A GENERAL MATHEMATICS OF NAMES

MURDOCH J. GABBAY
HERIOT-WATT UNIVERSITY, EDINBURGH
MURDOCH.GABBAY@GMAIL.COM

Abstract. We introduce FMG (Fraenkel-Mostowski Generalised) set theory, a generalisation of FM set theory which allows binding of infinitely many names instead of just finitely many names. We apply this generalisation to show how three presentations of syntax — de Bruijn indices, FM sets, and name-carrying syntax — have a relation generalising to all sets and not only sets of syntax trees. We also give syntax-free accounts of Barendregt representatives, scope extrusion, and other phenomena associated to α-equivalence.

Our presentation uses a novel presentation based not on a theory but on a concrete model $\mathcal{U}$.

Contents

Date: July 18, 2006.

I would like to thank Henk Barendregt, Antonino Salibra, and an anonymous referee. This work was done while at the École Polytechnique and INRIA in Paris, France.

1. Introduction

1.1. **Overview.** This paper is about names and binding in infinitary syntax as an abstract set-theoretic notion.

In previous work (Gabbay 2000, Gabbay & Pitts 2001) the author and Pitts introduced **Fraenkel-Mostowski set theory** (**FM sets**). This is a set theory in the style of Zermelo-Fraenkel sets (**ZF sets** (Bell & Machover 1977, Johnstone 1987)) in which we can build abstract syntax as inductively defined sets of abstract syntax trees. It is shown how in FM, a set-operation corresponding to quotienting by α-equivalence (in case the set is of abstract syntax trees — *identical* to quotienting by α-equivalence) can be defined on the entire set-universe.

Call a variable-symbol in object-level syntax (that's the sets of abstract syntax trees we build in, say, a set theory), an **atom**. This is consistent with the original works (Gabbay 2000, Gabbay & Pitts 2001).

Just like syntax has a notion of 'variables occurring in' and 'variables occurring free in', so FM sets has a notion of 'atoms occurring in' and 'atoms occurring free in', which is valid for *all* sets, not just those which happen to be inductively defined. We call the 'free atoms' of a set x its **support** and write it $\mathsf{S}x$.

In the FM theory literature we have so far concentrated on support, swapping $(a\ b)\cdot x$ (a notion of 'rename a as b and vice versa' in the support), and abstractions $[a]x$ (a set-theoretic notion of α-equivalence). These definitions are abstract, that is they apply to all sets, and this opened up the path not only to the algebraic structure of Nominal Sets, but also Nominal Domains[1] and much other work. We give a Nominal version of FMG sets in §6.4.

In FM $\mathsf{S}x$ is always finite; a Fraenkel-Mostowski set has a **finite supporting set**. In fact, this is an axiom of the theory (axiom **(Fresh)** (Gabbay & Pitts 2001)). This made FM a semantics of *finitary* syntax — but without the syntax.

This is (at least in the opinion of the author) a fascinating and uniquely characteristic idea with rich structure. However, it does invite the question

> "but what about infinitary syntax, which can mention infinitely many different atoms?".

So read this paper, and enjoy our answer.

While we are at it, we take a step back from the original presentation of α-equivalence in set theory and give a much more general treatment of the mathematics, even in the case of FM (finitary syntax). Notably, we generalise the Gabbay-Pitts И quantifier, prove an important basic correctness result (a form of set-theoretic scope extrusion), and exploit our possibly infinite number of atoms to give interesting characterisation of 'name-carrying' and 'de Bruijn' syntax (de Bruijn 1972). We discover that, while FM is based on atoms, FMG is based on well-orderable streams of distinct atoms. And last but not least, we propose a slogan 'small=well-orderable' which generalises the 'small=finite' of FM sets.

1.2. **Method of presentation.** Presenting this work poses some unique problems. The mathematics is not simple (i.e. it is technical, in the way that any set-theory can be), and while we would maintain that FM techniques are no harder than anything else in modern theoretical computer science, this work will not be familiar to the reader unless they have studied previous work on FM ((Gabbay 2000, Gabbay & Pitts 2001), and other publications).

Since we cannot be *simple* nor *familiar*, we do at least try to be *concrete*. Thus, we do not dive straight in to the language of FMG sets[2] and its axioms.

