right-hand side here? It does not: m is only used to reduce the support of the interpretations of constant symbols; see Definition 4.15. PNL only allows equivariant (Definition 7.2) interpretation of proposition-formers. If we considered a flavour of PNL in which proposition-formers could receive non-equivariant interpretation (so that in the syntax we would allow terms of the form $(\pi{\cdot}\mathsf{P})(r)$), then Definition 7.9 would mention m on the right. This makes no difference to expressivity since we can emulate the effect of a non-equivariant proposition-former using $\mathsf{P}(C, r)$. Our design follows the path of the simplest definitions and proofs.

Lemma 7.10. *Every $[l]x \in [\![\alpha]\!]^{[m]\mathscr{H}}$ has finite support.*

Proof. As for Lemma 6.4, but now using Notation 7.8 and our assumption that $x \in [\![\alpha]\!]^{\mathscr{H}}$ has medium support. □

Definition 7.11. Extend $atoms(r)$ (Definition 5.1) to propositions $atoms(\phi)$ inductively by:

$$\begin{array}{ll} atoms(\bot) = \varnothing & atoms(\phi \Rightarrow \psi) = atoms(\phi) \cup atoms(\psi) \\ atoms(\mathsf{P}(r)) = atoms(r) & atoms(\forall X.\phi) = atoms(\phi) \end{array}$$

Lemma 7.12 extends Proposition 5.6 to predicates, and is needed for Proposition 7.14. Recall from Definition 5.5 the definition of $\pi \circ \varsigma$:

Lemma 7.12. *Suppose ς is a valuation to an interpretation $\mathscr{H}$. Suppose ϕ is a predicate and π a permutation such that $nontriv(\pi) \subseteq \mathbb{A}^<$ and $nontriv(\pi) \cap atoms(\phi) = \varnothing$.*

Then $[\![\phi]\!]^{\mathscr{H}}_{\pi \circ \varsigma} = [\![\phi]\!]^{\mathscr{H}}_{\varsigma}$.

Proof. By a routine induction on ϕ. We consider two cases:

- *The case of* $\mathsf{P}(r)$. By definition $[\![\mathsf{P}(r)]\!]^{\mathscr{H}}_{\pi \circ \varsigma} = \top$ if and only if $[\![r]\!]^{\mathscr{H}}_{\pi \circ \varsigma} \in \mathsf{P}^{\mathscr{H}}$. By Proposition 5.6 $[\![r]\!]^{\mathscr{H}}_{\pi \circ \varsigma} = \pi \cdot [\![r]\!]^{\mathscr{H}}_{\varsigma}$. By assumption $\mathsf{P}^{\mathscr{H}}$ is equivariant (Definition 7.2).
- *The case of* $\forall X.\phi$.

$$\begin{array}{ll} [\![\forall X.\phi]\!]^{\mathscr{H}}_{\pi \circ \varsigma} & = \bigwedge \{ [\![\phi]\!]^{\mathscr{H}}_{(\pi \circ \varsigma)[X:=x]} \mid x \in [\![sort(X)]\!]^{\mathscr{H}},\ supp(x) \subseteq \mathbb{A}^< \} \\ & = \bigwedge \{ [\![\phi]\!]^{\mathscr{H}}_{(\pi \circ \varsigma)[X:=\pi \cdot x]} \mid x \in [\![sort(X)]\!]^{\mathscr{H}},\ supp(x) \subseteq \mathbb{A}^< \} \\ & = \bigwedge \{ [\![\phi]\!]^{\mathscr{H}}_{\pi \circ (\varsigma[X:=x])} \mid x \in [\![sort(X)]\!]^{\mathscr{H}},\ supp(x) \subseteq \mathbb{A}^< \} \\ & = \bigwedge \{ [\![\phi]\!]^{\mathscr{H}}_{\varsigma[X:=x]} \mid x \in [\![sort(X)]\!]^{\mathscr{H}},\ supp(x) \subseteq \mathbb{A}^< \} \\ & = [\![\forall X.\phi]\!]^{\mathscr{H}}_{\varsigma} \end{array}$$

□

Lemma 7.13. *$fa(\phi) \subseteq \mathbb{A}^< \cup atoms(\phi)$ and $fa(r) \subseteq \mathbb{A}^< \cup atoms(r)$.*

Proof. By a routine induction on Definitions 5.1 and 7.11 and by a routine calculation using part 2 of Theorem 4.14 for the base case of $\pi \cdot X$. □

Proposition 7.14. *Suppose ϕ is a proposition, $\mathscr{H}$ is a PNL interpretation with medium support (Definition 7.4), and ς is a valuation to $\mathscr{H}$. Suppose $A \subseteq \mathbb{A}$ is a finite set of atoms such that $atoms(\phi) \subseteq A$, and suppose $l \in |\mathbb{L}^<|$ and $supp(l) \cap A = \varnothing$.*

Then $\mathscr{H}, \varsigma \vDash \phi$ if and only if $[l]\mathscr{H}, [l]\varsigma \vDash \phi$.

