Nominal renaming sets (technical report)

Murdoch J. Gabbay

December 19, 2007

Abstract

Nominal techniques are based on the idea of sets with a finitely-supported atoms-permutation action.

In this paper we consider the idea of sets with a finitely-supported atoms-renaming action (renamings can identify atoms; permutations cannot). We show that these exhibit many of the useful qualities found in traditional nominal techniques; an elementary sets-based presentation, inductive datatypes of syntax up to binding, cartesian closure, and being a topos. Unlike in nominal techniques, the notion of names-abstraction coincides with functional abstraction.[1]

1 Introduction

It can be surprisingly difficult to find the right approach to handling variable symbols in abstract syntax. We need to do this in order to process sentences in formal languages, such as λ-calculus terms or logical predicates. Binders are particularly troublesome and they arise frequently: for example with λ-binding, $\forall$-quantification, declarations of formal parameters in procedures, name-restriction as in the π-calculus, and so on.

To deal with binders informally is usually not a problem (though sometimes significant errors are made). The difficulty is in handling binders in implementation, notably in program transformation systems and theorem provers.

My thesis [10] started the 'nominal' approach (supervised by Pitts). This is based on a structure called a *nominal set*. Nominal sets ([14] or Definition 4.3) are based on two ideas:

- An underlying set with a permutation action, with
- finite support.

Remarkably, the two ideas of 'permutation action' and 'finite support' suffice to build an account of names and binding in abstract syntax. That is, we can construct a set which is an inductive datatype of syntax-up-to-binding, and develop programming and reasoning principles on it.

In this paper we revisit these ideas and ask: how much of 'nominal' *had* to be that way, and how much of it depends on the choices which we happened

[1] I am grateful to Andrew Pitts for his comments on these ideas while he was supervising my thesis. I am also indebted to Martin Hofmann for his comments and suggestions during a visit to Munich, to Marino Miculan, to David Richter, and to Michael Gabbay.

to make at the time? During my thesis I suggested an alternative nominal approach[2] based on:

- An underlying set with a renaming action (Definition 2.8), with
- finite support.

It turns out that

'renaming action' and 'finite support'

is just as sufficent as

'permutation action' and 'finite support'

to capture much of the flavour of nominal techniques.

In this paper we shall explore this idea and show how it is possible to build an account of names and binding very similar in spirit to that familiar in the nominal literature based on permutations — only using renamings.

A few words on notation.

- We will call 'nominal sets' from the literature ([14, 10] or Definition 4.3) *nominal permutation sets*. This is to distinguish them from *nominal renaming sets* (Definition 2.8) introduced in this paper.
- We will write function application both as $f(x)$ and fx.
- $a \notin \mathit{supp}(x)$ ('a is not in the support of x') and $a\#x$ ('a is fresh for x') have the same meaning (Remark 2.14).

2 Nominal renaming sets

2.1 Definitions and important results

We use standard sets notation; X, Y, Z range over sets of elements and $x \in X$ means 'x is in X'. Note that $x, y, \ldots \in X$ means 'x, y, ... are in X'. If X and Y are sets then $X \to Y$ is the set of all functions from elements of X to elements of Y. This arrow should not be confused with two other arrows which come later: $\longrightarrow$ (arrows in a category) and $\Rightarrow$ (exponentials).

Definition 2.1. *Fix a countably infinite* **set of atoms** $\mathbb{A}$. *We assume that atoms are disjoint from numbers* $0, 1, 2, \ldots$*, truth-values* $\bot, \top$*, and other standard mathematical entities.*[3] *We let* $a, b, c, \ldots$ *range over atoms.*

We will adhere to a **permutative convention** *that* a, b, c *range* permutatively *over atoms, so for example* a *and* b *range over any two distinct atoms and* $a \neq b$ *always.*

Remark 2.2. We adopt a permutative convention because one of the major uses of atoms is to model variable symbols in abstract syntax (see [14] or Theorem 6.11); it is a fact that 'x' and 'y' are two distinct variable symbols. Later a valuation might associate x and y to the same value, or a subtitution might substitute x and y for the same term. Then, we might write '$x = y$'. However, we must not forget that this is a different kind of equality which arises only after the abstract syntax has been constructed and associated to a denotation.

[2] Unpublished manuscript discussed with Andrew Pitts and later with Martin Hoffman.

[3] $\mathbb{A}$ is a collection of *urelemente* [4, 14, 12].

Remark 2.3. The slogan in nominal techniques is:

> Names exist.

Definition 2.1 makes that formal.

For comparison, techniques based on Higher-Order Abstract Syntax [23] (variable symbol are modelled by variables, which exist at the meta-level), and de Bruijn indexes [6] (variable symbols are numbered indexes, pointing to positions) do not give names concrete reality in the underlying sets. In the context of the current state of the art, Definition 2.1 is not taken for granted.

Definition 2.4. *Let* Fin *be the set of functions* $\sigma \in \mathbb{A} \to \mathbb{A}$ *such that there exists some finite* $S \subseteq \mathbb{A}$ *such that*

$$\text{for all } b \in \mathbb{A} \setminus S \text{ it is the case that } \sigma(b) = b.$$

σ, τ *will range over elements of* Fin*. We call these* **(finitely supported) renamings**.

Definition 2.5. *Write* $[a_1 \mapsto y_1, \ldots, a_k \mapsto y_k]$ *for the function that maps* a_i *to* y_i *for* $1 \leq i \leq k$ *and maps all other* b *(that is, atoms* b *not in the set* $\{a_1, \ldots, a_k\}$*) to themselves.*

Write $[a \mapsto b]$ *for the function such that*

$$[a \mapsto b](a) = b \qquad [a \mapsto b](b) = b \quad \textit{and} \quad [a \mapsto b](c) = c.$$

Call this an **atomic renaming**. *Intuitively,* $[a \mapsto b]$ *'maps* a *to* b*'.*

Write $(a\ b)$ *for the function such that*

$$(a\ b)(a) = b \qquad (a\ b)(b) = a \qquad (a\ b)(c) = c.$$

Consistent with [14], call this a **swapping**. *Intuitively,* $(a\ b)$ *'swaps* a *and* b*'.*

We write $\circ$ *for functional composition. For example* $[a \mapsto b] \circ [b \mapsto a] = [a \mapsto b]$ *(and* $[a \mapsto b] \circ [b \mapsto a] \neq [a \mapsto b, b \mapsto a]$*). Write* id *for the* **identity renaming**. $id(a) = a$ *always.*

Lemma 2.6. Fin *with* $\circ$ *and* id *is a monoid.*

Proof. That is, $id \circ \sigma = \sigma \circ id = \sigma$ and $\sigma \circ (\sigma' \circ \sigma'') = (\sigma \circ \sigma') \circ \sigma''$. This follows by elementary properties of functional composition. □

Definition 2.7. *If* $S \subseteq \mathbb{A}$ *and* $\sigma \in \mathsf{Fin}$*, write* $\sigma|_S$ *for the partial function defined by*

- $\sigma|_S(a) = a$ *if* $a \in S$.
- $\sigma|_S(a)$ *is undefined if* $a \in \mathbb{A} \setminus S$.

Note that if $\sigma' \in \mathsf{Fin}$ *then*

$$\sigma|_S = \sigma'|_S \quad \textit{means} \quad \textit{for all } a \in \mathbb{A}, \textit{ if } a \in S \textit{ then } \sigma(a) = \sigma'(a).$$

Definition 2.8. *A* **nominal renaming set** $\mathbb{X}$ *is a pair* $(|\mathbb{X}|, \cdot)$ *of an* **underlying set** $|\mathbb{X}|$ *and* **finitely-supported renaming action** $\cdot$.

A finitely-supported renaming action $\cdot$ *is a function from* $\mathsf{Fin} \times |\mathbb{X}|$ *to* $|\mathbb{X}|$ *such that:*

- $id \cdot x = x$ *for all* $x \in |\mathbb{X}|$.
- $\sigma' \cdot (\sigma \cdot x) = (\sigma' \circ \sigma) \cdot x$ *for all* $x \in |\mathbb{X}|$ *and* $\sigma', \sigma \in \mathsf{Fin}$.
- *For every* $x \in |\mathbb{X}|$ *there exists some finite* $S \subseteq \mathbb{A}$ *such that for all* $\sigma, \sigma' \in \mathsf{Fin}$ *if* $\sigma|_S = \sigma'|_S$ *then* $\sigma \cdot x = \sigma' \cdot x$.

  We say that every $x \in |\mathbb{X}|$ *has* **finite support**.

Remark 2.9. Some jargon: a nominal renaming set $\mathbb{X}$ is a set with a finitely-supported Fin-monoid action.

Remark 2.10. $supp(x)$ coincides with 'the variables in' if x is abstract syntax; if the reader thinks of $fv(t)$ every time they see $supp(x)$, then they will not go too far wrong [14, Example 6.11]. However support is an abstract notion valid for any nominal renaming set. As we shall see, there are very many nominal renaming sets that are not built of abstract syntax.

2.2 The support of a nominal renaming set

Definition 2.11. *Suppose that* $\mathbb{X}$ *is a nominal renaming set and* $x \in |\mathbb{X}|$. *Say that* $S \subseteq \mathbb{A}$ ***supports*** x *when for all* $\sigma, \sigma' \in \mathsf{Fin}$, *if* $\sigma|_S = \sigma'|_S$ *then* $\sigma \cdot x = \sigma' \cdot x$.

Lemma 2.12 and Theorem 2.13 echoes [14, Proposition 3.4] (definition of support for a nominal permutation set). The proofs for nominal renaming set are simpler:

Lemma 2.12. *Suppose that* $\mathbb{X}$ *is a nominal renaming set and suppose that* $x \in |\mathbb{X}|$. *If* $S \subseteq \mathbb{A}$ *supports* x *and* $S' \subseteq \mathbb{A}$ *supports* x *then* $S \cap S'$ *supports* x.

Proof. Suppose that $S \subseteq \mathbb{A}$ and $S' \subseteq \mathbb{A}$. Suppose that $\sigma|_{S \cap S'} = \sigma'|_{S \cap S'}$. Define $\sigma'' \in \mathsf{Fin}$ by

- $\sigma''(a) = \sigma(a)$ if $a \in S$.
- $\sigma''(a) = \sigma'(a)$ if $a \in \mathbb{A} \setminus S$.

$\sigma''|_S = \sigma|_S$ so $\sigma'' \cdot x = \sigma \cdot x$. Also it is not hard to verify that $\sigma''|_{S'} = \sigma'|_{S'}$ so $\sigma'' \cdot x = \sigma' \cdot x$. The result follows. □

Theorem 2.13. *Suppose that* $\mathbb{X}$ *is nominal renaming set and suppose* $x \in |\mathbb{X}|$.

- *There exists a unique least set of atoms supporting* x, *and that set is finite. Write it* $supp(x)$ *and call it the* **support** *of* x.
- *Suppose that* $\sigma \in \mathsf{Fin}$ *and* $\sigma' \in \mathsf{Fin}$. *Then if* $\sigma|_{supp(x)} = \sigma'|_{supp(x)}$ *then* $\sigma \cdot x = \sigma' \cdot x$.

Proof. By assumption there exist some finite $S \subseteq \mathbb{A}$ supporting x. The first part follows immediately using Lemma 2.12. The second part also follows immediately using Definition 2.11. □

Remark 2.14. Consistent with other work we write $a\#x$ to mean $a \notin supp(x)$ and read this as 'a is fresh for x'. We may also write $a\#x, y$ to mean '$a\#x$ and $a\#y$', and so on.

Definition 2.15. *Let* PFin *be the nominal renaming set such that:*

- $|\mathsf{PFin}|$ *is the collection of finite sets of atoms.*
- *If* $S \subseteq \mathbb{A}$ *is finite and* $\sigma \in \mathsf{Fin}$ *then*

$$\sigma \cdot S = \{\sigma(a) \mid a \in S\}.$$

As is standard, we call this the **pointwise** *renaming action.*

It is not hard to prove that $supp(S) = S$ always.

The following result generalises a known property of nominal permutation sets:

Lemma 2.16. *Recall the definition of* $\sigma \cdot S$ *from Definition 2.15. Suppose that* $\mathbb{X}$ *is nominal renaming set and suppose* $x \in |\mathbb{X}|$ *and* $\sigma \in \mathsf{Fin}$*. Then:*

1. $supp(\sigma \cdot x) \subseteq \sigma \cdot supp(x)$.
2. *If* σ *is injective on* $supp(x)$*, that is, if*

$$a \in supp(x) \text{ and } y \in supp(x) \text{ and } \sigma(y) = \sigma(a) \text{ imply } y = a$$

then

$$supp(\sigma \cdot x) = \sigma \cdot supp(x).$$

Proof. 1. Suppose that $\sigma'(a) = a$ for all $a \in \sigma \cdot supp(x)$. Then $(\sigma' \circ \sigma)|_{supp(x)} = \sigma|_{supp(x)}$. By assumption $\sigma' \cdot (\sigma \cdot x) = (\sigma' \circ \sigma) \cdot x$. By Theorem 2.13 $(\sigma' \circ \sigma) \cdot x = \sigma \cdot x$. The result follows.

2. By part 1 of this result $supp(\sigma \cdot x) \subseteq \sigma \cdot supp(x)$. We need only prove the reverse inclusion; this is routine:

   Suppose that $\sigma|_{supp(x)} = [a_1 \mapsto y_1, \ldots, a_n \mapsto y_n]$. By assumption if $y_i = y_j$ then $i = j$ for $1 \leq i, j \leq n$. So we can write $\sigma' = [y_1 \mapsto a_1, \ldots, y_n \mapsto a_n]$. By Theorem 2.13 $\sigma' \cdot \sigma \cdot x = x$. By part 1 of this result $supp(\sigma' \cdot \sigma \cdot x) \subseteq \sigma' \cdot \sigma \cdot supp(x)$. But $\sigma' \cdot \sigma \cdot supp(x) = supp(x)$ by construction, so we are done.

□

For a counterexample to converse of the inclusion in part 1 of Lemma 2.16, see Lemma 2.18.

2.3 Examples of nominal renaming sets

The set of atoms.

$\mathbb{A}$ with action $\sigma \cdot a = \sigma(a)$ is a nominal renaming set. $supp(a) = \{a\}$.

Above, we wrote $\mathbb{A}$ for the set of atoms. We will not be pedantic enough to change notation now, so we write $\mathbb{A}$ for both $\mathbb{A}$ (the nominal renaming set) and $\mathbb{A}$ (its underlying set). For all other nominal rewriting sets $\mathbb{X}$ we will distinguish between $\mathbb{X}$ and $|\mathbb{X}|$.