[1] Published since this paper was submitted for publication (Shinwell & Pitts 2005).

[2] First-order logic (Barwise 1977, van Dalen 2002) extended with a constant symbol $\mathbb{A}$ called **the set of atoms** and a binary relation $\in$ called **set inclusion**.

Instead, we start with $\mathcal{U}$ a simple but (almost) completely sufficiently expressive fragment of the **cumulative hierarchy model** ((Bell & Machover 1977, Johnstone 1987)) of FMG sets.This allows us to:

- Construct $\mathcal{U}$ and explain FMG ideas in a concrete structure, using natural language.
- Build $\mathcal{U}$ directly in standard ZF (we use any sufficiently large set to interpret $\mathbb{A}$).

In this paper, we begin by investigating all our significant constructions in a *particular cumulative hierarchy* $\mathcal{U}$, and then we generalise by stages until we are in full FMG.

The technical content of this paper, section by section, is as follows:

- §2: We define $\mathcal{U}$ as a cumulative hierarchy and investigate its basic properties.
- §3: We construct sets describing name-carrying, de Bruijn (de Bruijn 1972), and FM syntax (that is, syntax-up-to-alpha-equivalence, in a sense which has been made formal (Gabbay & Pitts 2001) and will be made formal later in this paper) over a fixed but arbitrary signature E.
- §4: We show they are all isomorphic.
- §5: We show that the isomorphisms generalise to all elements of $\mathcal{U}$ (not just those which happen to represent abstract syntax trees for E).
- §6: We replace $\mathcal{U}$ with full FMG sets. The constructions of the concrete model $\mathcal{U}$ generalise and we show how.

Our voyage is therefore from what the reader most probably knows very well, namely operations on concrete syntax trees as occurs everywhere in the literature, to logical properties of a concrete cumulative set hierarchy, to logical provability in a first-order theory of FMG sets.

1.3. **Mathematical overview of FMG vs FM.** So just how does FMG differ from FM sets?

An axiom of FM states that every set has finite support, so in a suitable formal sense, we can say that a set of atoms is 'small' when it is finite. If ϕ is a predicate on atoms, we write $\mathsf{N}a.\phi(a)$ for 'ϕ is true for all but a small (finite) set of atoms'.

The slogan of FMG sets is 'small'='well-orderable'. Thus in FMG sets a set of atoms is 'small' when there is a set which is a well-ordering of that set of atoms. If ϕ is a predicate $\mathsf{N}a.\phi(a)$ means 'there exists a set of atoms S and a set which is a well-ordering of S, such that for all $a \notin S$, $\phi(a)$ holds'.

FMG in its 'vanilla' form makes no committment to just how large the largest well-orderable set of atoms is; but we can add an axiom saying 'well-orderable' means 'finite', or 'countably infinite', or whatever we please. Thus, an observation of this paper is that the only property of finite sets which we really *needed* in FM, was that they are well-orderable.

The technical reason well-orderability is so useful, is simply that it is a convenient way of guaranteeing that for any two small sets of atoms, one of them injects into the other (from the property of ordinals that for any two ordinals treated as ordered structures, one of them is an initial segment of the other (Johnstone 1987, Lemma 6.3)). This guarantees that there are enough functions between small sets of atoms that we can always rename them to be fresh, and as we shall see this will be very useful.

$\mathcal{U}$ the concrete model where we start, takes 'small' to equal 'countable' (and 'large' to equal 'uncountable').

2. The universe $\mathcal{U}$

2.1. **We build $\mathcal{U}$.** Fix an uncountable **set of atoms** $a, b, c, \ldots \in \mathbb{A}$.

Call a **permutation** π a bijection on $\mathbb{A}$. Define

$$\mathsf{S}\pi \stackrel{\text{def}}{=} \{a \in \mathbb{A} \mid \pi(a) \neq a\}. \tag{1}$$

Say π has **countable support** when $\mathsf{S}\pi$ is countable. Write $P_{\mathbb{A}}$ for the set of permutations π with countable support, which is a group under functional composition $\circ$. Write **Id** for the identity element which is the identity function on atoms.