Proof. By induction on ϕ. We consider a selection of cases:

- *The case of* $\mathsf{P}(r)$. We consider the two implications separately.

  ($\Leftarrow$) Suppose $[l]\mathscr{H}, [l]\varsigma \vDash \mathsf{P}(r)$. This means that $[\![r]\!]^{[l]\mathscr{H}}_{[l]\varsigma} \in \mathsf{P}^{[l]\mathscr{H}}$. By Theorem 5.2 $[\![r]\!]^{[l]\mathscr{H}}_{[l]\varsigma} = [l][\![r]\!]^{\mathscr{H}}_{\varsigma}$ (note that $atoms(\mathsf{P}(r)) = atoms(r)$), and so by Definitions 7.9 and 4.3 $\pi' \cdot [\![r]\!]^{\mathscr{H}}_{\varsigma} \in \mathsf{P}^{\mathscr{H}}$ for some $\pi' \in fix(supp([\![r]\!]^{\mathscr{H}}_{\varsigma}) \setminus supp(l))$. By equivariance of $\mathsf{P}^{\mathscr{H}}$ it immediately follows that $[\![r]\!]^{\mathscr{H}}_{\varsigma} \in \mathsf{P}^{\mathscr{H}}$ and so that $\mathscr{H}, \varsigma \vDash \mathsf{P}(r)$.

  ($\Rightarrow$) Now suppose $\mathscr{H}, \varsigma \vDash \mathsf{P}(r)$, so that by definition $[\![r]\!]^{\mathscr{H}}_{\varsigma} \in [\![\mathsf{P}]\!]^{\mathscr{H}}$. As in the previous paragraph by Theorem 5.2 $[l][\![r]\!]^{\mathscr{H}}_{\varsigma} = [\![r]\!]^{[l]\mathscr{H}}_{[l]\varsigma}$. It follows by Definition 7.11 that $[\![r]\!]^{[l]\mathscr{H}}_{[l]\varsigma} \in \mathsf{P}^{[l]\mathscr{H}}$.

- *The case of* $\forall X.\phi$. Again we consider the two implications separately:

  ($\Leftarrow$) Suppose $\mathscr{H}, \varsigma \not\Vdash \forall X.\phi$. Unpacking definitions, this means there is some $x \in |[\![sort(X)]\!]^{\mathscr{H}}|$ with $supp(x) \subseteq \mathbb{A}^<$ and $\mathscr{H}, \varsigma[X{:=}x] \not\Vdash \phi$.

  By inductive hypothesis $[l]\mathscr{H}, [l](\varsigma[X{:=}x]) \not\Vdash \phi$. We can use Lemma 5.4 to write $[l](\varsigma[X{:=}x])$ as $([l]\varsigma)[X{:=}[l]x]$. Furthermore, by assumption $supp(x) \subseteq \mathbb{A}^<$ so by Lemma 4.6 $supp([l]x) \subseteq \mathbb{A}^< \setminus supp(l) \subseteq \mathbb{A}^<$. It follows by Definition 7.3 that $[l]\mathscr{H}, [l]\varsigma \not\Vdash \forall X.\phi$.

  ($\Rightarrow$) Suppose $[l]\mathscr{H}, [l]\varsigma \not\Vdash \forall X.\phi$. Unpacking Definition 7.3 this means there are $x' \in |[\![sort(X)]\!]^{\mathscr{H}}|$ and $l' \in |\mathbb{L}^<|$ such that

  $$supp([l']x') \subseteq \mathbb{A}^< \quad \text{and} \quad [l]\mathscr{H}, ([l]\varsigma)[X{:=}[l']x'] \not\Vdash \phi.$$

  If $supp([l']x') \cap supp(l) = \varnothing$ then we may use Lemma 4.8 and write $[l']x'$ as $[l](([l']x')@l)$ and deduce by inductive hypothesis that $\mathscr{H}, \varsigma \not\Vdash \phi$.

