An observation on support and freshness in nominal sets (Technical report)

Murdoch J. Gabbay

There are two natural ways of excluding an atom a in nominal techniques: we can either consider the sets X such that a is fresh for X, or we can consider the sets X such that a is fresh for every $x \in X$.

The statements of 'being fresh for all the elements of that set' and 'being fresh for a set' are not the same: it is not the case that $\forall x \in X.a\#x$ if and only if $a\#X$.

Both notions encode natural notions of 'fresh a'. In this paper, it is proved that these notions lead naturally to two categories that are isomorphic, so that in a suitable generalised sense they are the same.

The result is mathematically attractive and has an interesting reading: it is equivalent to add a fresh atom to the underlying universe, and to add a symbol to the meta-language referencing a fresh atom. Or to put it slightly differently: we prove the intuitively appealing but non-obvious fact that a fresh atom in the object-level is categorically isomorphic to a fresh atom in the meta-level.

Contents

1 Introduction

1.1 A little background on nominal semantics

Nominal techniques were introduced in [14]. With Pitts, we considered the problem of giving semantics to inductive definitions with binding. We did this by considering inductively defined datatypes in *Fraenkel-Mostowski set theory* (**FM sets**).

FM sets was developed to prove the independence of the Axiom of Choice [2]. It is not obvious that a representation of binding is to be found in a set theory from the earlier half of the 20^{th} century, but it is. This semantics is now used directly or through various logical and operational presentations such as nominal logic, FreshML, or the nominal datatypes package. See Mulligan's online bibliography [21].

Nominal techniques give variable symbols a denotational reality as atoms, or *urelemente*. If the reader is familiar with nominal logic [23], FreshML [26], or the nominal datatypes package in Isabelle [28], they will recognise this idea in the *datatype of atoms* $\mathbb{A}$, used to represent variable symbols. Atoms are data.

The specific mathematical content here is in the fact that in FM sets and the systems which were then derived from it, this 'datatype of atoms' displays some striking behaviour which is a good semantic model of names and binding. This behaviour is distinctive and distinguishes nominal techniques from models of variable symbols based on functions [22], numbers [5], or links [1, Section 1].

Much of the content of FM sets is shared with work using presheaves [8]. The extra benefit of using FM sets is its *sets-based* presentation, which enjoys a unique notion of support.[1]

1.2 The support of a set X is not the support of its elements $x \in X$

In FM sets, every element has a notion of *support*, $supp(x)$ (full definitions will follow). This generalises the syntactic notion of 'free variable symbols of'. In the case that the set represents an abstract syntax tree, support and 'free variable symbols of' can be made to coincide in a natural way.

However, the support of X does not coincide with 'atoms in the transitive closure of X', nor with 'union of the supports of all $x \in X$'.[2]

In my own experience, it is hard for people to see how a set that 'contains a' might not contain a in its support. There are strong intuitions that $a \in X$ should imply $a \in supp(X)$, and that $a \notin x$ for all $x \in X$ should imply $a \notin supp(X)$.

This will be very familiar to some readers, and perhaps less so to others. Here are two illustrative examples:

Example 1.1.
- $supp(\mathbb{A}) = \varnothing$ but the atoms in the transitive closure of $\mathbb{A}$ is $\mathbb{A}$. Also $\bigcup\{supp(x) \mid x \in \mathbb{A}\} = \mathbb{A}$.

  a is 'in' the set of all atoms, but because so is every other atom, it is not in the support of the set of all atoms.

[1] Presheaves have a version of support, but in a sense that can be made entirely formal, it is possible for an element to have many distinct supports. See [12] for a more detailed discussion.

[2] The transitive closure of X is the set containing the elements of X, the elements of the elements of X, and so on. It is the least fixedpoint of the mapping $X \mapsto X \cup \bigcup X$.

- Also, $supp(\mathbb{A}\setminus\{a\})=\{a\}$ but the atoms in the transitive closure of $\mathbb{A}\setminus\{a\}$ and $\bigcup\{supp(x)\mid x\in\mathbb{A}\setminus\{a\}\}$ are both equal to $\mathbb{A}\setminus\{a\}$.

  a is 'not in' the set of all atoms minus a, but since it thus distinguishes itself by this absence, it *is* in the support of the set of all atoms minus a.

The behaviour illustrated above is important: the correctness of the self-duality of the И-quantifier, of atoms-abstraction, and the nominal model of abstract-syntax-with-binding, intimately depend on this.

As standard, write $a\#x$ for $a\notin supp(x)$. The fact is that it is not the case that $a\#X$ if and only if $\forall x\in X.a\#x$. This is why we not only do not, but *can not* define *supp* by induction on sets (whereas 'free variables of' is and must be defined by induction). Thinking of X as a predicate and $x\in X$ as a datum, and using a bit of nominal jargon, this is related to the fact that in nominal semantics equivariant functions and predicates can and do act on non-equivariant data.

1.3 The main results

In this paper, I will use a little bit of category theory and some elementary sets constructions to exhibit a sense in which *natural generalisations* of 'support of X' and 'support of all $x\in X$' — *are* equivalent.

For the reader already familiar with nominal techniques I will now briefly sketch the main definition and two results of this paper; full details will follow.

Write FMSet for the category with objects FM sets and arrows function(-sets) between them (see Definition 3.2 for the precise definition).

Our first main result is Theorem 3.3. This presents the strongest positive and direct connection I know of in general, between the support of X in FMSet and the support of $x\in X$.

To state the next result we need just a little notation.