If $S \subseteq \mathbb{A}$ write

$$\mathsf{Fix}(S) \stackrel{\text{def}}{=} \{\pi \in P_{\mathbb{A}} \mid \forall a \in S.\ \pi(a) = a\}.$$

We may drop the brackets and **write** $\mathsf{Fix}S$ **for** $\mathsf{Fix}(S)$ without comment. Say $\pi \in \mathsf{Fix}S$ **fixes** S **pointwise**.

Lemma 2.1. (1) *$\mathsf{S}\pi$ is the least S such that $\pi \in \mathsf{Fix}(\mathbb{A} \setminus S)$.*
(2) *$\pi \in \mathsf{Fix}S$ if and only if $\mathsf{S}\pi \cap S = \emptyset$.*
(3) *$\mathsf{Fix}S$ is a group.*

The proofs are by basic group-theoretic calculations.

Write $\mathcal{P}_{\leq\omega}(X)$ for the set of countable subsets of X.

We inductively define $\mathcal{U}$ a set of **elements** $u, v, \ldots$, a **permutation action** $P_{\mathbb{A}} \to \mathcal{U} \to \mathcal{U}$ which we write $\pi \cdot u$, and a function **support** $\mathcal{U} \to \mathcal{P}_{\leq\omega}(\mathbb{A})$ which we write $\mathsf{S}u$. (The notation clash with $\mathsf{S}\pi$ is deliberate; the values of the two functions will turn out to be identical.)

(1) $a \in \mathbb{A}$ is an element.

$$\pi \cdot a \stackrel{\text{def}}{=} \pi(a) \quad \text{and} \quad \mathsf{S}a \stackrel{\text{def}}{=} \{a\}.$$

(2) $U \in \mathcal{P}_{\leq\omega}(\mathcal{U})$ is an element.

$$\pi \cdot U \stackrel{\text{def}}{=} \{\pi \cdot u \mid u \in U\} \quad \text{and} \quad \mathsf{S}U \stackrel{\text{def}}{=} \bigcup\{\mathsf{S}u \mid u \in U\}.$$

Note that $\mathcal{U}$ can encode **tuples** as $\langle x, y\rangle = \{\{x, y\}, \{x\}\}$ (this is the standard set-theoretic encoding (Johnstone 1987)).

(3) If x and y are elements then **the abstraction**

$$[x]y \stackrel{\text{def}}{=} \{\langle \pi \cdot x, \pi \cdot y\rangle \mid \pi \in \mathsf{Fix}(\mathsf{S}y \setminus \mathsf{S}x)\} \tag{2}$$

is an element.

$$\pi \cdot [x]y \stackrel{\text{def}}{=} [\pi \cdot x]\pi \cdot y \quad \text{and} \quad \mathsf{S}[x]y \stackrel{\text{def}}{=} \mathsf{S}y \setminus \mathsf{S}x.$$

Write arbitrary elements of $\mathcal{U}$ as u, v, arbitrary atoms a, b, c, arbitrary countable sets U, V, S and arbitrary abstractions $\hat{u}, \hat{v}$.

In summary, any element of $\mathcal{U}$ has precisely one of the following three forms:

- an atom a,
- a countable collection U, or
- an abstraction $[x]y$.

Write

(1) $x\#y$ when $\mathsf{S}x \cap \mathsf{S}y = \emptyset$.
(2) $\pi\#x$ when $\mathsf{S}\pi \cap \mathsf{S}x = \emptyset$. Unfolding definitions, this is equivalent to $\pi \in \mathsf{Fix}(\mathsf{S}x)$.

For two distinct atoms a and b write $(a\ b)$ for the permutation mapping a to b, b to a, and $n \neq a, b$ to n. This is a **swapping**.

Then $(a\ b)\#a$ is false and $(a\ b)\#c$ is true, $(a\ b)\#\langle a, c\rangle$ is false (we assume a, b, c are all distinct). $(a\ b)\#[a]a$ is true, but $(a\ b)\#[a]b$ is false. $a\#[a]a$ and $a\#[a]b$ are both true; $b\#\langle [a]a, [a]b\rangle$ is false.