  Otherwise, we choose some π' that maps $supp([l']x') \cap supp(l) \neq \varnothing$ to a set of atoms in $\mathbb{A}^<$ that is disjoint from $supp(l) \cup atoms(\phi)$, and π' fixes all other atoms. This is possible because by construction $supp([l']x')$ is finite and $\mathbb{A}^< \setminus (supp(l) \cup atoms(\phi))$ is infinite (recall that $\mathbb{A}^< \setminus \mathbb{A}^{\ll}$ is assumed infinite). Using Lemma 7.12

  $$[l]\mathscr{H}, ([l]\varsigma)[X{:=}\pi' \cdot ([l']x')] \not\Vdash \phi.$$

  We now proceed as in the case where $supp([l']x') \cap supp(l) = \varnothing$.

- *The case of* $\phi \Rightarrow \psi$. Suppose $\mathscr{H}, \varsigma \vDash \phi \Rightarrow \psi$. This means that $\mathscr{H}, \varsigma \not\Vdash \phi$ or $\mathscr{H}, \varsigma \vDash \psi$. By inductive hypothesis this is if and only if $[l]\mathscr{H}, [l]\varsigma \not\Vdash \phi$ or $[l]\mathscr{H}, [l]\varsigma \vDash \psi$. In either case $[l]\mathscr{H}, [l]\varsigma \vDash \phi \Rightarrow \psi$, and we are done.

□

Theorem 7.15. $\vDash_{fin} \phi$ *if and only if* $\vDash_{\ll} \phi$.

Proof. The right-to-left implication is immediate, just as in Theorem 6.8. The left-to-right implication follows using Proposition 7.14 and Lemma 7.10. □

7.4. Relevance of the theorem

It will help to establish some new terminology:

Notation 7.16. Suppose X is a set with a permutation action and $x \in |\mathsf{X}|$.

- Say the element $x \in |\mathsf{X}|$ is **finite-namespace** when $supp(x)$ is finite. Similarly say X is finite-namespace when every $x \in |\mathsf{X}|$ has finite support.
  This is synonymous with X being a nominal set in the sense of [GP01]; cf. also Definition 6.6.
- Say x is $\mathbb{A}^{\ll}$**-namespace** when $supp(x) \subseteq \pi \cdot \mathbb{A}^{\ll}$ for some π. Similarly say that X is $\mathbb{A}^{\ll}$-namespace when every $x \in |\mathsf{X}|$ is $\mathbb{A}^{\ll}$-namespace.
  This is synonymous with *medium* support from Definition 7.4.
- Say x is $\mathbb{A}^<$**-namespace** when $supp(x) \subseteq \pi \cdot \mathbb{A}^<$ for some π. Similarly say that X is $\mathbb{A}^<$**-namespace** when every $x \in |\mathsf{X}|$ is $\mathbb{A}^{\ll}$-namespace.
  This is synonymous with X being a permissive-nominal set in the sense of Definition 2.7 or [Gab12b].

Similarly we will call interpretations finite-namespace, $\mathbb{A}^{\ll}$-namespace, and $\mathbb{A}^{<}$-namespace in accordance with the support of their underlying sets.

The relevance of Theorem 7.15 is that a PNL predicate is valid over $\mathbb{A}^{\ll}$-namespace interpretations if and only if it is valid over finite-namespace interpretations.[7]

The PNL of [DG10, DG12, Gab12b] has a sequent system giving a notion of logical entailment which is proved sound and complete for $\models$, that is, for the collection of $\mathbb{A}^{<}$-namespace interpretations. This differs from the validity $\models_{\ll}$, which is validity over $\mathbb{A}^{\ll}$-namespace interpretations (those with medium support). This is a more restricted class of models.

Medium support is a new idea to the theory of PNL. When models are restricted, more statements become valid (usually). In this case we get a family of theorems, which is exemplified by Proposition 7.7. It remains to devise a complete proof system for PNL over medium support.

We would not speculate on whether large or medium support is 'better'; we suspect that the situation is similar to the intuitionistic/classical question of whether to allow double negation elimination: sometimes we may want it and sometimes we may not.

The value of Theorem 7.15 is that it tells us that $\mathbb{A}^{\ll}$ is as small as we need go in exploring validity: restricting models of PNL further to smaller namespaces, and in particular to finite support, will not give us any extra valid propositions. As we shall argue in the Conclusions, working with sets with infinite support can be easier than working with sets with finite support, so this matters.

And note the obvious: once we carried out our constructions and applied them to permissive-nominal algebra, we could re-use them for permissive-nominal logic with a relatively slight effort of two pages of mathematics in Subsection 7.3.