Sets with a trivial renaming action.

The empty nominal renaming set $\mathbb{0} = (\{\}, \cdot)$ and the unit nominal renaming set $\mathbb{1} = (\{*\}, \cdot)$ are nominal renaming sets; in both cases there is only one possible renaming action.

More generally, call a nominal renaming set $\mathbb{X} = (|\mathbb{X}|, \cdot)$ **trivial** when $\sigma \cdot x = x$ for all $x \in |\mathbb{X}|$ and $\sigma \in \mathsf{Fin}$.

Then $\mathbb{B} = (\{\top, \bot\}, \cdot)$ and $\mathbb{N} = (\{0, 1, 2, \ldots\}, \cdot)$, both with the trivial renaming actions, are nominal renaming sets. $supp(x) = \emptyset$ always.

Finite and cofinite sets of atoms.

Recall the example of PFin (the set of finite sets of atoms, with pointwise renaming action) from Definition 2.15.

Definition 2.17. *Call $S \subseteq \mathbb{A}$* **cofinite** *when $\mathbb{A} \setminus S$ is finite.*

The set of cofinite subsets of $\mathbb{A}$ is a nominal renaming set with action given by

$$\sigma \cdot S = \mathbb{A} \setminus (\sigma \cdot (\mathbb{A} \setminus S))$$

is a nominal renaming set. It is not hard to verify that $supp(S) = \mathbb{A} \setminus S$ always.

For example, $[a{\mapsto}b] \cdot (\mathbb{A} \setminus \{a, b\}) = \mathbb{A} \setminus \{b\}$.

Finally, the set of all finite or cofinite sets of atoms is a nominal renaming set with action obtained by combining the two actions just given, is a nominal renaming set. $supp(S) = S$ if S is finite, and $supp(S) = \mathbb{A} \setminus S$ if S is cofinite.

Examples of sets with renaming actions that are not nominal renaming sets.

The set of all subsets of $\mathbb{A}$ is not a nominal renaming set. It has a pointwise renaming action given by $\sigma \cdot S = \{\sigma(a) \mid a \in S\}$, but this action is not finitely-supported.

For example if we write $\mathbb{A} = \{a_1, a_2, a_3, a_4, \ldots\}$ and let $A = \{a_{2i} \mid 1 \leq i\}$ — so A is 'every other atom' — then A has no finite supporting set. See also Lemma 2.20 below.

Exploding models.

The converse inclusion in part 1 of Lemma 2.16 need not hold. That is:

Lemma 2.18. *There exists a nominal renaming set $\mathbb{X}$, $x \in |\mathbb{X}|$, and $\sigma \in \mathsf{Fin}$ such that $supp(\sigma \cdot x) \neq \sigma \cdot supp(x)$.*

Proof. It suffices to find an example. Consider $\mathsf{PFin}_{exploding}$ specified by:

- $|\mathsf{PFin}_{exploding}|$ is the set of all finite subsets of $\mathbb{A}$.
- $\sigma \cdot_{exploding} S = \{\sigma(a) \mid a \in S\}$ if $\{\sigma(a) \mid a \in S\}$ has the same cardinality as S (that is, if $\sigma|_S$ is injective as a function out of S).
- $\sigma \cdot_{exploding} S = \emptyset$ otherwise (that is, if there exist $a \in S$ and $b \in S$ such that $\sigma(a) = \sigma(b)$).

It is not hard to verify that this *is* a nominal renaming set, and $supp(S) = S$.

Note that

$$\begin{aligned} [a{\mapsto}b] \cdot supp(\{a,b\}) &= [a{\mapsto}b] \cdot \{a,b\} = \{b\} \quad \text{and} \\ supp([a{\mapsto}b] \cdot_{exploding} \{a,b\}) &= supp(\emptyset) = \emptyset, \end{aligned}$$

and $\emptyset \neq \{b\}$. □

I call this behaviour 'exploding' because, for example, $\{a,b,c\}$ 'explodes' like a balloon when 'pricked' by the non-injective function $[a{\mapsto}b]$.

Suppose $\mathbb{X}$ is a nominal renaming set, suppose $\sigma \in \mathsf{Fin}$, and suppose $x \in |\mathbb{X}|$.

If σ is injective on $supp(x)$ then its action is invertible in the sense that if we are given $\sigma \cdot x$ and σ then we can uniquely deduce what x must have been (the construction is in the proof of part 2 of Lemma 2.16).

Nominal permutative techniques [14] only admit permutations, and their action is always invertible, so this is the only situation considered so far in the literature on nominal techniques.

If $\sigma \in \mathsf{Fin}$ is not injective on $supp(x)$ then its action need not be invertible. For example σ acts pointwise on finite $S \subseteq \mathbb{A}$ (Definition 2.15) and

$$[a{\mapsto}b] \cdot \{a,b\} = \{b\} = [a{\mapsto}b] \cdot \{b\}.$$

So given $\{b\}$ and $[a{\mapsto}b]$, there is no unique finite $S \subseteq \mathbb{A}$ such that $[a{\mapsto}b] \cdot S = \{b\}$.

The point which exploding models make is that the action on the underlying set of a nominal renaming set might be far more non-invertible than we expect, if our intuitions are based only on the behaviour of renamings acting on abstract syntax.

(Possibly infinite) tuples.

Definition 2.19. *Let D be a (possibly infinite) indexing set. Let $\mathbb{X}_I$ for $I \in D$ be an D-indexed collection of nominal renaming sets. Write $(x_I)_{I \in D}$ or just (x_I) for a D-tuple of elements $x_I \in |\mathbb{X}_I|$. For example $(x_I)_{I \in \{1,2\}}$ is an ordered pair. $\Pi_{I \in D}|\mathbb{X}_I|$ (the set of all D-tuples of elements from $|\mathbb{X}_I|$) inherits a pointwise renaming action given by $\sigma \cdot (x_I) = (\sigma \cdot x_I)$.*

Lemma 2.20. *1. (x_I) need not have finite support in general.*

2. (x_I) has finite support if and only if there exists some finite $S \subseteq \mathbb{A}$ such that $supp(x_I) \subseteq S$ for all $I \in D$.

Proof. 1. Let $D = \mathbb{A}$ and consider the tuple (a_I) — the tuple of all atoms, in some order. It is not hard to prove that this has no finite supporting set.

2. Suppose that $S \subseteq \mathbb{A}$ is finite and supports (x_I). Suppose that $I' \in D$; we show that S supports $x_{I'}$. If $\sigma|_S = id|_S$ then $\sigma \cdot (x_I) = (x_I)$. By definition $\sigma \cdot (x_I) = (\sigma \cdot x_I)$, so $\sigma \cdot x_{I'} = x_{I'}$. It follows that S supports $x_{I'}$.

   Now suppose that $S \subseteq \mathbb{A}$ is finite and supports x_I for every $I \in D$. By a similar calculation it is easy to verify that S supports (x_I).

□

Using notation and terminology from Definition 2.19 and Lemma 2.20 we define:

Definition 2.21. *Let D be a (possibly infinite) indexing set. Let $\mathbb{X}_I$ for $I \in D$ be a D-indexed collection of nominal renaming sets. Then let $\Pi_{I\in D}\mathbb{X}_I$ be the nominal renaming set with:*

- *$|\Pi_{I\in D}\mathbb{X}_I|$ is those $(x_I) \in \Pi_{I\in D}|\mathbb{X}_I|$ with finite support.*

  That is, $x_I \in |\mathbb{X}_I|$ for each $I \in D$ and there exists some finite $S \subseteq \mathbb{A}$ such that for all $I \in D$ it is the case that $supp(x_I) \subseteq S$.

- *Pointwise renaming action.*

  That is, $\sigma \cdot (x_I) = (\sigma \cdot x_I)$.

2.4 Renaming action vs. atomic renaming action

Recall from Definition 2.5 the *atomic renaming* $[a{\mapsto}b] \in \mathsf{Fin}$ which maps a to b, and *swapping* $(a\ b) \in \mathsf{Fin}$ which swaps a and b.

Lemma 2.22. Fin *is not generated as a monoid (under functional composition) by atomic renamings. In other words, atomic renamings are not enough to generate every $\sigma \in \mathsf{Fin}$.*

Proof. Atomic renamings are not injective. A functional composition of non-injective functions is also non-injective. The swapping $(a\ b)$ is injective, so no functional composition of atomic renamings can generate a swapping. □

In spite of Lemma 2.22 atomic renamings *are* enough to construct **Sub**, in a suitable formal sense:

Theorem 2.23. *Suppose that $\mathbb{X} = (|\mathbb{X}|, \cdot)$ and $\mathbb{X}' = (|\mathbb{X}'|, \cdot')$ are nominal renaming sets with the same underlying set (so $|\mathbb{X}| = |\mathbb{X}'|$). Suppose that for all atomic renamings $[a{\mapsto}b]$ and all $x \in |\mathbb{X}|$ it is the case that*

$$[a{\mapsto}b] \cdot x = [a{\mapsto}b] \cdot' x.$$

Then for all $\sigma \in \mathsf{Fin}$ and all $x \in |\mathbb{X}|$ it is the case that

$$\sigma \cdot x = \sigma \cdot' x,$$

and so $\mathbb{X} = \mathbb{X}'$.

In words:

> The renaming action is determined by the atomic renaming action.

In view of Lemma 2.22 the issue is that not every finitely-supported renaming can be expressed as a chain of atomic renamings. For comparison, it *is* the case that every finitely-supported permutation can be expressed as a chain of atomic permutations, so Theorem 2.23 is simply not an issue in nominal permutative techniques.

Proof. Suppose that $[a{\mapsto}b] \cdot x = [a{\mapsto}b] \cdot' x$ always. Suppose that $x \in |\mathbb{X}|$ and choose any $\sigma \in \mathsf{Fin}$. We wish to show that $\sigma \cdot x = \sigma \cdot' x$.

Now suppose that $x \in |\mathbb{X}|$ and $\sigma \in \mathsf{Fin}$. Write $\mathit{supp}(x) = \{c_1, \ldots, c_k\}$. Choose some fresh $\{c'_1, \ldots, c'_k\}$ (so $c'_i \# x$ and $\sigma(c'_i) = c'_i$ for $1 \leq i \leq k$). Define

$$\tau = [c_1{\mapsto}c'_1, \ldots, c_k{\mapsto}c'_k] \quad \text{and} \quad \tau' = [c'_1{\mapsto}\sigma(c_1), \cdots, c_k{\mapsto}\sigma(c_k)].$$

Note that

$$\tau = [c_1{\mapsto}c'_1] \circ \cdots \circ [c_k{\mapsto}c'_k] \quad \text{and} \quad \tau' = [c'_1{\mapsto}\sigma(c_1)] \circ \cdots \circ [c'_k{\mapsto}\sigma(c_k)].$$

That is, both τ and τ' can be expressed as functional compositions of atomic renamings. Also, by Theorem 2.13

$$(\tau' \circ \tau) \cdot x = \sigma \cdot x.$$

The result follows. □

Atomic renamings have the benefits of simplicity and concreteness. We now prove Corollary 2.24 and Lemma 2.25. These use atomic renamings, and they are useful for concrete calculations we shall perform later.

Corollary 2.24 is a corollary of part 1 of Lemma 2.16. It gives a concrete way to calculate support. We will find it useful for Lemmas 4.7 and 5.3.

Corollary 2.24. *Suppose that $\mathbb{X}$ is a nominal renaming set and suppose that $x \in |\mathbb{X}|$. If $a \in \mathit{supp}(x)$ and $c \notin \mathit{supp}(x)$ then $[a{\mapsto}c] \cdot x \neq x$.*

Taking the contrapositive, if there exists some $c\#x$ such that $[a{\mapsto}c] \cdot x = x$, then $a\#x$.

Proof. By part 1 of Lemma 2.16

$$\mathit{supp}([a{\mapsto}c] \cdot x) \;\subseteq\; [a{\mapsto}c] \cdot \mathit{supp}(x) \;=\; (\mathit{supp}(x) \setminus \{a\}) \cup \{c\}.$$

In particular,

$$a \notin (\mathit{supp}[a{\mapsto}c] \cdot x).$$

Since we assumed that $a \in \mathit{supp}(x)$, the result follows. □

Lemma 2.25 is useful for proving Lemma 5.6 and Corollary 5.2:

Lemma 2.25. *Suppose that $\mathbb{X}$ is a nominal renaming set, $x, x' \in |\mathbb{X}|$, and $a, a' \in \mathbb{A}$. Suppose that $c\#x$ and $c\#x'$.*

Then $[a{\mapsto}c] \cdot x = [a'{\mapsto}c] \cdot x'$ if and only if $a'\#x$ and $x' = [a{\mapsto}a'] \cdot x$.

Proof. Note that by our permutative convention, c, a, and a' are all distinct.

Suppose that $a'\#x$ and $x' = [a{\mapsto}a'] \cdot x$. Then we reason as follows:

$$\begin{array}{lll} [a'{\mapsto}c] \cdot x' & = [a'{\mapsto}c] \cdot [a{\mapsto}a'] \cdot x & x' = [a{\mapsto}a'] \cdot x \\ & = [a{\mapsto}c] \cdot x & \text{Theorem 2.13, } a'\#x \end{array}$$

Conversely suppose that $[a{\mapsto}c] \cdot x = [a'{\mapsto}c] \cdot x'$. Then

$$[c{\mapsto}a'] \cdot [a{\mapsto}c] \cdot x = [c{\mapsto}a'] \cdot [a'{\mapsto}c] \cdot x'.$$

By Theorem 2.13

$$[c{\mapsto}a'] \cdot [a'{\mapsto}c] \cdot x' = x' \quad \text{and} \quad [c{\mapsto}a'] \cdot [a{\mapsto}c] \cdot x = [a{\mapsto}a'] \cdot x.$$

It follows that $x' = [a{\mapsto}a'] \cdot x$.

By similar reasoning we have that $x = [a'{\mapsto}a] \cdot x'$. By Lemma 2.16 it then follows that $a'\#x$. □

3 Nominal renaming sets and the exponential

Recall from Subsection 2.1 that $|\mathbb{X}| \to |\mathbb{Y}|$ is the set of functions from the set $|\mathbb{X}|$ to the set $|\mathbb{Y}|$.