2.2. **Support and permutation acting on elements of $\mathcal{U}$.** We prove some simple lemmas:

Lemma 2.2. *Suppose S is a countable set of atoms. Then:*
(1) $\pi \cdot S = \{\pi \cdot a \mid a \in S\}$.
(2) $\mathsf{S}S = S$.

Proof. (1) S is a countable set, and the permutation action is defined to be pointwise.
(2) S is a countable set, so $\mathsf{S}S = \bigcup\{\mathsf{S}a \mid a \in S\}$. We now observe that by definition $\mathsf{S}a = \{a\}$.

□

Lemma 2.3. (1) *If $x \in \mathcal{U}$ then $\mathsf{S}x \in \mathcal{U}$.*
(2) $\pi \cdot (\mathsf{S}x) = \mathsf{S}(\pi \cdot x)$.
(3) $x\#[x]y$.
(4) *If $\pi\#x$ then $\pi \cdot x = x$ ($\pi\#x$ is logically equivalent to $\pi \in \mathsf{Fix}\mathsf{S}x$).*
(5) *If $\pi(a) = \pi'(a)$ for all $a \in \mathsf{S}x$ then $\pi \cdot x = \pi' \cdot x$.*

Proof. (1) Proof by induction on x; since $a \in \mathcal{U}$ always it suffices to show that $\mathsf{S}x$ is countable.
- $\{a\}$ is countable.
- Any countable union of countable sets is necessarily countable.
- Any subset of a countable set is necessarily countable.

(2) Proof by induction on x.
- $\pi \cdot \{a\} = \{\pi(a)\}$.
- $\pi \cdot \bigcup \left\{\mathsf{S}u \;\middle|\; u \in U\right\} = \bigcup \left\{\pi \cdot \mathsf{S}u \;\middle|\; u \in U\right\}$ and we can use the inductive hypothesis.
- $\pi \cdot (\mathsf{S}y \backslash \mathsf{S}x) = (\pi \cdot \mathsf{S}y) \backslash (\pi \cdot \mathsf{S}x)$ and again we use the inductive hypothesis.

(3) Unpacking definitions.
(4) By induction on x.
- $\pi\#a$ when $\pi(a) = a$ and we recall that $\pi \cdot a = \pi(a)$.
- $\pi\#U$ when $\pi\#u$ for each $u \in U$ and we use the inductive hypothesis.
- We observe that $\pi \cdot [x]y = [\pi \cdot x]\pi \cdot y$ by definition. Now suppose $\pi \in \mathsf{Fix}(\mathsf{S}y \setminus \mathsf{S}x)$.
  We note that $\mathsf{S}(\pi \cdot y) \setminus \mathsf{S}(\pi \cdot x) = \pi \cdot (\mathsf{S}y \setminus \mathsf{S}x) = \mathsf{S}y \setminus \mathsf{S}x$, and also recall that $\mathsf{Fix}(\mathsf{S}y \setminus \mathsf{S}x)$ is a group. By group-theoretic calculations it follows that:

$$\pi \cdot [x]y = \left\{\langle \pi' \cdot (\pi \cdot x), \pi' \cdot (\pi \cdot y)\rangle \;\middle|\; \pi' \in \mathsf{Fix}(\mathsf{S}(\pi \cdot y) \setminus \mathsf{S}(\pi \cdot x))\right\}$$
$$= \left\{\langle \pi'' \cdot x, \pi'' \cdot y\rangle \;\middle|\; \pi'' \in \mathsf{Fix}(\mathsf{S}y \setminus \mathsf{S}x)\right\} = [x]y.$$

  Here π'' is 'morally' $\pi' \circ \pi$.

(5) This is a corollary of the last part, observing that $\pi(a) = \pi'(a)$ for all $a \in \mathsf{S}x$ precisely when $\pi^{-1} \circ \pi' \in \mathsf{Fix}(\mathsf{S}x)$, and this is logically equivalent to asserting $\pi^{-1} \circ \pi'\#x$.

□

Note that $\pi \cdot x = x$ does not imply $\pi\#x$. For example $\mathsf{S}(a\ b) = \{a, b\}$ and $\pi \cdot \{a, b\} = \{a, b\}$ ($(a\ b)$ is a swapping, defined at the end of §2.1).