8. More permission sets, more permutations

In Definition 2.5 we followed [Gab12b] and took permission sets to be sets of the form $\pi{\cdot}\mathbb{A}^{<}$. This captures a simple assertion language about the atoms permitted in unknowns. The results in this paper are sensitive to the expressivity of this language: if we make it slightly more powerful then the results in this paper fail.

8.1. More permission sets

If we follow e.g. [DGM10] and take permission sets to be sets of the form $(\mathbb{A}^{<} \setminus A) \cup B$ where $A \subseteq \mathbb{A}^{<}$ and $B \subseteq \mathbb{A}^{>}$ are finite, then the results in this paper fail.

This genuinely enlarges the set of permission sets (and so makes the assertion language which they represent, more expressive). For instance, if $b \in \mathbb{A}^{>}$ then there is no finite permutation π such that $\pi{\cdot}\mathbb{A}^{<} = \mathbb{A}^{<} \cup \{b\}$.

Notation 8.1. Write $\mathcal{P}$ for the set of all sets of atoms differing finitely from $\mathbb{A}^{<}$ as just described.

[7] For comparison, *nominal logic* does not have this property [Pit03]: there are nominal logic predicates that are valid of all finite-namespace interpretations but not of all $\mathbb{A}^{\ll}$-namespace interpretations (and thus also not valid of all $\mathbb{A}^{<}$-namespace interpretations). Nominal logic, of course, is a first-order theory; an axiomatisation in first-order logic similar to the axiomatisation of Fraenkel-Mostowski sets from which it is descended. What makes the languages of this paper different is that they are purpose-built using the dedicated new syntax of (permissive-)nominal terms.

Proposition 8.2. *There exists a theory* T *in permissive-nominal algebra with permission sets from* $\mathcal{P}$*, and an assertion* $r' = s'$ *in that theory, such that* $\mathsf{T} \vDash_{\mathit{fin}} r' = s'$ *and* $\mathsf{T} \nvDash r' = s'$ *(where models are permissive-nominal sets with permission sets in* $\mathcal{P}$*).*

Proof. Assume one base type τ and one term former $0 : \tau$ with $\mathit{pmss}(0) = \varnothing$. Assume an axiom $X = 0$ where $\mathit{fa}(X) = \mathbb{A}^<$. Assume an unknown Z with $\mathit{sort}(Z) = \tau$ and $\mathit{fa}(Z) = \mathbb{A}^< \cup \{b\}$ where $b \in \mathbb{A}^>$. Then:

- $\mathsf{T} \vDash_{\mathit{fin}} Z = 0$.

  For suppose $\mathscr{F}$ is an interpretation of T with finite support: then for any $x \in [\![\tau]\!]^{\mathscr{F}}$, there is some finite π with $\mathit{supp}(\pi{\cdot}x) \subseteq \mathbb{A}^<$, hence by our single axiom $\pi{\cdot}x = 0$ and by equivariance $x = \pi^{-1}{\cdot}0 = 0$, since 0 has empty support.

- However, $\mathsf{T} \nvDash Z = 0$.

  To see this, interpret τ to be the set $\{\pi{\cdot}(\mathbb{A}^< \cup \{b\}) \mid \pi \text{ finite}\} \cup \{\varnothing\}$, interpret 0 by $\varnothing$, and take $\varsigma(Z) = \mathbb{A}^< \cup \{b\}$.

□

Initially we used $\mathcal{P}$; notably in [DG10, DGM10, DG12]. However, in later papers such as [Gab12b] we preferred the design of Definition 2.5 because it seemed to make some proofs easier to express. In the light of the results of this paper we can now better understand the significance of our design choices: Proposition 8.2 suggests that our design in Definition 2.5 is mathematically more elementary and somewhat closer to the design 'nominal terms + finitely-supported nominal sets' from the previous literature. That is, the design of Definition 2.5 and [Gab12b] is the closest 'permissive' version of traditional nominal techniques, and the design of [DG10, DGM10, DG12] is slightly but measurably more expressive.

8.2. shift-*permutations*

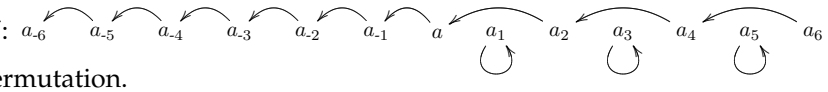
In the presence of infinite permutations, the results in this paper fail. We sketch the mathematics involved, starting with a justification of why infinite permutations are an interesting case to consider.