Definition 1.2. Fix some atom a and define two new categories using FMSet:

- Define a category $\mathsf{FMSet}_{\#\mathsf{a}}$ by:
  - Objects are elements X of FMSet such that $\forall x{\in}X.a\#x$.
  - Arrows are the full set of arrows $f:X\longrightarrow Y\in\mathsf{FMSet}$.
- Define a category $\mathsf{FMSet}_{\nu\mathsf{a}}$ by:
  - Objects are elements X of FMSet such that $a\#X$.
  - Arrows are arrows $f:X\longrightarrow Y\in\mathsf{FMSet}$ such that $a\#f$.

Remark 1.3. $\mathsf{FMSet}_{\#\mathsf{a}}$ and $\mathsf{FMSet}_{\nu\mathsf{a}}$ both implement notions of 'exclude a':

- In $\mathsf{FMSet}_{\#\mathsf{a}}$, a is excluded in the sense of 'fresh for all the elements of'.
- In $\mathsf{FMSet}_{\nu\mathsf{a}}$, a is excluded in the sense of 'fresh for'.

The second main result of this paper is Theorem 5.11. This states that $\mathsf{FMSet}_{\#\mathsf{a}}$ and $\mathsf{FMSet}_{\nu\mathsf{a}}$ are isomorphic categories. In this sense, 'fresh for' and 'fresh for all elements of' are the same after all.

Another way of reading this is particularly interesting.

- In $\mathsf{FMSet}_{\#\mathsf{a}}$ a is fresh 'at the object level'; the arrows represent functions that operate on elements without a in their support.

  Note that the functions themselves will have a in their support; think of the characteristic function of $\mathbb{A}\setminus\{a\}$.

- In $\mathsf{FMSet}_{\mathsf{v}\mathsf{a}}$ a is fresh 'at the meta-level'; the arrows represent functions without a in their support.

  Note that the functions may operate on elements with a in their support; think of the characteristic function of $\mathbb{A}$.

(Just to locate all of this relative to the category of nominal sets NOM: the category NOM of equivariant FM sets and equivariant functions between them is a limit of the $\mathsf{v}a$ construction. It represents a totally equivariant meta-language, operating on data that need not be equivariant.)

The main result asserts an equivalence between freshness at the meta-level and freshness at the object-level in the sense above. Or, to put it slightly differently: we prove the intuitively appealing but non-obvious fact that a fresh atom in the object-level is categorically isomorphic to a fresh atom in the meta-level.

This paper presents part of the mathematics of a larger manuscript [11], in a more accessible form and with an exposition tailored to Theorems 3.3 and 5.11.

2 Basic nominal constructions

2.1 The cumulative hierarchy

Definition 2.1. Fix a countably infinite set $\mathbb{A}$ of **atoms**. $a,b,c,\dots$ will range over distinct elements of $\mathbb{A}$. We call this the **permutative convention**.

Definition 2.2. We define a collection of **elements** $\mathscr{U}$ in the style of von Neumann [17] by ordinal induction as follows:

1. $\mathscr{U}_0 = \mathbb{A}$.
2. If $\alpha < \beta$ and $U \in \mathscr{U}_\alpha$ then $U \in \mathscr{U}_\beta$.
3. If $U \subseteq \bigcup_{\alpha<\beta} \mathscr{U}_\alpha$ then $U \in \mathscr{U}_\beta$.

We let $\mathscr{U}$ be the collection of all x such that $x \in \mathscr{U}_\alpha$ for some α.

$\mathscr{U}$ is a standard **cumulative hierarchy model** of Zermelo-Fraenkel set theory with atoms (**ZFA**). Examples are illustrated in Figure 1. This construction has been used for example in [9, 14, 4].

$\mathscr{U}$ is the least pre-fixedpoint of the operator 'take the powerset of'; $\mathit{powerset}(\mathscr{U}) \subseteq \mathscr{U}$.

Definition 2.3. Write $x \in \mathscr{U}$ for '$x \in \mathscr{U}_\alpha$, for some ordinal α', and read this as *x is an element*. x will range over elements of $\mathscr{U}$.

Definition 2.4. Call a non-atomic element a **set**. That is, x is a set when $x \in \mathscr{U}$ and $x \notin \mathbb{A}$.

If X is a set then $X = \{x \mid x \in X\}$. This is not the case of atoms. For example $a \neq \{x \mid x \in a\} = \varnothing$.

Definition 2.5. Let x and y be elements. Let X and Y be sets. Implement the **ordered pair** (x,y) and **product set** $X \times Y$ by

$$(x,y) = \{\{x\},\{x,y\}\} \qquad X \times Y = \{(x,y) \mid x \in X,\ y \in Y\}.$$

Functions are implemented as *graphs* $f = \{(x, f(x)\}$ that are sets, as is standard. We let f, g range over elements that are function-sets.

Write $X \to Y$ for the set of function-sets with domain X and range a subset of Y.

$$
\begin{array}{cccc}
a \in \mathscr{U}_0 & b \in \mathscr{U}_0 & \varnothing \subseteq \mathscr{U}_0 & \mathbb{A} \subseteq \mathscr{U}_0 \\
\{a\} \in \mathscr{U}_1 & \{a,b\} \in \mathscr{U}_1 & \varnothing \in \mathscr{U}_1 & \mathbb{A} \in \mathscr{U}_1 \\
\{\{a\},\{a,b\}\} \in \mathscr{U}_2 & \{\varnothing\} \in \mathscr{U}_2 & \mathbb{A}\cup\{\mathbb{A}\} \in \mathscr{U}_2 & \\
& \vdots & & \\
& \mathbb{N} = \{0,1,2,\dots\} \in \mathscr{U}_\omega & & \\
& \vdots & &
\end{array}
$$

Figure 1: Example sets in the cumulative hierarchy

2.2 Permutations

Definition 2.6. A **permutation** π is a bijection on $\mathbb{A}$ such that $\{a \mid \pi(a) \neq a\}$ is finite (we say that π has **finite support**). π, π', τ will range over permutations. We also use the following notation:

- Write *id* for the **identity permutation**, so $id(a) = a$ always.
- Write $\circ$ for functional composition. So $(\pi \circ \pi')(a) = \pi(\pi'(a))$.
- Write $\pi^{\text{-}1}$ for the inverse of π, so $\pi \circ \pi^{\text{-}1} = id = \pi^{\text{-}1} \circ \pi$.
- Write $\mathbb{P}$ for the set of all permutations.