Definition 3.1. *Suppose that $\mathbb{X}$ and $\mathbb{Y}$ are nominal renaming sets. Let $|\mathbb{X} \Rightarrow \mathbb{Y}|$ be the set of functions $f \in |\mathbb{X}| \to |\mathbb{Y}|$ such that there exists some finite $S_f \subseteq \mathbb{A}$ (for each f, we fix one such S_f) such that for all $\sigma \in \mathsf{Fin}$ and $x \in |\mathbb{X}|$ if $\sigma|_{S_f} = id|_{S_f}$ then*

$$\sigma \cdot f(x) = f(\sigma \cdot x). \tag{1}$$

In Definition 3.6 $|\mathbb{X} \Rightarrow \mathbb{Y}|$ will serve as the underlying set of a nominal renaming set $\mathbb{X} \Rightarrow \mathbb{Y}$. In Theorem 7.6 we will prove that $\mathbb{X} \Rightarrow \mathbb{Y}$ is an exponential in a suitable cateogory of nominal renaming sets. First, we consider some examples and some properties of $|\mathbb{X} \Rightarrow \mathbb{Y}|$.

Remark 3.2. Recall from Subsection 2.3 that $\mathbb{B}$ has underlying set $\{\bot, \top\}$ and trivial renaming action so that $\sigma \cdot \bot = \bot$ and $\sigma \cdot \top = \top$ always.

For example:

1. Let $\mathbb{X}$ and $\mathbb{Y}$ be nominal renaming sets. The map $\pi_1 \in |\mathbb{X} \times \mathbb{Y}| \to |\mathbb{X}|$ mapping (x, y) to x is in $|(\mathbb{X} \times \mathbb{Y}) \Rightarrow \mathbb{X}|$.

2. The map $=\in |\mathbb{A} \times \mathbb{A}| \to |\mathbb{B}|$ mapping (x, y) to $\top$ if $x = y$ and mapping (x, y) to $\bot$ if $x \neq y$, is *not* an element of $|(\mathbb{A} \times \mathbb{A}) \Rightarrow \mathbb{B}|$. This is because there is no finite set S such that if $\sigma|_S = id|_S$ then for all $x, y \in \mathbb{A}$ it is the case that $x = y$ if and only if $\sigma(x) = \sigma(y)$.

3. Recall from Definition 2.15 that PFin is the set of finite sets of atoms with pointwise renaming action. Suppose that $\mathbb{X}$ is any nominal renaming set.

   The map $supp_{\mathbb{X}} \in |\mathbb{X}| \to |\mathsf{PFin}|$ mapping x to $supp(x)$, is sometimes an element of $|\mathbb{X} \Rightarrow \mathsf{PFin}|$ — and sometimes not.

   In the case that $\mathbb{X} = \mathbb{A}$ then $supp(a) = \{a\}$ and

   $$\sigma \cdot supp(a) = \sigma \cdot \{a\} = \{\sigma(a)\} = supp(\sigma(a)).$$

   Therefore $supp_{\mathbb{A}} \in |\mathbb{A} \Rightarrow \mathsf{PFin}|$. Similarly for $supp_{\mathsf{PFin}}$ and for many other examples.

   In the case that $\mathbb{X}$ is an 'exploding model', as discussed in Lemma 2.18, it is not necessarily the case that $supp_{\mathbb{X}} \in |\mathbb{X} \Rightarrow \mathsf{PFin}|$. Taking the example from the proof of Lemma 2.18 we note that for any finite $S \subseteq \mathbb{A}$ we can take $a, b \notin S$ and

$$[a{\mapsto}b]{\cdot}supp(\{a, b\}) = \{b\} \quad \text{whereas} \quad supp([a{\mapsto}b]{\cdot_{exploding}}\{a, b\}) = supp(\emptyset) = \emptyset.$$

Remark 3.3. Intuitively, a map that does not compare atoms in its argument for inequality will be in the underlying set of the exponential. A map that compares atoms for inequality, will not. See the Conclusions for further discussion.

A key technical lemma is that elements of $|\mathbb{X} \Rightarrow \mathbb{Y}|$ are determined only by their 'asymptotic behaviour', in the following sense:

Lemma 3.4. *Suppose that $f \in |\mathbb{X} \Rightarrow \mathbb{Y}|$ and $g \in |\mathbb{X} \Rightarrow \mathbb{Y}|$. Suppose that $S \subseteq \mathbb{A}$ is finite and assume that for all $x \in |\mathbb{X}|$ if $supp(x) \cap S = \emptyset$ ('asymptotic' is in the sense of '$supp(x) \cap S = \emptyset$') then $f(x) = g(x)$. Then $f = g$.*

Proof. Without loss of generality (extending S with $S_f \cup S_g$ if necessary) we may suppose that if $supp(x) \cap S = \emptyset$ then

$$\sigma \cdot f(x) = f(\sigma \cdot x) \quad \text{and} \quad \sigma \cdot g(x) = g(\sigma \cdot x).$$

Now choose any $x \in |\mathbb{X}|$. There are two cases:

- If $supp(x) \cap S = \emptyset$ then by assumption $f(x) = g(x)$.
- If $supp(x) \cap S \neq \emptyset$ then let $C = \{c_1, \ldots, c_k\}$ be a fresh choice of atoms (so $C \cap S = \emptyset$). Let

  $$\tau = [a_1 \mapsto c_1, \ldots, a_k \mapsto c_k] \quad \text{and} \quad \tau^{-1} = [c_1 \mapsto a_1, \ldots, c_k \mapsto a_k].$$

  By Lemma 2.16 we know that $supp(\tau \cdot x) \cap S = \emptyset$. By Theorem 2.13 we know that $x = \tau^{-1} \cdot \tau \cdot x$. Therefore we reason as follows:

  $$\begin{array}{ll} f(x) = f(\tau^{-1} \cdot \tau \cdot x) & \text{Theorem 2.13} \\ = \tau^{-1} \cdot f(\tau \cdot x) & \text{(1), Definition 3.1} \\ = \tau^{-1} \cdot g(\tau \cdot x) & \text{Assumption} \\ = g(\tau^{-1} \cdot \tau \cdot x) & \text{(1), Definition 3.1} \\ = g(x) & \text{Theorem 2.13} \end{array}$$

The result follows. □

Following on from Lemma 3.4, a total function can be uniquely reconstructed from its 'asymptotic' behaviour:

Lemma 3.5. *Suppose that $f \in |\mathbb{X}| \rightharpoonup |\mathbb{Y}|$ is a partial function from $|\mathbb{X}|$ to $|\mathbb{Y}|$. Suppose also that there exists some finite $S \subseteq \mathbb{A}$ such that:*

- *For all $x \in |\mathbb{X}|$ if $supp(x) \cap S = \emptyset$ then $f(x)$ is defined.*
- *For all σ such that $\sigma|_S = id|_S$, if $f(x)$ and $f(\sigma \cdot x)$ are both defined then $\sigma \cdot f(x) = f(\sigma \cdot x)$.*

Then there exists a unique total function $f' \in |\mathbb{X} \Rightarrow \mathbb{Y}|$ extending the partial function f, in the sense that $f'(x) = f(x)$ if $f(x)$ is defined.

Proof. The proof splits into three parts:

1. The statement of the definition of f'.
2. A proof that the statement is well-defined.
3. A proof that f' is unique.

We consider each part in turn:

1. *Statement of the definition of f'.*

   Consider any $x \in |\mathbb{X}|$. Suppose $\mathit{supp}(x) = \{a_1, \ldots, a_k\}$. There are two cases:

   - If $\mathit{supp}(x) \cap S = \emptyset$ then we set
   $$f'(x) = f(x).$$
   - If $\mathit{supp}(x) \cap S \neq \emptyset$ then let $C = \{c_1, \ldots, c_k\}$ be a choice of fresh atoms (so $C \cap S = \emptyset$ and $C \cap \mathit{supp}(x) = \emptyset$). Let
   $$\tau = [a_1 \mapsto c_1, \ldots, a_k \mapsto c_k] \quad \text{and} \quad \tau^{-1} = [c_1 \mapsto a_1, \ldots, c_k \mapsto a_k].$$
   We set
   $$f'(x) = \tau^{-1} \cdot f(\tau \cdot x).$$

2. *Proof that the statement is well-defined.*

   We must show that the choice of fresh C does not matter. Suppose that $C' = \{c'_1, \ldots, c'_k\}$ is some other choice of fresh atoms (so $C' \cap S = \emptyset$ and $C' \cap \mathit{supp}(x) = \emptyset$). Write
   $$\tau' = [a_1 \mapsto c'_1, \ldots, a_k \mapsto c'_k] \quad \text{and} \quad \tau'^{-1} = [c'_1 \mapsto a_1, \ldots, c'_k \mapsto a_k].$$
   We must show that
   $$\tau^{-1} \cdot f(\tau \cdot x) = \tau'^{-1} \cdot f(\tau' \cdot x)$$
   Without loss of generality we assume the special case that
   $$C \cap C' = \emptyset \quad \text{and} \quad C' \cap \mathit{supp}(f(\tau \cdot x)) = \emptyset.$$
   The general case then follows by two applications of the special case for an 'even fresher' set of fresh atoms C''.

   We write
   $$\mu = [c_1 \mapsto c'_1, \ldots, c_k \mapsto c'_k] \quad \text{and} \quad \mu^{-1} = [c'_1 \mapsto c_1, \ldots, c'_k \mapsto c_k].$$
   By Theorem 2.13 and our assumption that $C' \cap \mathit{supp}(x) = \emptyset$, we know that $\tau' \cdot x = \mu \cdot \tau \cdot x$. Also, by our assumption that $C' \cap S = \emptyset$, we know that $f(\tau' \cdot x) = \mu \cdot f(\tau \cdot x)$. It follows that
   $$\tau'^{-1} \cdot f(\tau' \cdot x) = \tau^{-1} \cdot \mu^{-1} \cdot \mu \cdot f(\tau \cdot x).$$
   By Theorem 2.13 and our assumption that $C' \cap \mathit{supp}(f(\tau \cdot x)) = \emptyset$ it follows that
   $$\mu^{-1} \cdot \mu \cdot f(\tau \cdot x) = f(\tau \cdot x).$$
   The result follows.

3. *Proof that f' is unique.*

   This follows from Lemma 3.4.

□

Definition 3.6. *If $\mathbb{X}$ and $\mathbb{Y}$ are nominal renaming sets let $\mathbb{X} \Rightarrow \mathbb{Y}$ be specified as follows:*

- *$|\mathbb{X} \Rightarrow \mathbb{Y}|$ is defined in Definition 3.1.*
- *We specify a renaming action by*

$$(\sigma \cdot f)x = \sigma \cdot f(x) \tag{2}$$

for $x \in |\mathbb{X}|$ such that $supp(x) \cap S_f = \emptyset$ (S_f fixed in Definition 3.1).

By Lemma 3.5 the renaming action is uniquely specified for all $x \in |\mathbb{X}|$.

Theorem 3.7. *Suppose that $\mathbb{X}$ and $\mathbb{Y}$ are nominal renaming sets, $x \in |\mathbb{X}|$, and $f \in |\mathbb{X} \Rightarrow \mathbb{Y}|$.*
Then $\sigma \cdot f(x) = (\sigma \cdot f)(\sigma \cdot x)$ always.
As an immediate corollary if $\sigma \cdot x = x$ then $(\sigma \cdot f)x = \sigma \cdot f(x)$.

Proof. Write $supp(x) = \{a_1, \ldots, a_k\}$. Make some choice of fresh $C = \{c_1, \ldots, c_k\}$ — so that

- $C \cap supp(x) = \emptyset$.
- $C \cap S_f = \emptyset$.
- $C \cap S_{\sigma \cdot f} = \emptyset$.
- $\sigma(c) = c$ for all $c \in C$.

Let

$$\tau = [a_1 \mapsto c_1, \ldots, a_k \mapsto c_k] \quad \text{and} \quad \tau^{-1} = [c_1 \mapsto a_1, \ldots, c_k \mapsto a_k].$$

Define σ' by

- $\sigma'(c_i) = c_j$ if $1 \leq i \leq k$ and $\sigma(a_i) \neq a_i$ and $\sigma(a_i) = a_j$.
- $\sigma'(y) = y$ otherwise.

We now note three equalities in (3), (4), and (5). Because $C \cap supp(x) = \emptyset$ is easy to verify that

$$(\tau^{-1} \circ \sigma' \circ \tau)|_{supp(x)} = \sigma|_{supp(x)}. \tag{3}$$

Because $\sigma(c) = c$ for all $c \in C$, and by the construction of σ', we have that

$$\tau^{-1} \circ \sigma' \circ \sigma = \sigma \circ \tau^{-1}. \tag{4}$$

By construction

$$(\tau^{-1} \circ \sigma')|_{S_{\sigma \cdot f}} = id|_{S_{\sigma \cdot f}}. \tag{5}$$

Then we reason as follows:

$$\begin{aligned}
(\sigma \cdot f)(\sigma \cdot x) &= (\sigma \cdot f)((\tau^{-1} \circ \sigma') \cdot \tau \cdot x) && \text{Theorem 2.13 and (3)} \\
&= (\tau^{-1} \circ \sigma') \cdot (\sigma \cdot f)(\tau \cdot x) && \text{(5) and (1), Definition 3.1} \\
&= (\tau^{-1} \circ \sigma') \cdot \sigma \cdot f(\tau \cdot x) && \text{(5) and (2), Definition 3.6} \\
&= (\sigma \circ \tau^{-1}) \cdot f(\tau \cdot x) && \text{(4)} \\
&= \sigma \cdot f(\tau^{-1} \cdot \tau \cdot x) && \text{(1), Definition 3.1} \\
&= \sigma \cdot f(x) && \text{Theorem 2.13}
\end{aligned}$$

□

Corollary 3.8. *Suppose that $\mathbb{X}$ and $\mathbb{Y}$ are nominal renaming sets. Suppose that $f \in |\mathbb{X} \Rightarrow \mathbb{Y}|$. Then $S \subseteq \mathbb{A}$ is such that*

$$\textit{for all } x \in |\mathbb{X}| \textit{ if } \sigma|_S = id|_S \textit{ then } \sigma \cdot f(x) = f(\sigma \cdot x)$$

if and only if $supp(f) \subseteq S$.

Proof. Suppose $\sigma|_{supp(f)} = id|_{supp(f)}$. By Theorems 3.7 and 2.13 we have

$$\sigma \cdot f(x) = (\sigma \cdot f)(\sigma \cdot x) = f(\sigma \cdot x)$$

for any $x \in |\mathbb{X}|$.