$\pi \cdot ([x]y)$ is defined componentwise as $[\pi \cdot x]\pi \cdot y$. Also, $\pi \cdot U$ is defined pointwise. Is it possible for the componentwise action on $[x]y$ to conflict with the pointwise action on it as some U?

In fact this question does not arise since $[x]y$ is uncountable and according to our definitions $U \in \mathcal{U}$ must be countable.

However, it is easy to prove that the two collections are equal anyway:

Lemma 2.4. *The componentwise and pointwise actions agree for abstractions:*

$$[\pi \cdot x]\pi \cdot y = \left\{ \langle \pi \cdot \pi' \cdot x, \pi \cdot \pi' \cdot y \rangle \;\middle|\; \pi' \in \mathsf{Fix}(\mathsf{S}y \setminus \mathsf{S}x) \right\}.$$

Proof. We must show that for *any* π,

$$\left\{ \langle \pi' \cdot \pi \cdot x, \pi' \cdot \pi \cdot y \rangle \;\middle|\; \pi' \in \mathsf{Fix}(\mathsf{S}\pi \cdot y \setminus \mathsf{S}\pi \cdot x) \right\} = \left\{ \langle \pi \cdot \pi' \cdot x, \pi \cdot \pi' \cdot y \rangle \;\middle|\; \pi' \in \mathsf{Fix}(\mathsf{S}y \setminus \mathsf{S}x) \right\}.$$

We use the notation below and observe that $\pi' \cdot \pi = \pi \cdot {\pi'}^{\pi^{-1}}$, and $\pi' \in \mathsf{Fix}(\mathsf{S}\pi \cdot y \setminus \mathsf{S}\pi \cdot x)$ if and only if ${\pi'}^{\pi^{-1}} \in \mathsf{Fix}(\mathsf{S}y \setminus \mathsf{S}x)$. Finally we note, as we noted before, that $\pi' \in \mathsf{Fix}(\mathsf{S}y \setminus \mathsf{S}x)$ is a group. The result now follows by group-theoretic calculations. □

As a matter of notation write ${\pi'}^{\pi} = \pi \circ \pi' \circ \pi^{-1}$ (the **conjugation** of π' by π (Fraleigh 1994)). It is easy to calculate that π' maps a to b if and only if ${\pi'}^{\pi}$ maps $\pi(a)$ to $\pi(b)$; in particular, $\pi' \in \mathsf{Fix}S$ if and only if ${\pi'}^{\pi} \in \mathsf{Fix}(\pi \cdot S)$.

For example, if a, b, f and x are atoms then $[a]a = [b]b$, $[a]b = [x]b$, and $[f][x]\langle f, x\rangle = [x][f]\langle x, f\rangle$. These reprise well-known α-equivalences in (say) the λ-calculus and logic (but without the λ-calculus and logic; equality on abstractions is precisely α-equivalence, in a suitable formal sense which we analyse in depth in the rest of this paper).

2.3. More complex elements of $\mathcal{U}$.

Just that $\mathcal{U}$ is closed under countable subsets already gives it considerable power. For example it contains:

(1) The natural numbers $\mathbb{N}$ encoded as $0 \stackrel{\text{def}}{=} \emptyset$, and $i+1 \stackrel{\text{def}}{=} \{i, \{i\}\}$.
(2) A two-element set $\mathbb{B}$ encoded as $\{0, 1\}$.
(3) Ordered pairs (as we already noted)

$$\langle x, y\rangle \stackrel{\text{def}}{=} \{\{x\}, \{x, y\}\}.$$

Given $X, Y \subseteq \mathcal{U}$ (X and Y need not necessarily be countable, and not necessarily be elements of $\mathcal{U}$) write

$$X \times Y \stackrel{\text{def}}{=} \{\langle x, y\rangle \mid x \in X,\ y \in Y\}.$$

This is of course the standard notion of **cartesian product**.
(4) Disjoint sums

$$\mathbf{Inl}(x) \stackrel{\text{def}}{=} \langle 0, x\rangle \qquad \text{and} \qquad \mathbf{Inr}(y) \stackrel{\text{def}}{=} \langle 1, y\rangle.$$