Definition 2.7. We define a **permutation action** inductively by:

$$\pi{\cdot}a = \pi(a) \qquad \pi{\cdot}X = \{\pi{\cdot}x \mid x \in X\} \quad (X \text{ not an atom})$$

Lemma 2.8. $\mathrm{id}{\cdot}x = x$ *and* $\pi'{\cdot}(\pi{\cdot}x) = (\pi' \circ \pi){\cdot}x$.
In words: permutation is a group action on $\mathscr{U}$.

Proof. By a routine induction on $\mathscr{U}$.

- The case of an atom a. From Definition 2.7 it is immediate that $\mathrm{id}{\cdot}a = a$ and $\pi'{\cdot}(\pi{\cdot}a) = \pi'(\pi(a)) = (\pi' \circ \pi){\cdot}a$.
- The case of a set X. From Definition 2.7 and the inductive hypothesis for every $x \in X$. □

2.3 Support

Definition 2.9. Let A be a finite set of atoms.

- Write $\mathit{fix}(A) = \{\pi \mid \forall a \in A.\pi(a) = a\}$.
- Say that A **supports** x when $\pi{\cdot}x = x$ for all $\pi \in \mathit{fix}(A)$.
- Say x has **finite support** when some finite A supporting x exists.
- Define $\mathit{supp}(x)$ the **support** of x by

  $$\mathit{supp}(x) = \bigcap\{A \mid A \text{ a finite set of atoms supporting } x\}$$

  if x has finite support, and $\mathit{supp}(x)$ is undefined otherwise.
- Write $a\#x$ when $a \notin \mathit{supp}(x)$ and call a **fresh for** x. Write $a\#x,y,z$ for '$a\#x$ and $a\#y$ and $a\#z$', and so on.

Remark 2.10. Not every element of $\mathscr{U}$ has finite support. Make a fixed but arbitrary choice of bijection of $\mathbb{A}$ with the natural numbers $\{0,1,2,3,4,5,\dots\}$. Let *comb* $\subseteq \mathbb{A}$ be the element corresponding under this bijection with the even numbers $\{0,2,4,\dots\}$.

comb contains 'every other atom' $\{a,c,e,g,\dots\}$.

There is no finite $A \subseteq \mathbb{A}$ such that if $\pi \in \mathit{fix}(A)$ then $\pi{\cdot}\mathit{comb} = \mathit{comb}$.

Theorem 2.11. *A supports x if and only if $\pi{\cdot}A$ supports $\pi{\cdot}x$.*

As an immediate corollary, $\pi{\cdot}\mathit{supp}(x) = \mathit{supp}(\pi{\cdot}x)$.

Definition 2.12. If π is a permutation and $A \subseteq \mathbb{A}$ is a set of atoms, write $\pi|_A$ for the partial function such that

$$\pi|_A(a) = \begin{cases} \pi(a) & \text{if } a \in A \\ \text{undefined} & \text{if } a \in \mathbb{A}\setminus A. \end{cases}$$

Theorem 2.13. *Let x be any element. If A and B are finite and support x then so does $A \cap B$. As a corollary:*

1. *If x has a finite supporting set then it has a least finite supporting set and this is equal to $\mathit{supp}(x)$.*
2. *If $\pi|_{\mathit{supp}(x)} = \pi'|_{\mathit{supp}(x)}$ then $\pi{\cdot}x = \pi' x$.*

Proof. Suppose τ fixes $A \cap B$ pointwise. We must show $\tau{\cdot}x = x$. Write

$$K \quad \text{for} \quad \{a \mid \tau(a) \neq a\}.$$

Choose an injection ι of $B\setminus A$ into $\mathbb{A}\setminus(A \cup B \cup K)$. Define a permutation π by $\pi(a) = \iota(a)$ if $a \in B\setminus A$, $\pi(\iota(a)) = a$ if $a \in B\setminus A$, and $\pi(b) = b$ if $b \notin B\setminus A$ and $\iota^{-1}(b) \notin B\setminus A$. Note that $\pi\circ\pi = \mathit{id}$, so $\pi = \pi^{-1}$. π fixes A pointwise so $\pi{\cdot}x = x$. Also $\pi\circ\tau\circ\pi$ fixes B pointwise so $(\pi\circ\tau\circ\pi){\cdot}x = x$. We apply π to both sides, use Lemma 2.8, and simplify, and conclude that $\tau{\cdot}x = x$ as required.

The first corollary follows from the fact that a descending chain of finite sets ordered by strict subset inclusion, is finite. The second corollary follows directly from the definition of support in Definition 2.9. □

2.4 Equivariance

Definition 2.14. The language of ZFA set theory is first-order logic with equality $=$ and a binary predicate $\in$ called **set inclusion**. ϕ will range over predicates in this language.