Conversely suppose that $S \subseteq \mathbb{A}$ is such that for all $x \in |\mathbb{X}|$ if $\sigma|_S = id|_S$ then $\sigma \cdot f(x) = f(\sigma \cdot x)$. If we show that S supports f then by the 'unique least' property of support (Theorem 2.13) we are done. Suppose that $\sigma|_S = id|_S$. By Theorem 3.7

$$\sigma \cdot f(x) = (\sigma \cdot f)(\sigma \cdot x).$$

By assumption

$$\sigma \cdot f(x) = f(\sigma \cdot x).$$

By Theorem 2.13 if $supp(x) \cap S = \emptyset$ then $\sigma \cdot x = x$. So for all x such that $supp(x) \cap S = \emptyset$ we have that

$$(\sigma \cdot f)x = fx.$$

It follows by Lemma 3.4 that $\sigma \cdot f = f$, as required. □

Compare Corollary 3.9 with [14, Example 4.9, (24)].

Corollary 3.9. *Suppose that $\mathbb{X}$ and $\mathbb{Y}$ are nominal renaming sets. Suppose that $x \in |\mathbb{X}|$ and $f \in |\mathbb{X} \Rightarrow \mathbb{Y}|$. Then*

$$supp(f(x)) \subseteq supp(f) \cup supp(x).$$

Proof. Suppose that $\sigma|_{supp(f) \cup supp(x)} = \sigma'|_{supp(f) \cup supp(x)}$. Using Theorem 3.7 and Theorem 2.13 we easily calculate that $\sigma \cdot f(x) = \sigma' \cdot f(x)$. □

4 Nominal permutation sets

In this section we recall definitions and results from nominal techniques based on permutations, which will be useful for Section 5 and later sections.

4.1 Definitions and results, and connection with nominal renaming sets

The following definition is taken from [14]:

Definition 4.1. *Let* Per *be the set of bijective functions from $\mathbb{A}$ to $\mathbb{A}$ (***permutations***) such that there exists some finite $S \subseteq \mathbb{A}$ such that*

$$\textit{for all } b \in \mathbb{A} \setminus S \textit{ it is the case that } \pi(b) = b.$$

π, π' will range over elements of Per*. We call these* **(finitely supported) permutations**.

Lemma 4.2. Per *is a group, with identity id (the identity on atoms) and group composition function composition* $\circ$.

Furthermore, $\mathsf{Per} \subseteq \mathsf{Fin}$ *from Definition 2.4 and* Per *is a submonoid of* Fin.

Proof. Routine. □

Definition 4.3. *A* **nominal permutation set** $\mathbb{P}$ *is a pair* $(|\mathbb{P}|, \cdot)$ *of an* **underlying set** $|\mathbb{P}|$ *and* **finitely-supported permutation action** $\cdot$.

A finitely-supported permutation action $\cdot$ *is a function from* $\mathsf{Per} \times |\mathbb{P}|$ *to* $|\mathbb{P}|$ *such that:*

- $id \cdot x = x$ *for all* $x \in |\mathbb{P}|$.
- $\pi' \cdot (\pi \cdot x) = (\pi' \circ \pi) \cdot x$ *for all* $x \in |\mathbb{X}|$ *and* $\pi', \pi \in \mathsf{Per}$.
  (So $\cdot$ *is a group action.)*
- *For every* $x \in |\mathbb{P}|$ *there exists some finite* $S \subseteq \mathbb{A}$ *such that for all* $\pi \in \mathsf{Per}$ *and* $\pi' \in \mathsf{Per}$*, if* $\pi|_S = \pi'|_S$ *then* $\pi \cdot x = \pi' \cdot x$.
  We say that every $x \in |\mathbb{P}|$ *has* **finite support**.

Definition 4.4. *Suppose that* $\mathbb{P}$ *is a nominal permutation set and* $x \in |\mathbb{P}|$. *Say that* $S \subseteq \mathbb{A}$ **supports** x *when for all* $\pi \in \mathsf{Per}$ *and* $\pi' \in \mathsf{Per}$*, if* $\pi|_S = \pi'|_S$ *then* $\pi \cdot x = \pi' \cdot x$.

Remark 4.5. A nominal permutation set $\mathbb{P}$ is a set with a finitely-supported Per-group action.

Lemma 4.6. *If* $\mathbb{X}$ *is a nominal renaming set then* $(|\mathbb{X}|, \cdot)$ *where* $\cdot$ *is restricted to* Per*, is a nominal permutation set.*

Proof. Routine. □

Lemma 4.7. *supp*(x) *from Theorem 2.13 (the minimal set of atoms that supports* x *as an element of a nominal renaming set) is a subset of* S *from Definition 4.3 (a set of atoms that supports* x *as an element of a nominal permutation set).*

Proof. If $supp(x) = \emptyset$ the result is immediate. Suppose that $a \in supp(x)$ (that is, suppose that $a\#x$ does not hold). Choose fresh c (so $c\#x$ and $c \notin T$).

Recall swappings $(a\ c)$ (Definition 2.5). By Theorem 2.13 $[a{\mapsto}c]\cdot x = (a\ c)\cdot x$. By Corollary 2.24 $[a{\mapsto}c]\cdot x \neq x$. It follows that $(a\ c)\cdot x \neq x$. Therefore $a \in S$. □

Theorem 4.8. *Suppose that* $S \subseteq \mathbb{A}$ *is finite. Suppose that* $|\mathbb{X}|$ *is a nominal renaming set and suppose that* $x \in |\mathbb{X}|$.

Then S *supports* x *in the sense of Definition 4.3 (considering* x *as an element of a nominal permutation set) if and only if* S *supports* x *in the sense of Definition 2.8 (considering* x *as an element of a nominal renaming set).*

Proof. We must show that

$$\text{for all } \pi, \pi' \in \mathsf{Per}, \text{ if } \pi|_S = \pi'|_S \text{ then } \pi \cdot x = \pi' \cdot x,$$

if and only if

$$\text{for all } \sigma, \sigma' \in \mathsf{Fin}, \text{ if } \sigma|_S = \sigma'|_S \text{ then } \sigma \cdot x = \sigma' \cdot x.$$

The bottom-to-top implication is immediate from Lemma 4.2. The top-to-bottom implication is an easy corollary of Lemma 4.7. □

Remark 4.9. An informal corollary of Theorem 4.8, which can be made formal, is this:

If $\mathbb{X}$ is a nominal renaming set and $x \in |\mathbb{X}|$ then the notion of support from Theorem 2.13 coincides with the notion of support obtained by restricting to permutations and using Definition 4.4 (plus known results from nominal permutative techniques, such as [14, Proposition 3.4]).

We shall write '$supp(x)$' and use results about support such as Theorem 2.13 and Lemma 2.16 without concern for whether we consider x as acted on by Fin (all renamings) or we consider x as acted on by Per (only permutations).

4.2 The Gabbay-Pitts (nominal) model of abstraction

Definition 4.10. *Suppose $\mathbb{X}$ is a nominal renaming set and suppose $x \in |\mathbb{X}|$ and $a \in \mathbb{A}$. Define*

$$[a]x = \{(a, x)\} \cup \{(c, [a{\mapsto}c] \cdot x) \mid c\#x\}.$$

Define $|[\mathbb{A}]\mathbb{X}|$ by

$$|[\mathbb{A}]\mathbb{X}| = \{[a]x \mid a \in \mathbb{A},\ x \in |\mathbb{X}|\}.$$

A renaming action will follow in Definition 5.11.

Recall from Definition 2.5 that the swapping $(a\ b)$ in the function on atoms which swaps a and b.

Remark 4.11. By Theorem 2.13 if $c\#x$ then $(a\ c) \cdot x = [c{\mapsto}a] \cdot x$. Therefore, $|[\mathbb{A}]\mathbb{X}|$ in Definition 4.10 is the same set as underlying set of abstraction from [14, Lemma 5.1].

We now recall properties of abstraction which will be useful later. By Theorem 4.8 results about abstractions built in nominal permutation sets can be directly imported to abstractions built using nominal renaming sets; as we have observed, the underlying sets are identical and notions of support are the same.

Lemma 4.12. *Suppose that $\mathbb{X}$ is a nominal renaming set. Suppose that $x, x' \in |\mathbb{X}|$, and suppose that $a, a' \in \mathbb{A}$.*

Then $[a]x = [a']x'$ if and only if $a'\#x$ and $x' = [a{\mapsto}a'] \cdot x$.

Proof. By calculations from [14, Lemma 5.1]. □

Lemma 4.13. *Suppose that $\mathbb{X}$ is a nominal renaming set, suppose $x \in |\mathbb{X}|$, and suppose that $a, b \in \mathbb{A}$ and $b\#x$. Then*

$$[a]x = [b]((b\ a) \cdot x) = [b]([a{\mapsto}b] \cdot x).$$

Proof. By calculations from [14, Lemma 5.1], and by Theorem 2.13. □

Lemma 4.14. *Suppose that $\mathbb{X}$ is a nominal renaming set and suppose $z \in |[\mathbb{A}]\mathbb{X}|$. Then if $(a, x) \in z$ then $z = [a]x$.*

Proof. By calculations from [14, Lemma 5.1]. □

5 The atoms-exponential $\mathbb{A} \Rightarrow \mathbb{X}$ as a nominal-style atoms-abstraction

The results of this subsection demonstrate that if $\mathbb{X}$ is a nominal renaming set then $\mathbb{A} \Rightarrow \mathbb{X}$ and $[\mathbb{A}]\mathbb{X}$ are essentially the same thing.

From the point of view of research into nominal techniques, this is one of the important technical points of this paper because it makes a connection between nominal techniques as we know them from previous work [14, 10], and functional abstraction as supported by — and extensively used in — most major theorem-provers and programming languages [7].

Work based on presheaves also exhibits an abstraction as a function-space [8, Equation (8)]. We discuss this further in the Conclusions.

Lemma 5.1. *For every $f \in |\mathbb{A} \Rightarrow \mathbb{X}|$ there exists some $x \in |\mathbb{X}|$ and $a\#f$ such that $f = \lambda y{\in}\mathbb{A}.\ [a{\mapsto}y] \cdot x$.*

Proof. We unpack Definition 3.6. $f \in |\mathbb{A} \Rightarrow \mathbb{X}|$ when $f \in |\mathbb{A}| \to |\mathbb{X}|$ and there exists some finite $S \subseteq \mathbb{A}$ such that for all $\sigma \in \mathsf{Fin}$ and $a \in \mathbb{A}$, if $\sigma|_S = id|_S$ then $\sigma \cdot f(a) = f(\sigma(a))$.

Choose $a \notin S$ and any $y \in \mathbb{A}$. Then

$$f(y) = f([a{\mapsto}y] \cdot a) = [a{\mapsto}y] \cdot f(a).$$

So take $x = f(a)$ and we are done. □

Corollary 5.2. *Suppose that $\mathbb{X}$ is a nominal renaming set. Suppose that $f \in |\mathbb{A} \Rightarrow \mathbb{X}|$ and $f' \in |\mathbb{A} \Rightarrow \mathbb{X}|$ and suppose that $c\#f$ and $c\#f'$.*

Then $fc = f'c$ if and only if $f = f'$.

Proof. The right-to-left implication is immediate. So assume that $fc = f'c$. By Lemma 5.1

- there exists some $x \in |\mathbb{X}|$ and $a \in \mathbb{A}$ such that $a\#f$ and $f = \lambda y{\in}\mathbb{A}.\ [a{\mapsto}y] \cdot x$, and
- there exists some $x' \in |\mathbb{X}|$ and $a' \in \mathbb{A}$ such that $a'\#f'$ and $f' = \lambda y{\in}\mathbb{A}.\ [a'{\mapsto}y] \cdot x'$.

We assumed that $fc = f'c$ so $[a{\mapsto}c] \cdot x = [a'{\mapsto}c] \cdot x'$. By Lemma 2.25

$$a'\#x \quad \text{and} \quad x' = [a{\mapsto}a'] \cdot x.$$

Choose any $y \in \mathbb{A}$. We reason as follows:

$$\begin{array}{lll} f'y & = [a'{\mapsto}y] \cdot x' & f' = \lambda y{\in}\mathbb{A}.\ [a'{\mapsto}y] \cdot x' \\ & = [a'{\mapsto}y] \cdot [a{\mapsto}a'] \cdot x. & x' = [a{\mapsto}a'] \cdot x \\ & = [a{\mapsto}y] \cdot x & \text{Theorem 2.13, } a'\#x \end{array}$$

□

Lemma 5.3. *Suppose that $f \in |\mathbb{X} \Rightarrow \mathbb{Y}|$.*

Then $a\#f$ if and only if $a\#fb$, for any $b\#f$ (by our permutative convention, $b \neq a$).

Equivalently, $supp(f) = supp(fb) \setminus \{b\}$ for any $b\#f$.

Proof. We prove two implications. Choose any $b\#f$.

If $a\#f$ then by Corollary 3.9 $a\#fb$ and we are done. Suppose that $a\#fb$. We have assumed $b\#f$ so by Corollary 2.24 it suffices to prove $[a\mapsto b]\cdot f = f$. Choose a fresh c (so $c\#f$ and $c\#[a\mapsto b]\cdot f$). By Corollary 5.2 it suffices to check that $fc = ([a\mapsto b]\cdot f)(c)$. Note that by Corollary 3.9 $a\#fc$. We reason as follows:

$$\begin{array}{ll} fc = [a\mapsto b]\cdot(fc) & \text{Theorem 2.13 and Corollary 3.9} \\ \quad = ([a\mapsto b]\cdot f)([a\mapsto b]\cdot c) & \text{Theorem 3.7} \\ \quad = ([a\mapsto b]\cdot f)c & [a\mapsto b]\cdot c = c \end{array}$$

□

Lemma 5.4. *Suppose that $\mathbb{X}$ is a nominal renaming set and suppose that $x \in |\mathbb{X}|$ and $a \in \mathbb{A}$.*

Then $\lambda y{\in}\mathbb{A}.\ [a\mapsto y]\cdot x$ is supported by $supp(x)$.

As a corollary, $\lambda y{\in}\mathbb{A}.\ [a\mapsto y]\cdot x \in |\mathbb{A}\Rightarrow\mathbb{X}|$.

Proof. Unpacking Definition 3.1, $\lambda y{\in}\mathbb{A}.\ [a\mapsto y]\cdot x \in |\mathbb{A}\Rightarrow\mathbb{X}|$ when

- $\lambda y{\in}\mathbb{A}.\ [a\mapsto y]\cdot x \in \mathbb{A}\to|\mathbb{X}|$ — this is true by construction — and
- $\lambda y{\in}\mathbb{A}.\ [a\mapsto y]\cdot x$ has a finite supporting set.

Therefore, the corollary is immediate given the first part. We now prove that $supp(x)$ supports $\lambda y{\in}\mathbb{A}.\ [a\mapsto y]\cdot x$.