For simplicity assume a single sort of atom.

Definition 8.3. Suppose $a \in \mathbb{A}^< = \{a, a_{\text{-}1}, a_{\text{-}2}, a_{\text{-}3}, \dots\}$ and $\mathbb{A}^> = \{a_1, a_2, a_3, \dots\}$.

Assume a bijection δ on atoms mapping $\mathbb{A}^<$ to $\mathbb{A}^< \setminus \{a\}$ and such that $\mathbb{A} \setminus \mathit{nontriv}(\pi)$ is infinite (we can do this because we assumed that $\mathbb{A}$ is countable).

We illustrate an example:

Illustration of δ: 

Call δ a **shift**-permutation.

δ has a measurable and favourable effect on the mathematics and algorithmics of nominal syntax. For instance:

- δ nontrivially increases the deductive power of $\forall X$ in PNL [DG12, Subsection 2.7].

  If $\mathit{fa}(X) = \mathbb{A}^<$ where $a \in \mathbb{A}^<$ then $\forall X.\mathsf{R}(X, X)$ does not entail $\mathsf{R}((X, a), (X, a))$ without δ, but it *does* entail $\mathsf{R}((X, a), (X, a))$ *with* δ (for R having an appropriate arity). This extra power is irrelevant if we only care about finitely-supported models, which is why the issue has not arisen in previous work.

- *shift*-permutations can be used to obtain a particularly concise unification algorithm [Gab12b, Section 4].

For more discussion see [Gab12b, Subsection 3.6].

This extra power is not particularly expensive: we can do what we are used to in nominal techniques, in the presence of δ. Indeed, the results of [Gab12b] are parameterised over a permutation group general enough to admit δ because this was *easier* than excluding it. In particular the specific design of the nominal unification algorithm and HSP result there, are shorter and simpler because of their use of δ.

However, in the presence of δ the results of this paper fail. Proposition 8.5 gives an example of a signature for which permissive-nominal algebra $\vDash$ (all permissive-nominal models) is complete, but $\vDash_{\mathit{fin}}$ (models with finite support) are not. In order to state this result we must 'upgrade' the material in this paper with δ.

Definition 8.4. To augment Sections 2 and 3 with a shift permutation δ, we proceed as follows:

1. In Definition 2.3 permutations are finitely generated by swappings *and* δ (they remain finitely representable, but $\mathit{nontriv}(\pi)$ is now not always finite).
   Write $\mathbb{P}_\delta$ for the group of bijections generated by swappings and δ.
2. In Definition 2.6 assume the permutation action has type $(\mathbb{P}_\delta \times |\mathsf{X}|) \to |\mathsf{X}|$. So permissive-nominal sets have an action by swappings and δ.
3. In Definition 2.6 we say that $A \subseteq \mathbb{A}$ supports $x \in |\mathsf{X}|$ when for every permutation $\pi \in \mathbb{P}_\delta$, if $\pi(a) = a$ for all $a \in A$ then $\pi{\cdot}x = x$.[8]
4. In the examples of Subsection 2.3 extend for the extra permutations in the natural way. So $\pi{\cdot}a = \pi(a)$ for $\pi \in \mathbb{P}_\delta$ and $\pi{\cdot}[a]x = [\pi(a)]\pi{\cdot}x$ for $\pi \in \mathbb{P}_\delta$.
5. In Definition 3.3 extend terms also with the extra permutations. So $\pi{\cdot}X$ is a term for $\pi \in \mathbb{P}_\delta$. The permutation action Definition 3.4 extends in the natural way.
6. We extend the notion of equivariance (Definition 3.9) with the extra permutations. So F is *equivariant* when $F(\pi{\cdot}x) = \pi{\cdot}F(x)$ for all permutations $\pi \in \mathbb{P}_\delta$ and $x \in |\mathsf{X}|$.[9]

Proposition 8.5. *There exists a theory* T *in permissive-nominal algebra with* δ*, and an assertion* $r' = s'$ *in that theory, such that* $\mathsf{T} \vDash_{\mathit{fin}} r' = s'$ *and* $\mathsf{T} \nvDash r' = s'$ *(where models are permissive-nominal sets with* δ*).*

Proof. Assume no term-formers and one base type τ. Assume $a \in \mathbb{A}^<$ and a shift permutation δ bijecting $\mathbb{A}^<$ with $\mathbb{A}^< \setminus \{a\}$, as illustrated just after Definition 8.3.