Theorem 2.15. *Suppose $\phi(\overline{x})$ is a predicate mentioning only variables from the list $\overline{x}$. Then*

$$\phi(\overline{x}) \Leftrightarrow \phi(\pi{\cdot}\overline{x}).$$

Sketch proof. Atoms are atomic; if we build one model of ZFA sets then we can permute its atoms to obtain another model. The result follows by soundness and completeness of first-order logic [27]. □

The reader can find much more on permutations in [25]. Indeed, the use of permutations of variable symbols predates nominal techniques; see for example [19, Subsection 9.2].

Definition 2.16. As is standard, we can specify a map χ using a predicate $\phi(\overline{x},z)$ such that

$$\forall \overline{x}.\Big((\exists z.\phi(\overline{x},z)) \wedge (\forall z,z'.\phi(\overline{x},z) \wedge \phi(\overline{x},z') \Rightarrow z = z')\Big).$$

Corollary 2.17. *Suppose $\chi(\overline{x})$ is a function specified using a predicate mentioning only variables from the list $\overline{x}, z$. Then*

$$\pi \cdot \chi(\overline{x}) = \chi(\pi \cdot \overline{x}).$$

Proof. We unpack Definition 2.16 and use equivariance (Theorem 2.15). □

Theorem 2.18 is useful and easy to prove:

Theorem 2.18. *Suppose $\chi(\overline{x})$ is a function on variables included in $\overline{x}$, which is $x_1, \ldots, x_n$. Suppose $\overline{x}$ denotes elements with finite support. Then*

$$supp(\chi(\overline{x})) \subseteq supp(x_1) \cup \cdots \cup supp(x_n).$$

As a corollary, if χ is injective then

$$supp(\chi(\overline{x})) = supp(x_1) \cup \cdots \cup supp(x_n).$$

Proof. The corollary follows by considering the result for χ and its inverse.

Suppose that $\pi \in fix(supp(x_1) \cup \cdots \cup supp(x_n))$. We reason as follows:

$$\begin{array}{rcll} \pi \cdot \chi(\overline{x}) & = & \chi(\pi \cdot \overline{x}) & \text{Corollary 2.17} \\ & = & \chi(\overline{x}) & \text{Theorem 2.13} \end{array}$$

The result follows. □

3 Relation between $supp(X)$ and $supp(x)$ for $x \in X$

Definition 3.1. Suppose that X is a set. If X has finite support and all $x \in X$ have finite support then say that X has finite support to **level** 1.

X and Y will range over sets with finite support to level 1.

Consider the notions 'fresh for' versus 'fresh for all elements of'. In symbols, consider the predicates

$$a\#X \quad \text{versus} \quad \forall x \in X.a\#x.$$

If X is *finite* then 'fresh for' and 'fresh for all elements of' coincide. This matches the naive expection discussed in the Introduction; this is exactly the behaviour displayed by (finite) name-carrying abstract syntax.

Also as discussed in the Introduction, a central feature of nominal techniques is that if X is *infinite* then the two notions part company, and no particular implication connects them in general. Recall Example 1.1.

So just because $a\#X$ does not mean that a is fresh for every element in X, and conversely, just because $a\#x$ for every $x \in X$ does not mean that a is fresh for X overall.

Definition 3.2. Define the category FMSet by:

- Objects are sets X in $\mathscr{U}$ with finite support to level 1.
- Arrows $f : X \longrightarrow Y$ are the function-sets in $X \to Y$ with finite support.

In [9, 14] we use the hereditarily finitely supported sets; in Definition 3.2 we use sets with finite support to level 1, which is not quite the same. The difference will not be important.

Theorem 3.3 is new to the best of my knowledge. It gives the strongest direct equality I know of between $supp(X)$ and $supp(x)$ for $x \in X$.

Theorem 3.3. *Suppose X has finite support to level 1. If $\bigcup\{supp(x) \mid x \in X\}$ is finite then*

$$supp(X) = \bigcup\{supp(x) \mid x \in X\}.$$

Proof. Suppose that $\bigcup\{supp(x) \mid x \in X\}$ is finite. We prove two set inclusions:

- $supp(X) \subseteq \bigcup\{supp(x) \mid x \in X\}$. If $\bigcup\{supp(x) \mid x \in X\}$ is finite then the result follows by Theorem 2.13 and the fact that $\pi \cdot X = \{\pi \cdot x \mid x \in X\}$.
- $\bigcup\{supp(x) \mid x \in X\} \subseteq supp(X)$. Suppose $x \in X$ and $a \in supp(x)$. Choose fresh b (so $b\#X$ and $b\#x'$ for every $x' \in X$). By Theorem 2.11 $supp((b\ a)\cdot x) = (b\ a)\cdot supp(x)$. Since X has no element y such that $b \in supp(y)$, we know that $(b\ a)\cdot x \neq X$ and by Theorem 2.13 it must be that $a \in supp(X)$.

□

Remark 3.4. The special case of Theorem 3.3 where X is a finite set is known. See for example [10, Lemma 2.22] or [16]. Note that Theorem 3.3 is more general, and holds for X infinite.

Thanks to anonymous referees of previous papers for suggesting the precise form of the result stated here, and for providing the reference to the Isabelle code.

4 The И quantifier

Definition 4.1. Suppose $\phi(\overline{z}, a)$ is a predicate on variables included in $\overline{z}, a$ — here $\overline{z}$ is shorthand for 'any other variables mentioned in ϕ', and we intend a to range over atoms.

The **NEW quantifier** $\text{И}a.\phi(\overline{z}, a)$ is defined by

$\text{И}a.\phi(\overline{z}, a)$ is true when $\{a \in \mathbb{A} \mid \phi(\overline{z}, a) \text{ is false}\}$ is finite.