By Corollary 3.8 it suffices to show that for all $\sigma \in \mathsf{Fin}$

$$\sigma|_{supp(x)} = id|_{supp(x)} \quad \text{implies} \quad \sigma\cdot\lambda y{\in}\mathbb{A}.\ [a\mapsto y]\cdot x = \lambda y{\in}\mathbb{A}.\ [a\mapsto y]\cdot x.$$

So suppose $\sigma|_{supp(x)} = id|_{supp(x)}$. By Corollary 5.2 it suffices to check

$$(\sigma\cdot\lambda y{\in}\mathbb{A}.\ [a\mapsto y]\cdot x)c = (\lambda y{\in}\mathbb{A}.\ [a\mapsto y]\cdot x)c$$

for fresh c. So choose c such that

$$c\#\sigma\cdot\lambda y{\in}\mathbb{A}.\ [a\mapsto y]\cdot x, \quad c\#\lambda y{\in}\mathbb{A}.\ [a\mapsto y]\cdot x, \quad \sigma(c) = c, \quad \text{and} \quad c\#x.$$

We assumed that $\sigma(c) = c$ so using Theorem 3.7 it suffices to check that

$$\sigma\cdot[a\mapsto c]\cdot x = [a\mapsto c]\cdot x.$$

Now we assumed that $\sigma|_{supp(x)} = id|_{supp(x)}$ and $c \notin supp(x)$. It follows that $(\sigma\circ[a\mapsto c])|_{supp(x)} = [a\mapsto c]|_{supp(x)}$. By Theorem 2.13 the result follows. □

Corollary 5.5. *Suppose that $\mathbb{X}$ is a nominal renaming set and suppose that $x \in |\mathbb{X}|$. Then $supp(\lambda y{\in}\mathbb{A}.\ [a\mapsto y]\cdot x) = supp(x)\setminus\{a\}$.*

Proof. Choose some fresh b (so $b\#\lambda y{\in}\mathbb{A}.\ [a\mapsto y]\cdot x$ and $b\#x$). By Lemma 5.4 $\lambda y{\in}\mathbb{A}.\ [a\mapsto y]\cdot x \in |\mathbb{A}\Rightarrow\mathbb{X}|$. Therefore by Lemma 5.3 we know that

$$supp(\lambda y{\in}\mathbb{A}.\ [a\mapsto y]\cdot x) = supp([a\mapsto b]\cdot x)\setminus\{b\}.$$

Since $b\#x$ by Theorem 2.13 $[a\mapsto b]\cdot x = (b\ a)\cdot x$. By part 2 of Lemma 2.16 $supp((a\ b)\cdot x) = (supp(x)\setminus\{a\})\cup\{b\}$. The result follows. □

Lemma 5.6. *Suppose that $\mathbb{X}$ is a nominal renaming set, $x, x' \in |\mathbb{X}|$, and $a, a' \in \mathbb{A}$.*

Then $\lambda y{\in}\mathbb{A}.\ [a{\mapsto}y] \cdot x = \lambda y{\in}\mathbb{A}.\ [a'{\mapsto}y] \cdot x'$ if and only if $a'\#x$ and $x' = [a{\mapsto}a'] \cdot x$.

(Here $y \in \mathbb{A}$ and y may be equal to a or a'. By our permutative convention $a \neq a'$.)

Proof. Choose some fresh c (so $c\#x$, $c\#x'$, and $c \neq a$ and $c \neq a'$).

If $\lambda y{\in}\mathbb{A}.\ [a{\mapsto}y] \cdot x = \lambda y{\in}\mathbb{A}.\ [a'{\mapsto}y] \cdot x'$ then $[a{\mapsto}c] \cdot x = [a'{\mapsto}c] \cdot x'$. By Lemma 2.25 $a'\#x$ and $x' = [a{\mapsto}a'] \cdot x$ and we are done.

Conversely if $a'\#x$ and $x' = [a{\mapsto}a'] \cdot x$ then by Lemma 2.25 $[a{\mapsto}c] \cdot x = [a'{\mapsto}c] \cdot x'$. Now by Theorem 2.13

$$[c{\mapsto}y] \cdot [a{\mapsto}c] \cdot x = [a{\mapsto}y] \cdot x \quad \text{and} \quad [c{\mapsto}y] \cdot [a'{\mapsto}c] \cdot x' = [a'{\mapsto}y] \cdot x',$$

for any $y \in \mathbb{A}$. The result follows. □

If $\mathbb{X}$ is a nominal renaming set then nominal abstractions $[\mathbb{A}]\mathbb{X}$ (Definition 4.10) and function-spaces $\mathbb{A} \Rightarrow \mathbb{X}$ are related in a very strict sense:

Theorem 5.7. *$|\mathbb{A} \Rightarrow \mathbb{X}|$ is in bijection with $|[\mathbb{A}]\mathbb{X}|$. The mutually inverse mappings are given by:*

- *α maps $z \in |[\mathbb{A}]\mathbb{X}|$ to $\lambda y{\in}\mathbb{A}.\ [a{\mapsto}y] \cdot x$, for $(a, x) \in z$.*
- *β maps $f \in |\mathbb{A} \Rightarrow \mathbb{X}|$ to $[a](fa)$, for $a\#f$.*

Proof. Given our results so far, proving this is not hard. We show that α and β are well-defined:

- *α is well-defined.*

  Suppose that $(a, x) \in z$ and $(a', x') \in z$ (recall that by our permutative convention $a \neq a'$). By Lemma 4.12 $a'\#x$ and $x' = [a{\mapsto}a'] \cdot x$. By Lemma 5.6 $\lambda y{\in}\mathbb{A}.\ [a{\mapsto}y] \cdot x = \lambda y{\in}\mathbb{A}.\ [a'{\mapsto}y] \cdot x'$.

- *β is well-defined.*

  Suppose that $a\#f$ and $b\#f$. By Lemma 4.13 $[a](fa) = [b]([a{\mapsto}b] \cdot (fa))$. Since $a\#f$ by (1) from Definition 3.1

  $$[a{\mapsto}b] \cdot (fa) = f([a{\mapsto}b] \cdot a).$$

  The result follows.

We show that α is injective and surjective:[4]

- *α is surjective.*

  Suppose $f \in |\mathbb{A} \Rightarrow \mathbb{X}|$. By Lemma 5.1 there exists some $x \in |\mathbb{X}|$ and $a\#f$ such that $f = \lambda y{\in}\mathbb{A}.\ [a{\mapsto}y] \cdot x$. By construction $(a, x) \in [a]x$ so by construction $\alpha([a]x) = f$.

[4] This follows also from the next step, in which we prove that α and β are mutually inverse. It is nevertheless instructive to spell out the calculations for proving injectivity and surjectivity.

- *α is injective.*

  Suppose that $(a, x) \in z$ and $(a', x') \in z'$ and $\lambda y{\in}\mathbb{A}.\ [a{\mapsto}y] \cdot x = \lambda y{\in}\mathbb{A}.\ [a'{\mapsto}y] \cdot x'$. By Lemma 5.6 $a'\#x$ and $x' = [a{\mapsto}a'] \cdot x$. By Lemma 4.12 it follows that $[a]x = [a']x'$ as required.

Finally, we prove that α and β are mutually inverse:

- *α followed by β is the identity.*

  Suppose that $z \in |[\mathbb{A}]\mathbb{X}|$ and $(a, x) \in z$. Then $\alpha(z) = \lambda y{\in}\mathbb{A}.\ [a{\mapsto}y] \cdot x$. By Corollary 5.5 $a\#\alpha(z)$, so

  $$\beta(\alpha(z)) = [a]([a{\mapsto}a] \cdot x) = [a]x.$$

  By Lemma 4.14 we are done.

- *β followed by α is the identity.*

  Suppose that $f \in |\mathbb{A} \Rightarrow \mathbb{X}|$ and $a\#f$. Then $\beta(f) = [a](fa)$. By construction $(a, fa) \in [a](fa)$ so that

  $$\alpha(\beta(f)) = \lambda y{\in}\mathbb{A}.\ [a{\mapsto}y] \cdot (fa).$$

  By (1) from Definition 3.1

  $$[a{\mapsto}y] \cdot (fa) = f([a{\mapsto}y] \cdot a) = f(y).$$

  The result follows.

□

Remark 5.8. From previous work [14, 10] we know that if $\mathbb{P}$ is a nominal permutation set then $|[\mathbb{A}]\mathbb{P}|$ has a permutation action. It is given by $\pi \cdot [a]x = [\pi(a)]\pi \cdot x$. Informally, we can say:

> A nominal permutation action on $|\mathbb{P}|$ induces one on $|[\mathbb{A}]\mathbb{P}|$ in a natural way.

$\mathbb{A} \Rightarrow \mathbb{X}$ is a nominal renaming set with action defined in Definition 3.6. By unpacking the maps α and β from the statement of Theorem 5.7 we can give $|[\mathbb{A}]\mathbb{X}|$ a nominal renaming set action. Informally, we can say:

> A nominal renaming action on $\mathbb{X}$ induces one on $|[\mathbb{A}]\mathbb{X}|$ in a natural way.

We need a technical definition:

Definition 5.9. *Write $a\#\sigma$ when*

- *$\sigma(a) = a$ and*
- *for all $y \in \mathbb{A} \setminus \{a\}$ it is the case that $\sigma(y) \neq a$.*

Remark 5.10. $a\#\sigma$ is merely a convenient notation; we do not give a renaming action to Fin.

Definition 5.11. *If $\mathbb{X}$ is a nominal renaming set let $[\mathbb{A}]\mathbb{X}$ be defined by:*

- *The underlying set of* $[\mathbb{A}]\mathbb{X}$ *is* $|[\mathbb{A}]\mathbb{X}|$ *from Definition 3.1.*
- $\sigma \cdot [a]x = [a]\sigma \cdot x$ *provided that* $a\#\sigma$.

Lemma 5.12. *The renaming action in Definition 5.11 is total and well-defined.*

Proof. We must show that:

- *The renaming action is total.* That is, for every $z \in |[\mathbb{A}]\mathbb{X}|$ there exists some $a \in \mathbb{A}$ and $x \in |\mathbb{X}|$ such that $z = [a]x$ and $a\#\sigma$.
- *The renaming action is well-defined.* If $[a]x = [a']x'$ and $a\#\sigma$ and $a'\#\sigma$ then $[a]\sigma \cdot x = [a']\sigma \cdot x'$.

We consider each point in turn.

- By construction if $z \in |[\mathbb{A}]\mathbb{X}|$ then there exist $a \in \mathbb{A}$ and $x \in |\mathbb{X}|$ such that $z = [a]x$. By Lemma 4.13 we may assume (renaming a to be fresh if necessary) that $a\#\sigma$.
- We reason as follows:

$$[a]\sigma \cdot x = \sigma \cdot [a]x = \sigma \cdot [a']x' = [a']\sigma \cdot x'.$$

□

We need a technical lemma for Theorem 5.14:

Lemma 5.13. *If* $a\#\sigma$ *and* $c\#\sigma$ *then* $[a{\mapsto}c] \circ \sigma = \sigma \circ [a{\mapsto}c]$.

Proof. By unpacking Definition 5.9 and by elementary calculations. □

Theorem 5.14. α *and* β *from Theorem 5.7 are arrows in* **Sub** *(Definition 7.1). That is, suppose that* $\mathbb{X}$ *is a nominal renaming set and suppose that* $z \in |[\mathbb{A}]\mathbb{X}|$. *Then:*

- $\sigma \cdot \alpha(f) = \alpha(\sigma \cdot f)$.
- $\sigma \cdot \beta(z) = \beta(\sigma \cdot z)$.

Furthermore, the renaming action on $|[\mathbb{A}]\mathbb{X}|$ *(Definition 5.11) is that induced by the renaming action on* $|\mathbb{A} \Rightarrow \mathbb{X}|$ *via the isomorphism expressed by* α *and* β. *That is,*

$$\sigma \cdot z = \beta(\sigma \cdot (\alpha(z))).$$

Proof. We shall show that

$$\sigma \cdot \alpha(f) = \alpha(\sigma \cdot f).$$

Choose some fresh a (so $a\#f$ and $a\#\sigma$). Since $a\#\sigma$ by Definition 5.9 $\sigma(a) = a$. Therefore by Theorem 3.7 we have that $(\sigma \cdot f)a = \sigma \cdot (fa)$ and so

$$\alpha(f) = [a](fa) \qquad \alpha(\sigma \cdot f) = [a]\sigma \cdot (fa).$$

By Definition 5.11 $\sigma \cdot [a](fa) = [a]\sigma \cdot (fa)$, and we are done.

We shall show that

$$\sigma \cdot \beta(f) = \beta(\sigma \cdot f).$$

We unpack definitions:

$$\beta([a]x) = \lambda y{\in}\mathbb{A}.\ [a{\mapsto}y] \cdot x \qquad \beta([a]\sigma \cdot x) = \lambda y{\in}\mathbb{A}.\ [a{\mapsto}y] \cdot \sigma \cdot x.$$

We wish to show that

$$\sigma \cdot \lambda y{\in}\mathbb{A}.\ [a{\mapsto}y] \cdot x = \lambda y{\in}\mathbb{A}.\ [a{\mapsto}y] \cdot \sigma \cdot x.$$

By Corollary 5.2 it suffices to check this for a fresh c (so $c\#\sigma \cdot \lambda y{\in}\mathbb{A}.\ [a{\mapsto}y] \cdot x$, $c\#\lambda y{\in}\mathbb{A}.\ [a{\mapsto}y] \cdot \sigma \cdot x$, and $c\#\sigma$). We reason as follows:

$$\begin{array}{lll} (\sigma \cdot \lambda y{\in}\mathbb{A}.\ [a{\mapsto}y] \cdot x)c = \sigma \cdot [a{\mapsto}c] \cdot x & & \text{Theorem 3.7 and } \sigma(c) = c \\ = [a{\mapsto}c] \cdot \sigma \cdot x & & a\#\sigma,\ c\#\sigma, \text{ Lemma 5.13} \\ = (\lambda y{\in}\mathbb{A}.\ [a{\mapsto}y] \cdot \sigma \cdot x)c & & \end{array}$$

We have just proved that $\sigma \cdot \beta(z) = \beta(\sigma \cdot z)$. We apply α to both sides and use Theorem 5.7 and the previous parts of this result to conclude

$$\sigma \cdot z = \beta(\sigma \cdot (\alpha(z)))$$

as required. □

We make one final 'sanity check' calculation about the relationship between $[\mathbb{A}]\mathbb{X}$ and $\mathbb{A} \Rightarrow \mathbb{X}$.