Assume an axiom $(b\ a){\cdot}X = X$ where $b \notin \mathbb{A}^<$. Then:

- If $\mathscr{F}$ is a model of T with finite support then $\mathit{supp}(x) = \varnothing$ for every $x \in [\![\tau]\!]^{\mathscr{F}}$. For suppose there exists x with $\mathit{supp}(x) \neq \varnothing$. By equivariance we may (apply a permutation to x to) assume without loss of generality that $a, b \notin \mathit{supp}(x)$.

[8] This is a little stronger than we need. We could also retain the condition that π be finite in the definition of supporting set, so we say that $A \subseteq \mathbb{A}$ supports $x \in |\mathsf{X}|$ when for every *finite* permutation $\pi \in \mathbb{P}_{\mathrm{fin}}$ (so no δ), if $\pi(a) = a$ for all $a \in A$ then $\pi{\cdot}x = x$. We only ever α-convert by finitely many atoms in this paper, so the proofs remain unchanged.

What does happen is that we admit models with elements which are fixed by finite permutations, but perhaps not by δ. For more on this design see [DG12], in particular Remark 3.3.

[9] We then call the notion of equivariance from Definition 3.9 **finite equivariance**. It is possible to be equivariant for finite permutations but not for δ. The proof of Proposition 8.5 will depend on this.

Now choose some $a' \in \mathit{supp}(x)$ and choose some π mapping $\mathit{supp}(x)$ to a subset of $\mathbb{A}^<$ and such that specifically $\pi(a') = a$. By our axiom, $(b\ a)\cdot(\pi\cdot x) = \pi\cdot x$. It follows by calculations on permutations that $(b\ a')\cdot x = x$ and so by Corollary 2.11 that $a' \notin \mathit{supp}(x)$, a contradiction.
Thus, $\mathsf{T} \vDash_{\mathit{fin}} \delta\cdot Y = Y$.

- $\mathsf{T} \nvDash \delta\cdot Y = Y$. To see this consider the elements $x_i = \{(\pi\circ\delta^i)\cdot\mathbb{A}^< \mid \pi \text{ finite}\}$ with the pointwise action, for every $i \in \mathbb{Z}$ (where $\mathbb{Z}$ is the integers; see Definition 2.1).
It is a fact that $(b\ a)\cdot x_i = x_i$, but it is also a fact that $\delta\cdot x_i = x_{i+1} \neq x_i$. We interpret τ to be the set $\{x_i \mid i \in \mathbb{Z}\}$ and see that $(b\ a)\cdot x_i = x_i$ for every i so the axiom $(b\ a)\cdot X = X$ is valid, but $\delta\cdot x_0 \neq x_0$ so $\mathsf{T} \nvDash \delta\cdot X = X$.
This observation is exactly the *fuzzy support* noted in [Gab07], see also Remark 3.3 from [DG12].

□

Remark 8.6. Where do the proofs fail? Failure occurs in the interaction of $\mathit{atoms}(r)$ (Definition 5.1) with Theorem 5.2.

The reasonable definition for $\mathit{atoms}(\delta\cdot X)$ is $\mathit{nontriv}(\delta)/\mathbb{A}^<$, which is infinite. This gives us infinitely many atoms to avoid in order to guarantee $\mathit{atoms}(r) \cap \mathit{supp}(l) = \varnothing$ in Theorem 5.2. Thus, $\mathit{supp}([l]\llbracket r\rrbracket^{\#}_{\zeta})$ need not be finite.

As a corollary we can clarify something that is evident but only semi-formal in previous work: permissive-nominal terms with δ are strictly more expressive than permissive-nominal terms without δ, and also strictly more expressive than 'ordinary' nominal terms.

9. Conclusions

We have seen permissive-nominal sets and how, given a permissive-nominal set X, we can build a corresponding nominal set $[\mathbb{L}]\mathsf{X}$ from X by applying to each $x \in |\mathsf{X}|$ an infinite simultaneous atoms-abstraction abstracting all but finitely many of the atoms in $\mathit{supp}(x)$. We have used this to translate between interpretations with differently sized sets of support, and we have used this translation to translate between different notions of validity for permissive-nominal syntaxes.

It can be easier to work with permissive-nominal models—even dramatically easier. To see an example, compare the direct completeness proof for nominal algebra with respect to finitely-supported models in [GM09] (subsections 4.3 and 4.4; over five pages including a 'trick') with the completeness proof for the permissive-nominal algebra used in this paper with respect to permissive-nominal models in [Gab12b] (subsection 7.5; under two pages, and the maths is straightforward). Even more extreme, compare the proof of the Nominal HSPA theorem from [Gab09] (twenty-eight pages) with the permissive-nominal HSP theorem from [Gab12b] (five pages).[10]

What this means is that—based on this author's experience—even if the reader is interested specifically in finitely-supported models, it might be shorter and cleaner to prove completeness with respect to some flavour of infinitely-supported permissive-nominal models first, and then to use this paper off-the-shelf.