Definition 4.2. If $\overline{z}$ is a list of variables $z_1, \dots, z_n$ write

$$a\#\overline{z} \quad \text{for} \quad a\#z_1 \wedge \dots \wedge a\#z_n.$$

Theorem 4.3 expresses the characteristic some/any property of the И-quantifier:

Theorem 4.3. *Suppose $\phi(\overline{z}, a)$ is a predicate on variables included in $\overline{z}, a$. Suppose $\overline{z}$ denotes a list of elements with finite support. Then the following are equivalent:*

$$\begin{array}{ll} \forall a.\ (a \in \mathbb{A} \wedge a\#\overline{z}) \implies \phi(\overline{z}, a) & \forall \textit{ form of } \text{И}a.\phi(\overline{z}, a) \\ \text{И}a.\phi(\overline{z}, a) & \\ \exists a.\ a \in \mathbb{A} \wedge a\#\overline{z} \wedge \phi(\overline{z}, a) & \exists \textit{ form of } \text{И}a.\phi(\overline{z}, a) \end{array}$$

Proof. All top-to-bottom implications are easy. Now suppose there exists some atom a such that $a\#\overline{z} \wedge \phi(\overline{z}, a)$. Choose any other atom b such that $b\#\overline{z}$. By Theorems 2.15 and 2.13 it follows that $\phi(\overline{z}, b)$. The result follows. □

Freshness $a\#x$ (Definition 2.9) can be characterised directly using И and equality (see [14, Equation 5] or [14, Equation 13]):

Theorem 4.4. *Let x be an element with finite support. Then*

$$a\#x \quad \textit{if and only if} \quad \text{И}b.(b\ a)\cdot x = x.$$

Proof. Suppose $a\#x$. By Theorem 2.13 if $(b\ a)\cdot x \neq x$ then $b \in supp(x)$. $supp(x)$ is finite by assumption. The result follows.

Now suppose that $\text{И}b.(b\ a)\cdot x = x$. Let B be a finite set such that for all $b \in \mathbb{A}\backslash B$, $(b\ a)\cdot x = x$. Choose any pair of distinct atoms b and b' in B. Note that

$$(b\ b') = (b\ b') \circ (b\ a) \circ (b\ a) = (b\ a) \circ (b'\ a) \circ (b\ a).$$

Therefore $(b\ b')\cdot x = x$ always.

$fix(B \cup \{a\})$ is generated as a group by elements of the form $(b'\ b)$ and $(b\ a)$ as considered above. It follows that if $\pi \in fix(B \cup \{a\})$ then $\pi\cdot x = x$. Therefore $a \notin supp(x)$. □

5 Statement and proof of the isomorphism

Fix an atom a. Recall Definitions 3.1 and 1.2 for the definitions of FMSet, $\mathsf{FMSet}_{\#\mathsf{a}}$ and $\mathsf{FMSet}_{\nu\mathsf{a}}$.

5.1 a-fresh sets

Definition 5.1. If $X \in \mathsf{FMSet}$ write

$$X_{\#a} = \{x \in X \mid a\#x\}.$$

Call this the ***a*-fresh** version of X.

If $X \in \mathsf{FMSet}$ and $Y \in \mathsf{FMSet}$ and $f \in X \longrightarrow Y$, write

$$f_{\#a} = \{(x, f(x)) \mid x \in X_{\#a}\}.$$

Definition 5.1 reprises a comment in [13, Section 7, page 9]. In [11] we use the construction to give some nice proofs of properties of atoms-abstraction. The same construction is used by Clouston in [4].

Lemma 5.2 underlines the distinctness of 'fresh for' and 'fresh for all elements of':

Lemma 5.2. *$a\#X$ does not necessarily imply that $X_{\#a} = X$.*

Proof. Consider $a\#\mathbb{A}$. Then $\mathbb{A}_{\#a} = \mathbb{A} \setminus \{a\} \neq \mathbb{A}$. □

Lemma 5.3. *If $f : X \longrightarrow Y \in \mathsf{FMSet}_{\nu\mathsf{a}}$ then $f_{\#a} : X_{\#a} \longrightarrow Y_{\#a} \in \mathsf{FMSet}_{\#\mathsf{a}}$.*

Proof. Suppose $f : X \longrightarrow Y \in \mathsf{FMSet}_{\nu\mathsf{a}}$. In particular, $a\#f$. It follows using Theorem 2.18 that $f_{\#a} : X_{\#a} \longrightarrow Y_{\#a}$. □

5.2 Atoms-restriction

A notion of *atoms-restriction* will be useful:

Definition 5.4. Suppose X is an object in FMSet. Define $\nu a.X$ by

$$\nu a.X = \{\pi'\cdot x' \mid \pi' \in fix(supp(X) \setminus \{a\}),\ x' \in X\}.$$

We read $\nu a.X$ as **restrict a in X**.

$\nu a.X$ is a model of the name-restriction of [6] and [24]; it is closely related to the *permutation orbits* of [10].

Lemma 5.5. *Suppose X is an object in* FMset*.*

1. *$supp(\nu a.X) \subseteq supp(X) \setminus \{a\}$.*
2. *It is not true in general that $supp(\nu a.X) = supp(X) \setminus \{a\}$.*
3. *$a\#X$ if and only if $\nu a.X = X$.*

Proof. 1. It is easy to check that if $\pi \in fix(supp(X) \setminus \{a\})$ then $\pi \cdot \nu a.X = \nu a.X$. The result follows by Theorem 2.13.

2. It suffices to provide a counterexample. Choose any b and take $X = (\mathbb{A} \times \mathbb{A}) \setminus \{(a,b)\}$ (Definition 2.5). It is easy to check that $\nu a.X = \mathbb{A} \times \mathbb{A}$.