Given $X, Y \subseteq \mathcal{U}$ write

$$X + Y \stackrel{\text{def}}{=} \{\mathbf{Inl}(x) \mid x \in X\} \cup \{\mathbf{Inr}(y) \mid y \in Y\}.$$

(5) Finite lists $[u_1, \ldots, u_n]$ encoded as nested pairs. Given $X \subseteq \mathcal{U}$ write X-list for the set of lists with elements in X.
(6) Countable streams $p = [p_1, p_2, \ldots]$ encoded as countable sets of finite lists of initial segments. Given $X \subseteq \mathcal{U}$ write X-stream for the collection of countable streams of elements in X.

The details of the encodings are irrelevant; it is only important that they exist.

Call an arbitrary subset $W \subseteq \mathcal{U}$ a **class** or a **collection** (as we did above). This need not be an element of $\mathcal{U}$, and if we know for a fact that it is not, call it a **proper class**. By construction of $\mathcal{U}$, a class is a proper class precisely when it is uncountable and not equal to $[x]y$ for some $x, y \in \mathcal{U}$.

If u is a finite list or a stream, we may write $hd(u)$ for u_1 and $tl(u)$ for $[u_2, u_3, \ldots]$, where these are defined. (We mostly use this notation much later on in this paper, to the particular case of a stream of *distinct* atoms $p \in \mathbb{L}$ which we define below.)

We briefly consider the permutation action and support of these constructs:

Lemma 2.5. *For all the constructs above, permutation acts componentwise and the support of the whole is the set union of the supports of the components. For example,* $\pi \cdot i = i$ *always and* $\mathsf{S}i = \emptyset$*, and*

$$\pi \cdot [u_1, u_2, \ldots] = [\pi \cdot u_1, \pi \cdot u_2, \ldots] \qquad \textit{and} \qquad \mathsf{S}[u_1, u_2, \ldots] = \bigcup \mathsf{S}u_i,$$

and similarly for pairs and finite lists.

Proof. For numbers, we work by induction on their construction, recalling that π acts pointwise on sets (so $\pi \cdot \emptyset = \emptyset$, for example) and $\mathsf{S}U$ takes the union of $\mathsf{S}u$ for all $u \in U$. These observations, along with some concrete calculations, prove the rest of the result for tuples, finite lists, and finally for streams. □

Proper classes with respect to $\mathcal{U}$ which are of particular interest are

(1) $\mathbb{A}$ the set of atoms.
(2) $\mathbb{L}$, which we now define.

Write $p, q \in \mathbb{L}$ for the set of countable streams of *distinct* atoms, that is, streams $p = [p_1, p_2, \ldots]$ such that $p_i = p_j$ implies $i = j$ and $p_i \in \mathbb{A}$ for all i.

Lemma 2.6. *Suppose* $p = [p_1, p_2, \ldots] \in \mathbb{L}$*. Then:*

(1) $p \in \mathcal{U}$.
(2) *The permutation action is pointwise on the* p_i*, that is,* $\pi \cdot p = [\pi \cdot p_1, \pi \cdot p_2, \ldots]$.
(3) $\mathsf{S}p = \{p_1, p_2, \ldots\}$.

Also, $\mathbb{L}$ *is a proper class* ($\mathbb{L} \notin \mathcal{U}$).

Proof. For the first parts, the proofs are as for Lemma 2.5. $\mathbb{L}$ is a proper class since it is uncountable (and, we can check it is not an abstraction $[x]y$ for any elements x and y). □

Note that for $p \in \mathbb{L}$, $\mathsf{S}p$ is infinite but countable.[3] Two uses of p are: to help model infinite behaviour in the presence of name-generation (so p might be a countable list of 'generated' names); and to help model of infinitary syntax.

$p \in \mathbb{L}$ behaves like a 'big atom'. For example $\mathbb{L}$ has a kind of permutation action:

Lemma 2.7. (1) *Given* $p = [p_1, \ldots], q = [q_1, \ldots] \in \mathbb{L}$*, if* $p\#q$ *then* $(p\ q)$ *is well-defined, where* $(p\ q)(p_i) = q_i$ *and* $(p\ q)(q_i) = p_i$ *and otherwise* $(p\ q)(c) = c$.
(2) *For* p *and* q *as in the first part, if* $p\#q$ *does not hold then* $(p\ q)$ *is not necessarily well-defined.*

Proof. (1) We just defined it.
(2) Consider p and $q = tl(p)$. Then p_1 maps to p_2, and p_2 maps to both p_1 and p_3.