Theorem 5.15. *In previous work [14, 10] $|[\mathbb{A}]\mathbb{X}|$ was endowed with a permutation action specified by*

$$\pi \cdot [a]x = [\pi(a)]\pi \cdot x.$$

The renaming action $\sigma \cdot [a]x$ from Definition 5.11 coincides with this definition when $\sigma \in \mathsf{Per}$.

That is, if the renaming σ happens to be a permutation then the renaming action on abstractions emerging from the results above, coincides with the known permutation action on abstractions.

Proof. According to Definition 5.11, if $\pi \in \mathsf{Per}$ and $a\#\pi$ (Lemma 5.13) then $\pi \cdot [a]x = [a]\pi \cdot x$. Also, it is a fact that if $a\#\pi$ then $\pi(a) = a$. Therefore if $a\#\pi$ then $\pi \cdot [a]x = [\pi(a)]\pi \cdot x$ as required.

So suppose that $a\#\pi$ is false. Choose some fresh b (so $b\#x$ and $b\#\pi$). By Lemma 4.13 $[a]x = [b]([a{\mapsto}b] \cdot x)$. Using Definition 5.11

$$\pi \cdot [a]x = [b](\pi \circ [a{\mapsto}b]) \cdot x.$$

By part 2 of Lemma 2.16 $\pi(a)\#(\pi \circ [a{\mapsto}b]) \cdot x$. Therefore by Lemma 4.13

$$[b](\pi \circ [a{\mapsto}b]) \cdot x = [\pi(a)]([b{\mapsto}\pi(a)] \circ \pi \circ [a{\mapsto}b]) \cdot x.$$

We assumed $b\#x$ so by Theorem 2.13

$$([b{\mapsto}\pi(a)] \circ \pi \circ [a{\mapsto}b]) \cdot x = \pi \cdot x.$$

The result follows. □

6 Inductive datatypes of syntax-with-binding using nominal renaming sets

The first motivation for developing nominal (permutation) techniques was to construct datatypes of abstract syntax with binding.

We now have the results we need to quite easily construct a very similar theory of inductively defined sets of syntax-up-to-binding using $\mathbb{A} \Rightarrow$ -.

That is, if we build abstract syntax in **Sub** using using $\mathbb{A} \Rightarrow$ - to model abstraction (in a kind of 'nominal weak higher-order abstract syntax' style) then we obtain a nominal renaming set with *exactly the same* underlying set as that of the nominal permutation set which we obtain if we build abstract syntax in the category of nominal permutation sets, using $[\mathbb{A}]$-, and which therefore admits 'nominal-style' inductive reasoning.

Clearly, the fact that $|\mathbb{A} \Rightarrow \mathbb{X}|$ equals $|[\mathbb{A}]\mathbb{X}|$ (Theorem 5.7) is a strong hint in this direction.

Definition 6.1. *Suppose that* $\mathbb{X} = (|\mathbb{X}|, \cdot)$ *and* $\mathbb{X} = (|\mathbb{X}'|, \cdot')$ *are nominal renaming sets. Write* $\mathbb{X} \subseteq \mathbb{X}'$ *when:*

- $|\mathbb{X}| \subseteq |\mathbb{X}'|$.
- *For all* $\sigma \in \mathsf{Fin}$ *and* $x \in |\mathbb{X}|$, $\sigma \cdot x = \sigma \cdot' x$.

Lemma 6.2.

- $\mathbb{X} \subseteq \mathbb{X}$.
- *If* $\mathbb{X} \subseteq \mathbb{X}'$ *and* $\mathbb{X}' \subseteq \mathbb{X}''$ *then* $\mathbb{X} \subseteq \mathbb{X}''$.

Proof. Routine. □

Definition 6.3. *A* **countably ascending chain** *of nominal renaming sets is a sequence* $\mathbb{X}_i$ *for* $0 \leq i$ *such that* $\mathbb{X}_i \subseteq \mathbb{X}_{i+1}$ *for* $0 \leq i$.

That is, a countably ascending chain of nominal renaming sets looks like this:

$$\mathbb{X}_0 \subseteq \mathbb{X}_1 \subseteq \mathbb{X}_2 \subseteq \mathbb{X}_3 \subseteq \ldots$$

Write $\bigcup_i \mathbb{X}_i$ *for the nominal renaming set specified by:*

- $|\bigcup_i \mathbb{X}_i| = \bigcup_i |\mathbb{X}_i|$.
- *The renaming action is constructed elementwise.*

  That is, if $\sigma \in \mathsf{Fin}$ *and* $x \in |\bigcup_i \mathbb{X}_i|$ *because* $x \in |\mathbb{X}_i|$ *for some* i*, then* σ *acting on* x *as an element of* $|\bigcup_i \mathbb{X}_i|$ *is equal to* σ *acting on* x *as an element of* $\mathbb{X}_i$*, included as an element of* $|\bigcup_i \mathbb{X}_i|$.

Definition 6.4. *Suppose that* $\mathcal{F}$ *maps nominal renaming sets to nominal renaming sets.*

- *Say that* $\mathcal{F}$ **preserves inclusions** *when* $\mathbb{X} \subseteq \mathbb{Y}$ *implies* $\mathcal{F}\mathbb{X} \subseteq \mathcal{F}\mathbb{Y}$.
- *Say that* $\mathcal{F}$ **preserves unions of countably ascending chains** *when* $\mathcal{F}\bigcup_i \mathbb{X}_i = \bigcup_i \mathcal{F}\mathbb{X}_i$.

Definition 6.5. *As is standard, write* $\mathbb{A} \Rightarrow$ - *for the mapping from nominal renaming sets to nominal renaming sets, which maps* $\mathbb{X}$ *to* $\mathbb{A} \Rightarrow \mathbb{X}$.

Lemma 6.6. $\mathbb{A} \Rightarrow$ - *preserves inclusions and preserves unions of countably ascending chains.*

Proof. $\mathbb{A} \Rightarrow$ - *preserves inclusions.* Suppose that $\mathbb{X} \subseteq \mathbb{X}'$. Unpacking Definition 3.1, $f \in |\mathbb{A} \Rightarrow \mathbb{X}|$ is a function $f \in |\mathbb{A}| \to |\mathbb{X}|$ with finite support. By elementary properties of functions, f is also an element of $|\mathbb{A}| \to |\mathbb{X}'|$ with finite support. Therefore $\mathbb{A} \Rightarrow \mathbb{X} \subseteq \mathbb{A} \Rightarrow \mathbb{X}'$.

$\mathbb{A} \Rightarrow$ - *preserves unions of countably ascending chains.* Suppose that $f \in |\mathbb{A} \Rightarrow \bigcup_i \mathbb{X}_i|$. Then $f \in |\mathbb{A}| \to \bigcup_i |\mathbb{X}_i|$ and f has finite support.

By Lemma 5.1 there exists some $a \in \mathbb{A}$ and $x \in \bigcup_i |\mathbb{X}_i|$ such that $f = \lambda y{\in}\mathbb{A}.\ [a{\mapsto}y] \cdot x$. By construction $x \in \mathbb{X}_i$ for some i such that $0 \leq i$. It follows by Lemma 5.4 that $f \in |\mathbb{A} \Rightarrow \mathbb{X}_i|$. Therefore $f \in \bigcup_i |\mathbb{A} \Rightarrow \mathbb{X}_i|$.

Now suppose that $f \in \bigcup_i |\mathbb{A} \Rightarrow \mathbb{X}_i|$. Then $f \in |\mathbb{A} \Rightarrow \mathbb{X}_i|$. That is, $f \in \mathbb{A} \to |\mathbb{X}_i|$ and f has finite support. Since $|\mathbb{X}_i| \subseteq \bigcup_i |\mathbb{X}_i| = |\bigcup_i \mathbb{X}_i|$ the result follows. □

Definition 6.7. *Suppose that $\mathcal{F}$ maps nominal renaming sets to nominal renaming sets.*

- *A* **fixedpoint** *of $\mathcal{F}$ is a nominal renaming set $\mathbb{X}$ such that $\mathcal{F}\mathbb{X} = \mathbb{X}$.*
- *A* **least** *fixedpoint of $\mathcal{F}$ is a fixedpoint $\mathbb{X}$ of $\mathcal{F}$ such that in addition, for all $\mathbb{Y}$ if $\mathcal{F}\mathbb{Y} \subseteq \mathbb{Y}$ then $\mathbb{X} \subseteq \mathbb{Y}$.*

Definition 6.8. *Suppose that $\mathcal{F}$ maps nominal renaming sets to nominal renaming sets. Suppose that $\mathbb{X}$ is a nominal renaming set. Define $\mathcal{F}^i\mathbb{X}$ for $0 \leq i$ inductively by:*

- $\mathcal{F}^0\mathbb{X} = \mathbb{X}$.
- $\mathcal{F}^{i+1}\mathbb{X} = \mathcal{F}(\mathcal{F}^i\mathbb{X})$.

That is, $\mathcal{F}^i\mathbb{X}$ is '$\mathcal{F}$ applied i times to $\mathbb{X}$'.

Lemma 6.9. *If a map $\mathcal{F}$ from nominal renaming sets to nominal renaming sets preserves inclusions and preserves unions of countably ascending chains, then it has a least fixedpoint $\mu(\mathcal{F})$.*

Proof. We take $\mu(\mathcal{F}) = \bigcup_i \mathcal{F}^i\emptyset$. The proof that this *is* a least fixedpoint of $\mathcal{F}$ is well-known and due to Tarski [26]. □

Definition 6.10. *Suppose that $\mathbb{X}$ and $\mathbb{Y}$ are nominal renaming sets. Let $\mathbb{X} \times \mathbb{Y}$ be the nominal renaming set with*

- *underlying set $|\mathbb{X} \times \mathbb{Y}| = |\mathbb{X}| \times |\mathbb{Y}|$ (that is, $z \in |\mathbb{X} \times \mathbb{Y}|$ is a pair (x, y) where $x \in \mathbb{X}$ and $y \in \mathbb{Y}$) and*
- *pointwise renaming action (that is, $\sigma \cdot (x, y) = (\sigma \cdot x, \sigma \cdot y)$).*

Call this the **cartesian product** *of $\mathbb{X}$ and $\mathbb{Y}$.*[5]

Let $\mathbb{X} + \mathbb{Y}$ be the nominal renaming set with

[5]This is nothing more than a special case of Definition 2.21. However, it is a *very useful* special case.

- *underlying set* $|\mathbb{X}|+|\mathbb{Y}|$ *(that is,* $z \in |\mathbb{X}+\mathbb{Y}|$ *is either* $\mathsf{inl}(x)$ *where* $x \in |\mathbb{X}|$ *or* $\mathsf{inr}(y)$ *where* $y \in |\mathbb{Y}|$*) and*
- *pointwise renaming action (that is,* $\sigma \cdot \mathsf{inl}(x) = \sigma \cdot \mathsf{inl}(x)$ *and* $\sigma \cdot \mathsf{inr}(y) = \mathsf{inr}(\sigma \cdot y)$*.*

Call this the **disjoint sum** *of* $\mathbb{X}$ *and* $\mathbb{Y}$*.*

Theorem 6.11. *The map* $\mathcal{F}$ *from nominal renaming sets to nominal renaming sets specified by*

$$\mathcal{F}\mathbb{X} = \mathbb{A} + \mathbb{X} \times \mathbb{X} + \mathbb{A} \Rightarrow \mathbb{X}$$

has a least fixedpoint, write it Λ*.* $|\Lambda|$ *can be identified with untyped* λ*-terms.*

Proof. It routine to prove that maps built up using product $\text{-}\times\text{-}$ and disjoint sum $\text{-}+\text{-}$ preserve inclusions and preserve unions of countably ascending chains. By Lemma 6.6 we can also use $\mathbb{A} \Rightarrow \text{-}$. $|\mathbb{A} \Rightarrow \mathbb{X}|$ coincides with $|[\mathbb{A}]\mathbb{X}|$ (Theorem 5.7). $[a]x$ can be viewed as an α-equivalence class and from previous work [14, 10, 17] (or by a long and detailed, but routine calculation [17]) we can verify that $|\Lambda|$ is indeed λ-terms up to α-equivalence. □

For the rest of this section we informally discuss the Gabbay-Pitts $\mathsf{И}$-quantifier and inductive reasoning principles for Λ.

Suppose that $\phi(y)$ is a predicate on atoms (we discuss what that means, below). Write $\mathsf{И}y.\ \phi(y)$ for the predicate which is:

- True if the set $\{y \in \mathbb{A} \mid \phi(y) \text{ is false}\}$ is finite.
- False otherwise.

For example if $a \in \mathbb{A}$ then:

- $\mathsf{И}y.\ (a = y)$ is false, because only finitely many atoms are equal to a.
- $\mathsf{И}y.\ (a \neq y)$ is true, because infinitely many atoms are not equal to a.

This is the Gabbay-Pitts $\mathsf{И}$ quantifier [14].

The following inductive reasoning principle is from [14]:

$$\begin{array}{ll} & \Big(\forall a \in \mathbb{A}.\ \phi(var(a)) \\ \wedge & \quad \forall x, y \in \Lambda.\ \phi(x) \wedge \phi(y) \Rightarrow \phi(app(x,y)) \\ \wedge & \quad \forall f \in \mathbb{A} \Rightarrow \Lambda.\ (\mathsf{И}y.\ \phi(fy)) \Rightarrow \phi(lam(f))\Big) \quad \Rightarrow \forall x \in \Lambda.\ \phi(x) \end{array}$$

Every nominal renaming set can be considered as a nominal permutation set. This inductive principle is as true of Λ viewed as a nominal renaming set, as it is of Λ viewed as a nominal permutation set in [14].

But for which inductive hypotheses ϕ? We shall see that the category of nominal renaming sets (see Definition 7.1 in just a moment) is a topos, but it is not a boolean topos. It is therefore important to distinguish between 'internal' and 'external' inductive hypotheses.

External. We can admit the predicates of Fraenkel-Mostowski sets, as in [14]; this includes everything that occurs in reasonable practice. Constructing the language of Fraenkel-Mostowski sets and proving the validity of the inductive

principle above is not hard, but it is outside the scope of this document. For more details of Fraenkel-Mostowski sets see [10, 14]. What amounts to nearly the same thing, but presented using the language of categories instead of the language of sets, is that we can use an adjunction between the category **Sub** (see below) and the category of nominal permutation sets from [14], in the style of [9, Section 3].

Internal. Recall Remark 3.3: intuitively, we can admit predicates that do not compare atoms for inequality, but the notion of truth for these predicates is not two-valued. See Subsection 7.4.