3. Suppose $a\#X$. Suppose $\pi' \in fix(supp(X) \setminus \{a\})$ and $x' \in X$. By Theorems 2.15 and 2.13, $\pi' \cdot x' \in X$. It follows easily that $\nu a.X = X$.

   Conversely, if $\nu a.X = X$ then by part 1 of this result $a\#X$.

□

Lemma 5.6. 1. *Suppose X is an object in $\mathsf{FMSet}_{\nu a}$. Then $\nu a.(X_{\#a}) = X$.*
2. *Suppose X is an object in $\mathsf{FMSet}_{\#a}$. Then $(\nu a.X)_{\#a} = X$.*

Proof. For the first part, we prove two set inclusions:

- *Proof that $X \subseteq \nu a.(X_{\#a})$.* Suppose $x \in X$. Choose some fresh b (so $b\#x, X$). By Theorem 2.13 $(b\ a) \cdot x \in X$ and furthermore by Theorem 2.11 $(b\ a) \cdot x \in X_{\#a}$. It follows from Definition 5.4 that $x \in \nu a.(X_{\#a})$.
- *Proof that $\nu a.(X_{\#a}) \subseteq X$.* Suppose $x \in \nu a.(X_{\#a})$. So $x = \pi' \cdot x'$ for some $\pi' \in fix(supp(X_{\#a}) \setminus \{a\})$ and some $x' \in X_{\#a}$. By Theorem 2.18 $supp(X_{\#a}) \setminus \{a\} \subseteq supp(X)$. Therefore $\pi' \in fix(supp(X))$. Now $x' \in X$, so by Theorem 2.13 $\pi' \cdot x' \in X$.

For the second part, again we prove two set inclusions:

- *Proof that $X \subseteq (\nu a.X)_{\#a}$.* Suppose $x \in X$ (so $a\#x$). Then $x \in \nu a.X$ and it is immediate that $x \in (\nu a.X)_{\#a}$.
- *Proof that $(\nu a.X)_{\#a} \subseteq X$.* Suppose $x \in (\nu a.X)_{\#a}$. So $x = \pi' \cdot x'$ for some $\pi' \in fix(supp(X) \setminus \{a\})$ and some $x' \in X$ (so $a\#x'$). Now choose some entirely fresh b (so $b\#X, x, a, \pi'$) and write $\pi = (b\ a) \circ \pi' \circ (b\ a)$. It is a fact that $\pi \in fix(supp(X) \cup \{a\})$, so by Theorem 2.13 $\pi \cdot X = X$. Since $a\#x'$, it is also a fact that $\pi|_{supp(x')} = \pi'|_{supp(x')}$. By Theorem 2.13 $\pi' \cdot x' = \pi \cdot x'$. It follows that $x \in X$, and by Theorem 2.11 it also follows that $a\#x$.

□

Proposition 5.7. *Suppose $f \in X \to Y$ is a function-set. Then $\pi \cdot f$ is a function-set in $\pi \cdot X \to \pi \cdot Y$, and it represents the function*

$$\lambda x \in \pi \cdot X . \pi \cdot (f(\pi^{-1} \cdot x)).$$

This is the **conjugation action***.*

Proof. From equivariance (Theorem 2.15). □

Definition 5.8. Suppose $f : X \longrightarrow Y \in \mathsf{FMSet}_{\#\mathsf{a}}$. Define $\nu a.f$ as follows: if $x \in \nu a.X$ then we set

$$\text{И}b.\ (\nu a.f)(x) = (b\ a)\cdot f((b\ a)\cdot x).$$

We read $\nu a.f$ as **restrict** a **in** f.

Lemma 5.9. *If* $f : X \longrightarrow Y \in \mathsf{FMSet}_{\#\mathsf{a}}$ *then:*

- $\nu a.f$ *is well-defined (the choice of fresh b does not matter).*
- $\nu a.f : \nu a.X \longrightarrow \nu a.Y \in \mathsf{FMSet}_{\nu\mathsf{a}}$.
- $supp(\nu a.f) \subseteq supp(f) \setminus \{a\}$.

Proof. First, we review what has to be proved.

Suppose $x \in \nu a.X$. Choose some fresh b (so $b\#f, a, x$).

By Theorem 4.3 ($\exists$ form), to calculate $\nu a.f$ it suffices to calculate $(b\ a)\cdot f((b\ a)\cdot x)$. It is a fact that if $b\#(b\ a)\cdot f((b\ a)\cdot x)$ then this result does not depend on the choice of fresh b.

We must also check that $supp(\nu a.f)$ is finite, and $a\#\nu a.f$; there is no need for a separate proof of this, since it is subsumed by a proof that $supp(\nu a.f) \subseteq supp(f) \setminus \{a\}$.

We sketch each part of the proof in turn:

- $(b\ a)\cdot f((b\ a)\cdot x)$ *well-defined.*

  $b\#(b\ a)\cdot f((b\ a)\cdot x)$ is immediate because we assumed that $f \in \mathsf{FMSet}_{\#\mathsf{a}}$.

  What is slightly non-trivial is to prove that if $x \in \nu a.X$ then $(b\ a)\cdot x \in X$. Since $x \in \nu a.X$ and $X \in FMSet_{\#a}$, there exists some $\pi' \in fix(supp(X) \setminus \{a\})$ and some $x' \in X$ such that $x = \pi'\cdot x'$ and $a\#x'$. We reason as follows:

  $$(b\ a)\cdot x \overset{\text{fact}}{=} (b\ a)\cdot\pi'\cdot x' \overset{\text{Lem. 2.8}}{=} ((b\ a)\circ\pi'\circ(b\ a))\cdot(b\ a)\cdot x' \overset{a,b\#x',\text{Thm. 2.13}}{=} ((b\ a)\circ\pi'\circ(b\ a))\cdot x'$$

  It is a fact that $((b\ a)\circ\pi'\circ(b\ a)) \in fix(supp(X))$ and it follows by Theorems 2.15 and 2.13 that $(b\ a)\cdot x \in X$.