[10]This is unfair. For instance, the five pages do not include setting up the syntax. Still, looking at the maths, a leap in difference in complexity is clear.

de Bruijn indexes. The technical construction at the heart of this paper, $[\mathbb{L}]\mathsf{X}$ from Definition 4.3, is arguably reminiscent of de Bruijn indexes [dB72]. Given an $x \in |\mathsf{X}|$ with infinite support, we form $[l]x$ where $supp(x) \setminus supp(l)$ is finite. In doing this, we in effect convert all but finitely many of the atoms in $supp(x)$ into numerical indexes, where a is identified by the position in l at which it occurs (if any).

Of course this is not a literal replacement in x, since we assume no internal structure. But it is an abstract de Bruijn indexing in the sense that binding names in order corresponds to turning them into numerical indexes. An explicit connection is made in [Gab07, Section 4] as mentioned below.

Infinite sets of atoms in the literature. The notions of infinite support, infinite lists of atoms, and infinite simultaneous atoms-abstraction were considered by the author in [Gab02, Gab07].

Translations between nominal abstract syntax, name-carrying syntax, and de Bruijn syntax were given in [Gab07, Section 4]. The precise definition used in this paper is different and tailored to our intended application (e.g. we restrict to abstractions such that $supp([l]x)$ is finite), and of course, we concentrate on things other than abstract syntax.

The notion of not-necessarily-finite support was also raised in [Pit03], and Cheney took up the suggestion in [Che06]. The *support ideals* considered there are not quite the same as the permission sets considered here (for instance, permission sets in this paper are never finite, and the emphasis on well-orderings is absent in Cheney's work), but the spirit of the maths is similar.

Implicit connection with unknowns. A non-evident connection exists between $[l]x$ and unknowns, which this paper has not explored. Elsewhere we explore how a model of unknowns X is given by infinite well-orderings of permission sets [Gab11b, Gab12a].

One way to view $[l]x$ is as 'x abstracted by an unknown X'.

Now $\mathbb{L}$ from Definition 4.1 is a *single permutation orbit* under finite permutations of some list l_* of atoms. This author calls this a *namespace*—$\mathbb{L}$ is a namespace, that is, a set of sets of atoms (in order) obtained by permuting *finitely* many of them at a time. We go from X to $[\mathbb{L}]\mathsf{X}$ essentially by abstracting a namespace, and because an unknown identifies a namespace, this can be read as a (level 1) abstraction of (the atoms in) a level 2 variable.

This paper does not make anything of these connections, but they exist in the background. At some point, we hope to produce a broader account which will bring the threads in the various papers together and makes clearer the overall picture. For now, the results in this paper have independent interest as discussed above.

shift-*permutations.* We concluded the technical part of this paper in Section 8 by discussing *shift*-permutations. These infinite permutations are useful and mathematically well-behaved, but they mark a point at which permissive-nominal techniques go strictly beyond the expressivity of nominal techniques, and this is made formal: we saw in Section 8 how the results of Section 6 depend on permutations π being finite and when we include infinite permutations in syntax, the results begin to fail. This is reasonable and as it should be.

So a lesson we can draw from this paper and from the translation in [DGM10, Section 4] is this: permissive-nominal terms with finite permutations are essentially the

same thing as (but somewhat better-behaved than) 'ordinary' nominal terms; permissive-nominal terms with possibly infinite permutations are different, and they are strictly more expressive.

Non-equivariance of $atoms$. One curious aspect of our proofs is that the function $atoms$ from Definition 5.1, which plays a role in Section 5, is not invariant under α-equivalence. For instance, $atoms([a]X) = \{a\}$ and $atoms([b](b\ a)\cdot X) = \{b, a\}$ (where $a \in \mathbb{A}^<$ and $b \in \mathbb{A}^>$).

It is shown in [Gab11b] and [Gab12b] that valuations ς (Definition 3.12) can be thought of as (finite-)equivariant maps out of *moderated* unknowns $\pi\cdot X$ considered as a permissive-nominal set.