7 The category Sub

Definition 7.1. *Nominal renaming sets form a category* **Sub** *as follows:*

- *An object is a nominal renaming set* $\mathbb{X}$*.*
- *An arrow* $F : \mathbb{X} \longrightarrow \mathbb{Y}$ *is a function* $F \in |\mathbb{X}| \to |\mathbb{Y}|$ *such that for all* $\sigma \in \mathsf{Fin}$ *and all* $x \in |\mathbb{X}|$ *it is the case that*

$$\sigma \cdot F(x) = F(\sigma \cdot x). \tag{6}$$

  We let F, G, H *range over arrows.*

7.1 Support and arrows

Lemma 7.2. *Arrows* $F : \mathbb{X} \longrightarrow \mathbb{Y}$ *are precisely the elements* $f \in |\mathbb{X} \Rightarrow \mathbb{Y}|$ *such that* $supp(f) = \emptyset$*.*

Proof. Suppose that $F : \mathbb{X} \longrightarrow \mathbb{Y}$ is an arrow. By Definition 7.1 $F \in |\mathbb{X}| \to |\mathbb{Y}|$. According to Definition 3.1 to show that $F \in |\mathbb{X} \Rightarrow \mathbb{Y}|$ we must exhibit some finite $S \subseteq \mathbb{A}$ such that for all $\sigma \in \mathsf{Fin}$ and $x \in |\mathbb{X}|$ if $\sigma|_S = id|_S$ then $\sigma \cdot F(x) = F(\sigma \cdot x)$. It suffices to take $S = \emptyset$. By Corollary 3.8, $supp(F) = \emptyset$.

Now suppose that $f \in |\mathbb{X} \Rightarrow \mathbb{Y}|$ is such that $supp(f) = \emptyset$. By Definition 3.1 $f \in |\mathbb{X}| \to |\mathbb{Y}|$. According to Definition 7.1 we must show that for all $\sigma \in \mathsf{Fin}$ and $x \in |\mathbb{X}|$ it is the case that $\sigma \cdot f(x) = f(\sigma \cdot x)$. This is immediate from Corollary 3.8. □

Lemma 7.3. *If* $F : \mathbb{X} \longrightarrow \mathbb{Y}$ *is an arrow in* **Sub** *then*

$$supp(F(x)) \subseteq supp(x).$$

Proof. A corollary of Corollaries 3.9 and 3.8. □

7.2 The exponential in Sub

Definition 7.4. *Suppose that* $\mathbb{X}$ *and* $\mathbb{Y}$ *are nominal renaming sets. Let* $\mathsf{app}_{\mathbb{X},\mathbb{Y}}$*, or just* app *for short, be the function from* $|\mathbb{X} \Rightarrow \mathbb{Y}| \times |\mathbb{X}|$ *mapping* $f \in |\mathbb{X} \Rightarrow \mathbb{Y}|$ *and* $x \in |\mathbb{X}|$ *to* $f(x) \in |\mathbb{Y}|$*. We call this* **application***.*

Lemma 7.5. *Suppose that* $\mathbb{X}$ *and* $\mathbb{Y}$ *are nominal renaming sets. Then application* $\mathsf{app} \in (|\mathbb{X} \Rightarrow \mathbb{Y}| \times |\mathbb{X}|) \to |\mathbb{Y}|$ *from Definition 7.4 is an arrow in* **Sub***.*

Proof. Unpacking Definition 6.10, application maps from $|(\mathbb{X} \Rightarrow \mathbb{Y}) \times \mathbb{X}|$ to $|\mathbb{Y}|$. The result follows by Theorem 3.7. □

Theorem 7.6. $\mathbb{X} \Rightarrow \mathbb{Y}$ *is an exponential in* **Sub**.

Proof. Suppose $F \in (\mathbb{X} \times \mathbb{Y}) \longrightarrow \mathbb{Z}$. We must show that

$$\lambda x{\in}|\mathbb{X}|.\ (\lambda y{\in}|\mathbb{Y}|.\ F(x,y)) \in \mathbb{X} \longrightarrow (\mathbb{Y} \Rightarrow \mathbb{Z}).$$

It suffices to show that if $\sigma|_{supp(x)} = id|_{supp(x)}$ then

$$\sigma \cdot F(x,y) = F(x, \sigma \cdot y).$$

This is immediate using (6) and Theorem 2.13.

Now suppose that $G \in \mathbb{X} \longrightarrow (\mathbb{Y} \Rightarrow \mathbb{Z})$. We must show that

$$\lambda(x,y){\in}|\mathbb{X} \times \mathbb{Y}|.\ G(x)(y) \in (\mathbb{X} \times \mathbb{Y}) \longrightarrow \mathbb{Z}.$$

It suffices to show that

$$\sigma \cdot G(x)(y) = G(\sigma \cdot x)(\sigma \cdot y).$$

Now $\sigma \cdot G(x)(y) = (\sigma \cdot G(x))(\sigma \cdot y)$ by Theorem 3.7 and $\sigma \cdot G(x) = G(\sigma \cdot x)$ by (6). The result follows. □

Corollary 7.7. Sub *is cartesian closed.*

Proof. We must show that

- **Sub** has a terminal object.
- **Sub** has cartesian products.
- **Sub** has exponentials.

It is routine to prove that $\mathbb{1} = (\{*\}, \cdot)$ ($\sigma \cdot * = *$ for all $\sigma \in \mathsf{Fin}$) from Subsection 2.3 is a terminal object in **Sub**.

Cartesian products are described in Definition 6.10.

Exponentials are described in Theorem 7.6. □

7.3 Limits and colimits

Recall the definition of tuples (u_I) from Definition 2.21.

Theorem 7.8. *Suppose that D is a diagram in* **Sub**. *Then:*

- *The limit* $\int_{I \in D} D$ *exists in* **Sub** *and can be concretely constructed as*

$$\int_{I \in D} D = \{(u_I) \in \prod_{I \in D} D(I) \mid \forall f : I \to J \in D.\ f u_I = u_J\}.$$

- *The colimit* $\int^{I \in D} D$ *also exists in* **Sub** *and can be concretely constructed as*

$$\int^{I \in D} D = \{[v_I]_\sim \mid I \in D,\ v_I \in D(I)\}$$

where

– $\sim$ *is the equivalence relation generated by* $v_I \sim D(f)(v_I)$ *for every* $v_I \in D(I)$ *and* $f : I \to J \in D$, *and*

– $[v_I]_\sim$ *is the* $\sim$*-equivalence class of* v_I.

Proof. Limits. Suppose X is a cone over D with arrows ξ_I. The corresponding map takes $x \in X$ to $(\xi_I x)$. The only nontrivial part is to verify that $(\xi_I x)$ has finite support, which is immediate from the fact that $supp(\xi_I x) \subseteq supp(x)$ by Corollary 3.9, and from part 2 of Lemma 2.20.

Colimits. The argument for colimits is similar, but simpler. □

7.4 Sub is a topos

Definition 7.9. *Let* Ω *be specified by:*

- $|\Omega|$ *is the set of* $U \subseteq \mathsf{Fin}$ *such that:*
  - *If* $\sigma \in U$ *then* $\mu \circ \sigma \in U$ *for all* $\mu \in \mathsf{Fin}$.
  - *There exists some finite* $S \subseteq \mathbb{A}$ *such that if* $\mu \in U$ *then*

$$\text{for all } \sigma \in \mathsf{Fin} \text{ if } \sigma|_S = id|_S \text{ then } \mu \circ \sigma \in U.$$

- *If* $U \in |\Omega|$ *and* $\sigma \in \mathsf{Fin}$ *then* $\sigma \cdot U$ *is defined by*

$$\sigma \cdot U = \{\mu \mid \mu \circ \sigma \in U\}.$$

Lemma 7.10. $U \in |\Omega|$ *is finitely supported according to the renaming action defined in Definition 7.9. Therefore,* Ω is *a nominal renaming set.*

Proof. Direct from the construction. □

Remark 7.11. For example:

- Fin and $\emptyset$ are in $|\Omega|$. As an immediate corollary $|\Omega|$ is non-empty.
- $\{\sigma \mid \sigma \in \mathsf{Fin} \wedge (\sigma \text{ not bijective})\}$ is in $|\Omega|$.
- Per is not in $|\Omega|$ (the set of finitely supported permutations; Definition 4.1).

Lemma 7.12. *Suppose that* $U \in |\Omega|$ *and* $\mu, \sigma \in \mathsf{Fin}$.
Then $\mu \in \sigma \cdot U$ *if and only if* $\mu \circ \sigma \in U$.
As an immediate corollary, $id \in \sigma \cdot U$ *if and only if* $\sigma \in U$.

Proof. By definition $\sigma \cdot U = \{\mu \mid \mu \circ \sigma \in U\}$. The result follows. □

Lemma 7.13. *Suppose that* $U \in |\Omega|$.
Then $id \in U$ *if and only if* $U = \mathsf{Fin}$.

Proof. We prove two implications. If $U = \mathsf{Fin}$ then $id \in U$ is immediate.

Now suppose that $id \in U$. By Definition 7.9, if $\sigma \in U$ then $\mu \circ \sigma \in U$ for all $\mu \in \mathsf{Fin}$. The result follows taking $\sigma = id$. □

Theorem 7.14. *A 1-1 correspondence (a bijection) between subobjects of* $\mathbb{X}$ *and arrows* $\mathbb{X} \to \Omega$ *in* **Sub** *is given by:*

- $\alpha \quad \mathbb{U} \subseteq |\mathbb{X}|$ *maps to* $\alpha(\mathbb{U}) = \lambda x{\in}|\mathbb{X}|.\ \{\mu \mid \mu \cdot x \in |\mathbb{U}|\}$.

- $\beta \quad F : \mathbb{X} \longrightarrow \Omega$ *maps to* $\beta(F) = \{x \mid id \in F(x)\}$ *(with the renaming action inherited from* $\mathbb{X}$*).*

Proof. We check that α and β are inverse. Suppose that $\mathbb{U} \subseteq \mathbb{X}$ and suppose that $F : \mathbb{X} \longrightarrow \Omega$. Then:

$$\begin{aligned}\beta(\alpha(\mathbb{U})) &= \{x \mid id \in \alpha(\mathbb{U})(x)\}\\ &= \{x \mid id \in \{\mu \mid \mu \cdot x \in |\mathbb{U}|\}\}\\ &= \{x \mid x \in |\mathbb{U}|\}\\ &= \mathbb{U}\end{aligned}$$

$$\begin{aligned}\alpha(\beta(F)) &= \alpha(\{x \mid id \in F(x)\})\\ &= \lambda x' \in |\mathbb{X}|.\ \{\mu \mid \mu \cdot x' \in \{x \mid id \in F(x)\}\}\\ &= \lambda x' \in |\mathbb{X}|.\ \{\mu \mid id \in F(\mu \cdot x')\}\\ &= \lambda x' \in |\mathbb{X}|.\ \{\mu \mid \mu \in F(x')\} && \text{Lemma 7.12}\\ &= \lambda x' \in |\mathbb{X}|.\ F(x')\\ &= F\end{aligned}$$

Suppose that $\mathbb{U} \subseteq \mathbb{X}$. We check that $\alpha(\mathbb{U})$ maps $|\mathbb{X}|$ to $|\Omega|$. Suppose that $x \in |\mathbb{X}|$. It suffices to verify that:

- If $\sigma \in \alpha(\mathbb{U})(x)$ then $\mu \circ \sigma \in \alpha(\mathbb{U})(x)$.

  If $\sigma \in \alpha(\mathbb{U})(x)$ then $\sigma \cdot x \in |\mathbb{U}|$. $|\mathbb{U}|$ is closed under the renaming action, so also $\mu \cdot \sigma \cdot x \in |\mathbb{U}|$. It follows that $\mu \circ \sigma \in \alpha(\mathbb{U})(x)$. The result follows.

- There exists some finite $S \subseteq \mathbb{A}$ such that if $\mu \in \alpha(\mathbb{U})(x)$ and $\sigma|_S = id|_S$ then $\mu \circ \sigma \in \alpha(\mathbb{U})(x)$.

  Take $S = supp(x)$. If $\mu \in \alpha(\mathbb{U})(x)$ then $\mu \cdot x \in |\mathbb{U}|$. By Theorem 2.13 $\sigma \cdot x = x$, therefore $\mu \cdot \sigma \cdot x \in |\mathbb{U}|$. Therefore $\mu \circ \sigma \in \alpha(\mathbb{U})(x)$ as required.

We check that $\alpha(\mathbb{U})$ is an arrow, that is, $\sigma \cdot \alpha(\mathbb{U})(x) = \alpha(\mathbb{U})(\sigma \cdot x)$:

$$\begin{aligned}\sigma \cdot \alpha(\mathbb{U}) &= \sigma \cdot \{\mu \mid \mu \cdot x \in \mathbb{U}\}\\ &= \{\mu' \mid \mu' \circ \sigma \in \{\mu \mid \mu \cdot x \in \mathbb{U}\}\}\\ &= \{\mu \mid (\mu \circ \sigma) \cdot x \in \mathbb{U}\}\\ &= \alpha(\mathbb{U})(\sigma \cdot x) && (\mu \circ \sigma) \cdot x = \mu \cdot (\sigma \cdot x).\end{aligned}$$

We check that $\beta(F)$ is a subobject of $\mathbb{X}$. By construction $\beta(F) \subseteq |\mathbb{X}|$. It suffices to check that if $x \in \beta(F)$ then $\sigma \cdot x \in \beta(F)$. Suppose that $x \in \beta(F)$. So $id \in F(x)$. We must show that $id \in F(\sigma \cdot x)$. Now F is an arrow so $F(\sigma \cdot x) = \sigma \cdot F(x)$. Unfolding definitions,

$$\sigma \cdot F(x) = \{\mu \mid \mu \circ \sigma \in F(x)\}.$$

So we must show that $\sigma \in F(x)$. By Lemma 7.13 since $id \in F(x)$ we know that $F(x) = \mathsf{Fin}$, and therefore $\sigma \in F(x)$ as required. □

Suppose that $\mathbb{X}$ and $\mathbb{Y}$ are nominal renaming sets.