- $b\#(b\ a)\cdot f((b\ a)\cdot x)$. Using Theorem 2.11.

- $b\#(b\ a)\cdot f((b\ a)\cdot x) \in \nu a.Y$. It is a fact that $f((b\ a)\cdot x) \in Y$. The result follows by the definition of $\nu a.Y$.

We now prove that $supp(\nu a.f) \subseteq supp(f) \setminus \{a\}$. By Theorem 2.18 $supp(\nu a.f) \subseteq supp(f) \cup \{a\}$. By Theorems 4.4 and 4.3 ($\exists$ form) it then suffices to check that $(a'\ a)\cdot(\nu a.f) = \nu a.f$ for some fresh a' (so $a'\#f, a$). Choose some $x \in \nu a.X$ and some fresh b. We reason as follows:

$$\begin{aligned}
((a'\ a)\cdot \nu a.f)(x) &= (b\ a)\cdot((a'\ a)\cdot f)((b\ a)\cdot(a'\ a)\cdot x) && \text{Theorem 4.3 } (\forall \text{ form})\\
&= (b\ a)\cdot(a'\ a)\cdot f((b\ a)\cdot x) && \text{Proposition 5.7 and Lemma 2.8}\\
&= (b\ a)\cdot f((b\ a)\cdot x). && \text{Theorems 2.18 and 2.11}\\
&= (\nu a.f)(x) && \text{Theorem 4.3 } (\exists \text{ form})
\end{aligned}$$

$\square$

5.3 Statement and proof of the main result

Lemma 5.10. *The following data specifies a pair of functors between* $\mathsf{FMSet}_{\#a}$ *and* $\mathsf{FMSet}_{\nu a}$*:*

- $\text{-}_{\#a} : \mathsf{FMSet}_{\nu a} \longrightarrow \mathsf{FMSet}_{\#a}$ *maps* X *to* $X_{\#a}$ *and* $f : X \longrightarrow Y$ *to* $f_{\#a} : X_{\#a} \to Y_{\#a}$*.*
- $\text{-}_{\nu a} : \mathsf{FMSet}_{\#a} \longrightarrow \mathsf{FMSet}_{\nu a}$ *maps* X *to* $\nu a.X$ *and* $f : X \longrightarrow Y$ *to* $\nu a.f : \nu a.X \to \nu a.Y$*.*

Proof. By routine calculations. □

Theorem 5.11. $\text{-}_{\#a}$ *and* $\text{-}_{\nu a}$ *define an isomorphism of categories between* $\mathsf{FMSet}_{\#a}$ *and* $\mathsf{FMSet}_{\nu a}$*.*

Proof. That the functors are inverse on objects follows quickly from Lemma 5.6.

We check that these functors are inverse on arrows. Suppose $f : X \longrightarrow Y \in \mathsf{FMSet}_{\nu a}$. So $f \in X \to Y$ and $a\#f,X,Y$. We must check that $\nu a.(f_{\#a}) = f$. Take any $x \in X$ and choose some entirely fresh b. We reason as follows:

$$\begin{aligned}
(\nu a.(f_{\#a}))(x) &= (b\ a)\cdot f((b\ a)\cdot x) && \text{Definition 5.4, Theorem 4.3 } (\forall \text{ form}) \\
&= f((b\ a)\cdot(b\ a)\cdot x) && \text{Theorems 2.15 and 2.13} \\
&= f(x) && \text{Lemma 2.8.}
\end{aligned}$$

Suppose $f : X \longrightarrow Y \in \mathsf{FMSet}_{\#a}$. So $f \in X_{\#a} \to Y_{\#a}$. We must check that $(\nu a.f)_{\#a} = f$. Take any $x \in X_{\#a}$ and choose some entirely fresh b. We reason as follows:

$$\begin{aligned}
(\nu a.f)_{\#a}(x) &= (b\ a)\cdot f((b\ a)\cdot x) && \text{Definition 5.4} \\
&= (b\ a)\cdot f(x) && \text{Theorem 2.13} \\
&= f(x) && \text{Theorems 2.18 and 2.13.} \quad \square
\end{aligned}$$

6 Conclusions and related work

It is known that the category of nominal sets admits a representation as pullback-preserving presheafs, but abstract categorical presentations of the properties of this 'nominal' category that makes it 'nominal', have been lacking. There has been quite a lot of interest, especially recently, in more abstract accounts of what 'nominal' really is.

As far as I know Menni was the first to think about this, for the И-quantifier [20]. Pitts and Clouston are developing a notion of 'FM category' [4]. Kurz and Petrisan are pursuing not dissimilar ideas, coming (speaking very roughly) from the point of view of many-sorted logic and cylindric algebra [18]. Fiore and Hur are developing their own categorical framework [7], of which aspects of nominal techniques are in a certain sense they make formal a special case. There are, of course, other models of of names at a very abstract semantic level; examples include [3] (not strictly speaking categorical) and [15].

Here, this paper could be timely and the observations in it could be of some use; to suggest categorical equivalences to look for in the authors' respective environments, or indeed directly as a property of the category of FM sets FMSet.