The non-equivariance of $atoms$ is an artefact of the fact that the syntax of this paper amounts to choosing $id\cdot X$ as a representative of the permutation equivalence class $\{\pi\cdot X \mid \pi \text{ a permutation}\}$. Permissive-nominal syntax is *already* non-equivariant, because for each unknown-up-to-permutations we have chosen a canonical representative X.

None of this matters for the proofs here, because we only care about avoiding capture with finitely many atoms.

Final words on set theory. The results of this paper are reminiscent of the upwards and downwards Löwenheim-Skolem theorems, which express that a first-order theory cannot 'count' the cardinality of its infinite models [Hod93]. The construction of this paper can be read as saying that first-order permissive-nominal syntax with finite permutations cannot 'count' the cardinality of its supporting sets.

We believe it would be fairly easy to strengthen and generalise this result to the following: first-order nominal syntax cannot 'count' the cardinality of the set of atoms or its supporting sets, so long as these are no smaller than the permutations in that syntax. Making this formal would require us to be just a little systematic but it should not be too hard using a free construction—and the syntax should be a natural generalisation sufficient to subsume permissive-nominal algebra and permissive-nominal logic.

More generally, we can also ask how the group of permutations can be independently enlarged or restricted in syntax and in the denotation. For instance, in this paper we have considered syntax and semantics using the *same* group of permutations. But the semantics could allow more permutations than the syntax, e.g. we could allow *shift* in the denotation (this is useful to 'make support smaller' in some element), but not in the syntax (so that we might avoid the issues discussed in Subsection 8.2 and specifically in Remark 8.6). In short, we see this paper as the first of a family of similar results which may become useful if and when further variations on logics based on nominal terms, and their models, are developed. We leave these thoughts to future work.

Acknowledgements.

Many thanks to an anonymous referee, without whose careful attention this paper would not have reached its current form. We acknowledge the support of the Leverhulme Trust and of grant RYC-2006-002131 at the Polytechnic University of Madrid.

References

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

[dB72] Nicolaas G. de Bruijn. Lambda calculus notation with nameless dummies, a tool for automatic formula manipulation, with application to the Church-Rosser theorem. *Indagationes Mathematicae*, 5(34):381–392, 1972.

[DG10] Gilles Dowek and Murdoch J. Gabbay. Permissive Nominal Logic. In *Proceedings of the 12th International ACM SIGPLAN Symposium on Principles and Practice of Declarative Programming (PPDP 2010)*, pages 165–176, 2010.

[DG12] Gilles Dowek and Murdoch J. Gabbay. Permissive Nominal Logic (journal version). *Transactions on Computational Logic*, 2012. In press.

[DGM10] Gilles Dowek, Murdoch J. Gabbay, and Dominic P. Mulligan. Permissive Nominal Terms and their Unification: an infinite, co-infinite approach to nominal techniques (journal version). *Logic Journal of the IGPL*, 18(6):769–822, 2010.

[Gab02] Murdoch J. Gabbay. FM-HOL, a higher-order theory of names. In F. Kamareddine, editor, *35 Years of Automath*, April 2002.

[Gab07] Murdoch J. Gabbay. A General Mathematics of Names. *Information and Computation*, 205(7):982–1011, July 2007.

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

[Gab11a] Murdoch J. Gabbay. Foundations of nominal techniques: logic and semantics of variables in abstract syntax. *Bulletin of Symbolic Logic*, 17(2):161–229, 2011.

[Gab11b] Murdoch J. Gabbay. Two-level nominal sets and semantic nominal terms: an extension of nominal set theory for handling meta-variables. *Mathematical Structures in Computer Science*, 21:997–1033, 2011.

[Gab12a] Murdoch J. Gabbay. Meta-variables as infinite lists in nominal terms unification and rewriting. *Logic Journal of the IGPL*, 2012. In press.

[Gab12b] Murdoch J. Gabbay. Nominal terms and nominal logics: from foundations to meta-mathematics. In *Handbook of Philosophical Logic*, volume 17. Kluwer, 2012.

[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, July 2007.

[GM09] Murdoch J. Gabbay and Aad Mathijssen. Nominal universal algebra: equational logic with names and binding. *Journal of Logic and Computation*, 19(6):1455–1508, December 2009.

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

[Hod93] Wilfrid Hodges. *Model theory*. Cambridge University Press, 1993.

[Mul10] Dominic P. Mulligan. Online nominal bibliography. www.citeulike.org/group/11951/, 2010.

[New42] M.H.A. Newman. On theories with a combinatorial definition of equivalence. *Annals of Mathematics*, 43(2):223–243, 1942.

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

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