Lemma 7.15. $F : \mathbb{X} \longrightarrow \mathbb{Y}$ *is mono if and only if* F *as a map on underlying sets* $F \in |\mathbb{X}| \to |\mathbb{Y}|$ *is injective.*

Proof. Routine. □

Lemma 7.16. *A subobject of $\mathbb{X}$ may be uniquely identified with a nominal renaming set $\mathbb{U}$ such that $\mathbb{U} \subseteq \mathbb{X}$ (Definition 6.1).*

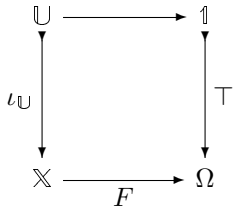
Proof. A subobject of $\mathbb{X}$ is an equivalence class of isomorphic monos into $\mathbb{X}$. It is not hard to see that $\mathbb{U}$ identifies the equivalence class of the mono which is the natural inclusion function, which we write

$$\iota_{\mathbb{U}} : \mathbb{U} \longrightarrow \mathbb{X}$$

mapping $x \in |\mathbb{U}|$ to itself, that is, to $x \in |\mathbb{X}|$. □

Definition 7.17. *Write $\top$ for the map from $|\mathbb{1}|$ (Subsection 2.3) to $|\Omega|$ mapping $* \in |\mathbb{1}|$ to $\mathsf{Fin} \in |\Omega|$. It is not hard to check that $\top$ is an arrow. By Lemma 7.15 $\top$ is also mono.*

Lemma 7.18. *A diagram*



is a pullback if and only if $\{x \mid id \in F(x)\} = |\mathbb{U}|$.

Proof. By routine calculations. □

Corollary 7.19. Sub *is a topos.*

Proof. It suffices (see [3, Volume III, Definition 5.1.3]) to prove that:

- **Sub** has finite limits.
- **Sub** is cartesian closed.
- **Sub** has a subobject classifier.

Sub is cartesian closed by Theorem 7.6. **Sub** has (finite) limits by Theorem 7.8. Ω is a subobject classifier by Theorem 7.14, and Lemmas 7.16 and 7.18. □

Sub is not a boolean topos (the category of nominal sets is [14]). We can read off what kind of logic **Sub** gives rise to; truth values are sets of substitutions. Intuitively, the truth-value associated to '$\phi(x)$' is the set of renamings such that ϕ *is* true of $\sigma \cdot x$. Intuitively, a truth-value of '$x = y$' in **Sub** is the collection of unifiers of x and y.

This will remain true also if we generalise the renaming action to substitute arbitrary elements (from some set) for atoms. Therefore, relatives of **Sub** might be useful for studying rewriting and unification. This is future work.

8 Conclusions

8.1 Variants on the theme of nominal renaming sets

Exploding models. Renaming sets are a set with a substitution action (of atoms for atoms). One of our long-term research aims is to suggest that substitution is a mathematical phenomenon worthy of independent study [13, 17].

Substitution is traditionally understood as a syntactic adjunct to function application. Function-application is modelled by function application in denotation (more-or-less by definition) — but by β-equivalence in syntax. β-equivalence in turn is implemented using substitution; substitution does not feature in the denotation and indeed neither do variable symbols (here, compare with Definition 2.1).

Exploding models (Subsection 2.3) display behaviour different from what we expect of datatypes of abstract syntax. There are two ways to look at this: as proof that substitution is more interesting than generally acknowledged, or as proof that **Sub** is not *quite* what we should be looking at. I prefer the former interpretation, but note that if we insist that $supp(\sigma \cdot x) = \sigma \cdot supp(x)$ in Definition 2.8 then the problem goes away by definition (and the resulting category seems to behave much like **Sub** in other respects; verifying this in full detail is future work). It is not hard to show that this can also be expressed by the condition

$$[a{\mapsto}b] \cdot [a'{\mapsto}b'] \cdot x = [a'{\mapsto}b'] \cdot x \Rightarrow [a{\mapsto}b] \cdot x = x$$

for all a', b', a, $x \in |\mathbb{X}|$, and fresh b (so $b \notin S$ where S is defined in Definition 2.8). This gives rise to a full subcategory of **Sub** containing precisely the 'non-exploding' models.

Models with more arrows. We just discussed how to remove parts of **Sub**. We can also reasonably add to it. In Definition 7.1 we insisted of an arrow $F : \mathbb{X} \longrightarrow \mathbb{Y}$ that $\sigma \cdot F(x) = F(\sigma \cdot x)$ for all $\sigma \in \mathsf{Fin}$ and $x \in |\mathbb{X}|$. We can alternatively insist just that $\pi \cdot F(x) = F(\pi \cdot x)$ for all $\pi \in \mathsf{Per}$ and $x \in |\mathbb{X}|$. This is the notion of arrow from nominal permutative techniques, but now it is pressed into service between sets with a renaming action.

An advantage is that there are more arrows between objects and more functions can be captured in the category, including those that distinguish between atoms.

This issue could also be addressed in the style of [9, Section 3] by setting up an adjunction between the category of nominal sets [14] and **Sub**; there should be no technical difficulty in doing this.

More general substitutions. Finally, we can consider a category of sets based on 'substitution plus finite support' where the substitution is of atoms for elements (for some collection of elements). A direct connection is possible with previous work by the myself with Mathijssen, where precisely this scenario is considered — but using purely syntactic techniques [13, 17].

It remains to investigate all of the variations above.

A basic message of this paper is that 'nominal techniques' can encompass 'substitution plus finite support' just as easily as 'permutations plus finite support', and the precise notion of 'substitution' and indeed of 'arrow' can be

fine-tuned. We have considered a renaming action (substitution of atoms for atoms) in this paper — but it is clear that other design decisions in the same style but with different details, are possible. We hope that the mathematics in this paper will be a stimulus for research to use categories similar to **Sub** as denotations for more than sets with a renaming action alone.

8.2 Nominal renaming sets and higher-order abstract syntax

I was involved in developing nominal techniques. As discussed, this paper makes a point that the inductive datatypes which 'permutations plus finite support' was developed to support, are supported just as well by 'renamings and finite support'.

A problem with nominal techniques is that their non-standard universe is not easily supported (see [11] and [10, Chapter 3]) by existing technology. This existing technology is almost universally based on Higher-Order Logic (HOL) or Zermelo-Fraenkel set theory (ZF) [7, 27, 22, 16] (I would also include COQ in this group [15]).

This puts nominal techniques at a disadvantage relative to other techniques, such as de Bruijn indexes [6] or weak or strong Higher-Order Abstract Syntax (HOAS) [23]. These have their own issues to overcome; a good argument in favour of nominal techniques is in [25]. Nevertheless, other techniques are easier to implement from a standing start and they have been studied longer and more extensively. Thus, at least for the moment, they exist in more highly-developed form (for example [24]) than the most advanced nominal-style implementation [5]. The current state of the art is that these difficulties can be circumvented but not removed.

A large part of the problem can be traced to the fact that $[\mathbb{A}]\mathbb{X}$ is not a set of functions (it is a set of *partial* functions [14, Equation (42)]). Therefore Theorem 5.7 of this paper holds out a hope for a 'nominal HOAS'. A place to begin checking this is to check whether the Theory of Contexts [18] is sound in **Sub**. This collection of axioms formalises some desirable properties of abstract syntax — properties which have a noticable 'nominal' flavour. The Theory of Contexts was developed independently but recent developments have exploited nominal methods [19]; perhaps **Sub** will serve as a useful stepping-stone between nominal techniques and the HOAS-style Theory of Contexts.

Another place to look for connections is to build Miller and Tiu's ∇-quantifier using a model based on **Sub** [21], or provide a semantics for higher-order patterns, also by Miller [20].

There is therefore a reasonable possibility that **Sub** will give useful guidance on how to reconcile nominal techniques with existing higher-order logic technology, be it with HOAS or just for an independently-developed implementation. This is future work.

8.3 Related work

An interesting feature of nominal techniques is that they do not give denotation only to datatypes of abstract syntax. Nominal notions such as 'freshness' and 'abstraction' can be applied equally to non-syntactic structures such as function-spaces. This has been done, for example, in work on 'nominal pointers'

[2] and 'nominal games' [1]. This gives nominal techniques a distinct foundational flavour. It is common to identify 'nominal techniqes' as 'an approach to syntax with binding'. That is certainly part of the story, but only a part; they are a mathematical model of names. This paper demonstrates that the model accommodates names that can be renamed as well as permuted.

The semantic approach to names is in common with work based on categories of presheaves [9, 8]. In that work, abstraction is also modelled using an exponential [8, Equation (8)]. Nevertheless, **Sub** is distinct from these. The crucial difference is Theorem 2.13, which states that every has a single unique least supporting set. This property is enjoyed in the categories of nominal permutation sets and nominal renaming sets. It is not enjoyed in the work based on presheaves, and this property leads to the (in my opinion, simple) sets-based presentation used in this paper and elsewhere.

We now propose two hypotheses:

Definition 8.1. *Let* **Var** *be the category with objects finite subsets of* $\mathbb{A}$ *and arrows functions between them. Let* **Set** *be the category of all sets and all functions between them.*

Let **PBM** *be the category of presheaves in* $\mathbf{Set}^{\mathbf{Var}}$ *(functors from* **Var** *to* **Set***) that preserve pullbacks of pairs of monos in* **Var** *(injections), and natural transformations between them.*

Hypothesis 8.1. Sub *is equivalent to* **PBM***.*

Intuitively, the property of preserving pullbacks of pairs of monos is the property of preserving intersections (see Lemma 2.12).

Definition 8.2. *For* $A \in \mathbf{Var}$ *write* $\mathbf{y}(A)$ *for the functor taking* B *to* $\mathbf{Var}(A, B)$*, with the natural right-composition action on arrows.*

Specify a topology on $\mathbf{Set}^{\mathbf{Var}}$ *by letting its basis be those subfunctors of* $\mathbf{y}(A)$ *generated by some injection* $f : A \longrightarrow\!\!\!\!\!\rightarrow B$*.*

Hypothesis 8.2. PBM *can be presented as a topos for the topology of Definition 8.2.*

In previous work with Mathijssen we have developed Nominal Algebra [17]. This is a framework for algebraic reasoning in the presence of binding. Using notation from [17] we propose one more hypothesis:

Hypothesis 8.3. Sub *is equivalent to the category of models of the following theory in nominal algebra:*

$$
\begin{aligned}
a\#x \vdash x[a{\mapsto}b] &= x \\
x[a{\mapsto}b][a'{\mapsto}b'] &= x[a'{\mapsto}b'][a{\mapsto}b] \\
x[a{\mapsto}b][b{\mapsto}c] &= x[a{\mapsto}c][b{\mapsto}c] \\
x[a{\mapsto}a] &= x \\
x[a{\mapsto}c][b{\mapsto}c] &= x[b{\mapsto}c][a{\mapsto}c]
\end{aligned}
$$

There is scope for a whole second paper in the checking (and possible refinement) of these hypotheses. This is future work.

We hope that **Sub** can be the basis of future research directly — starting perhaps with the Theory of Contexts and with the three hypotheses above — and will serve as an example of other, similar, objects of study based on the idea 'X plus finite support' for different values of X.

References

[1] S. Abramsky, D. R. Ghica, A. S. Murawski, C.-H. L. Ong, and I. D. B. Stark. Nominal games and full abstraction for the nu-calculus. In *LICS*, pages 150–159. IEEE, 2004.

[2] Nick Benton and Benjamin Leperchey. Relational reasoning in a nominal semantics for storage. In *Proc. of the 7th Int'l Conf. on Typed Lambda Calculi and Applications (TLCA)*, volume 3461 of *LNCS*, pages 86–101, 2005.

[3] F. Borceux. *Handbook of Categorical Algebra*. Number 50, 51, 52 in Encyclopedia of Mathematics and its Applications. Cambridge University Press, Great Britain, 1994.

[4] N. Brunner. 75 years of independence proofs by fraenkel-mostowski permutation models, 1996.

[5] Christine Tasson Christian Urban. Nominal techniques in isabelle/hol. In *CADE 2005*, volume 3632 of *Lecture Notes in Artificial Intelligence*, pages 38–53, 2005.

[6] N. 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.

[7] William M. Farmer. The seven virtues of simple type theory. Technical Report 18, McMaster University, SQRL, 2003 (revised 2006).

[8] M. P. Fiore, G. D. Plotkin, and D. Turi. Abstract syntax and variable binding. In *14th Annual Symposium on Logic in Computer Science*, pages 193–202. IEEE Computer Society Press, 1999.

[9] Marcelo Fiore and Daniele Turi. Semantics of name and value passing. In *Proc. 16$^{\text{th}}$ LICS Conf.*, pages 93–104. IEEE, Computer Society Press, 2001.

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

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

[12] Murdoch J. Gabbay. A general mathematics of names. *Information and Computation*, 205(7):982–1011, 2007.

[13] Murdoch J. Gabbay and Aad Mathijssen. Capture-avoiding substitution as a nominal algebra (journal version). *Formal Aspects of Computing*, 2008. Available online.

[14] 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.

[15] Gérard Huet, Gilles Kahn, and Christine Paulin-Mohring. The Coq proof assistant, a tutorial.
`http://pauillac.inria.fr/coq/doc/tutorial.html`. LogiCal Project.

[16] Thomas Jech. Set theory. In Edward N. Zalta, editor, *The Stanford Encyclopedia of Philosophy*. Fall 2002.

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

[18] Marino Miculan. Developing (meta)theory of lambda-calculus in the theory of contexts. *ENTCS*, 1(58), 2001.

[19] Marino Miculan, Ivan Scagnetto, and Furio Honsell. Translating specifications from nominal logic to cic with the theory of contexts. In *MERLIN*, pages 41–49. ACM, 2005.

[20] Dale Miller. A logic programming language with lambda-abstraction, function variables, and simple unification. *Extensions of Logic Programming*, 475:253–281, 1991.

[21] Dale Miller and Alwen Tiu. A proof theory for generic judgments: An extended abstract. In *LICS*, pages 118–127. IEEE, 2003.

[22] Lawrence C. Paulson. The foundation of a generic theorem prover. *Journal of Automated Reasoning*, 5(3):363–397, 1989.

[23] F. Pfenning and C. Elliot. Higher-order abstract syntax. In *PLDI '88: Proc. of the ACM SIGPLAN 1988 conf. on Programming Language design and Implementation*, pages 199–208. ACM Press, 1988.

[24] Frank Pfenning and Carsten Schürmann. System description: Twelf - a meta-logical framework for deductive systems. In H. Ganzinger, editor, *CADE-16, 16th Int'l Conf. on Automated Deduction*, pages 202–206. Springer, 1999.

[25] Andrew M. Pitts. Equivariant syntax and semantics. In *ICALP*, pages 32–36. Springer-Verlag, 2002.

[26] A. Tarski. A lattice-theoretical fixpoint theorem and its applications. *Pacific Journal of Mathematics*, 5:285–309, 1955.

[27] Johan van Benthem. Higher-order logic. In D.M. Gabbay and F. Guenthner, editors, *Handbook of Philosophical Logic, 2nd Edition*, volume 1, pages 189–244. Kluwer, 2001.