Let me suggest an alternative reading of the results in this paper. $\mathsf{FMSet}_{\#a}$ and $\mathsf{FMSet}_{\nu a}$ are both categories with 'an atom missing'. In $\mathsf{FMSet}_{\#a}$, the atom is missing because it is fresh for individual data. In $\mathsf{FMSet}_{\nu a}$, the atom is missing because it is fresh for the objects and arrows. Let us shift our point of view and consider a reading of FMSet as a version of $\mathsf{FMSet}_{\#a}$ and $\mathsf{FMSet}_{\nu a}$ *with an extra atom put in*. In that sense, the two notions of freshness considered in this paper correspond to two notions of 'add a fresh atom': if we start from $\mathsf{FMSet}_{\#\mathsf{a}}$ and add an atom to the underlying data, we get FMSet; and if we start from $\mathsf{FMSet}_{\nu\mathsf{a}}$ and give the language the power to resolve a, we also get FMSet. The main result of this paper is that these two starting-points are isomorphic.

The equivalence of $\mathsf{FMSet}_{\#\mathsf{a}}$ and $\mathsf{FMSet}_{\nu\mathsf{a}}$ seems an elegant result. Up to a categorical isomorphism, there is only one way to add/subtract an atom.

References

[1] Nicolas Bourbaki, *Théorie des ensembles*, Hermann, Paris, 1970.

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

[3] Anna Bucalo, Furio Honsell, Marino Miculan, Ivan Scagnetto, and Martin Hofmann, *Consistency of the theory of contexts*, Journal of Functional Programming **16** (2006), no. 3, 327–395.

[4] Ranald Clouston, *Equational logic for names and binding*, Ph.D. thesis, University of Cambridge, UK, 2010, Pending graduation.

[5] Nicolaas G. de Bruijn, *Lambda calculus notation with nameless dummies, a tool for automatic formula manipulation, with application to the Church-Rosser theorem*, Indagationes Mathematicae **5** (1972), no. 34, 381–392.

[6] Maribel Fernández and Murdoch J. Gabbay, *Nominal rewriting with name generation: abstraction vs. locality*, Proceedings of the 7th ACM SIGPLAN International Symposium on Principles and Practice of Declarative Programming (PPDP 2005), ACM Press, 2005, pp. 47–58.

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

[8] Marcelo P. Fiore, Gordon D. Plotkin, and Daniele Turi, *Abstract syntax and variable binding*, Proceedings of the 14th IEEE Symposium on Logic in Computer Science (LICS 1999), IEEE Computer Society Press, 1999, pp. 193–202.

[9] Murdoch J. Gabbay, *A Theory of Inductive Definitions with alpha-Equivalence*, Ph.D. thesis, University of Cambridge, UK, 2000.

[10] ———, *A study of substitution, using nominal techniques and Fraenkel-Mostowski sets*, Theoretical Computer Science **410** (2009), no. 12-13, 1159–1189.

[11] ———, *The mathematical foundations of nominal techniques*, (2010), Accepted subject to revision.

[12] Murdoch J. Gabbay and Martin Hofmann, *Nominal renaming sets*, Proceedings of the 15th International Conference on Logic for Programming, Artificial Intelligence, and Reasoning (LPAR 2008), Springer, 2008, pp. 158–173.

[13] Murdoch J. Gabbay and Andrew M. Pitts, *A New Approach to Abstract Syntax Involving Binders*, 14th Annual Symposium on Logic in Computer Science, IEEE Computer Society Press, 1999, pp. 214–224.

[14] ———, *A New Approach to Abstract Syntax with Variable Binding*, Formal Aspects of Computing **13** (2001), no. 3–5, 341–363.

[15] Martin Hofmann, *Semantical analysis of higher-order abstract syntax*, Proceedings of the 14th Annual Symposium on Logic in Computer Science (LICS 1999), IEEE Computer Society Press, 1999, pp. 204–213.

[16] Isabelle/HOL/Nominal, *Lemma* `supp_of_fin_sets`, 2009, http://isabelle.in.tum.de/repos/isabelle/file/Isabelle2009-1/src/HOL/Nominal/Nominal.thy#l2331.

[17] Thomas Jech, *Set theory*, Springer, 2006, Third edition.

[18] Alexander Kurz and Daniela Petrisan, *On universal algebra over nominal sets*, Mathematical Structures in Computer Science **20** (2010), 285–318.

[19] James McKinna and Randy Pollack, *Pure Type Systems formalized*, Proceedings of the International Conference on Typed Lambda Calculi and Applications (TLCA 1993), LNCS, no. 664, Springer-Verlag, March 1993, pp. 289–305.

[20] Matias Menni, *About И-quantifiers*, Applied Categorical Structures **11** (2003), no. 5, 421–445.

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

[22] Frank Pfenning and Conal Elliott, *Higher-order abstract syntax*, PLDI (Programming Language Design and Implementation), ACM Press, 1988, pp. 199–208.

[23] Andrew M. Pitts, *Nominal logic, a first order theory of names and binding*, Information and Computation **186** (2003), no. 2, 165–193.

[24] ______, *Nominal system T*, Proceedings of the 37th ACM SIGACT-SIGPLAN Symposium on Principles of Programming Languages (POPL 2010), ACM Press, January 2010, pp. 159–170.

[25] Dugald Macpherson Richard Kaye (ed.), *Automorphisms of first-order structures*, Oxford University Press, 1994.

[26] Mark R. Shinwell, *The fresh approach: Functional programming with names and binders*, Ph.D. thesis, Computer Laboratory, University of Cambridge, December 2004.

[27] Joseph Shoenfield, *Mathematical logic*, Addison-Wesley, 1967.

[28] Christian Urban, *Nominal reasoning techniques in Isabelle/HOL*, Journal of Automatic Reasoning **40** (2008), no. 4, 327–356.