□

2.4. **The И quantifier, properties of $\mathbb{L}$, abstraction and concretion.** If $S \subseteq \mathbb{A}$ is countable it follows that $\mathbb{A} \setminus S$ is *un*countable, and we can choose a $p \in \mathbb{L}$ such that $p\#S$. We say p **is fresh for** S. We use this without comment henceforth.

We may also just say '**choose a fresh** p', meaning that p should be fresh for the union of the supports of whatever elements are under consideration at that particular moment. This would be meaningless if we intend p to be fresh for a proper class of $\mathcal{U}$, but we will never do this.

If $a \in \mathbb{A}$ then $\mathsf{S}a$ is a singleton, which is *very* countable. Thus we can always choose a fresh $a \in \mathbb{A}$.

[3]...so it can never exist in FM sets, where everything has finite support.

- Given some $F : \mathbb{A} \to \mathcal{U}$ write $\text{И}a.Fa$ for the unique value of F at all but countably many $a \in \mathbb{A}$, if this exists.
- Similarly for $F : \mathbb{L} \to \mathcal{U}$ write $\text{И}p.Fp$ for the unique value of F at all $p\#S$ for some countable $S \subseteq \mathbb{A}$, if this exists.
- Given some ϕ a predicate on $\mathbb{A}$ write '$\text{И}a.\phi(a)$ is true' (or just '$\text{И}a.\phi(a)$') if ϕ is true for all atoms except for the elements of some countable $S \subseteq \mathbb{A}$, and '$\text{И}a.\phi(a)$ is false' if ϕ is false for all atoms except for the elements of some countable $S \subseteq \mathbb{A}$.
- Similarly for ϕ a predicate on $\mathbb{L}$, write '$\text{И}p.\phi$ is true' or just $\text{И}p.\phi$ when ϕ is true of all $p \in \mathbb{L}$ such that $p\#S$, for some countable $S \subseteq \mathbb{A}$, and so on.

The canonical examples are:

- $\text{И}a.a$ is not well-defined, but $\text{И}a.[a]a$ is well-defined and has value $[a]a = [b]b = [c]c = \ldots$.
- $\text{И}a.a = b$ is false, and $\text{И}a.a \neq b$ is true.
- Let V be a set containing uncountably many atoms, and *not* containing uncountably many atoms. Then $\text{И}a.a \in V$ is not defined.

In practice И*blah* is usually well-defined, because *blah* will be something *we* specified using a sentence in natural language or mathematical formalism, parameterised over some finite set of elements of $\mathcal{U}$ each of which has countable support; so *blah* should be either uniformly true or false away from that support. This informal argument becomes formal in FMG sets, both internally (abstractive functions in §5.3 and generalised И for them §5.4), and externally (equivariance Theorem 6.3).

The following result is the main reason that $\mathbb{A}$ and $\mathbb{L}$ are so important, we shall need it for later:

Lemma 2.8. (1) *$[a]u$ is the graph of a partial function defined precisely on those c such that $c = a$ or $c\#u$ (or equivalently; such that $c\#[a]u$), with value $(c\ a) \cdot u$ where defined.*

(2) *$[p]u$ is the graph of a partial function defined precisely on $q\#\mathsf{S}u \setminus \mathsf{S}p$ (or equivalently; such that $q\#[p]u$), with value $\text{И}r.(q\ r) \circ (r\ p) \cdot u$.*

Proof. We note that by definition, $c\#[a]u$ when $c\#u$ or $c = a$, and similarly $q\#[p]u$ when $q\#\mathsf{S}u \setminus \mathsf{S}p$.

(1) Expanding definitions,
$$[a]u = \left\{ \langle \pi(a), \pi \cdot u \rangle \;\middle|\; \pi \in \mathsf{Fix}(\mathsf{S}u \setminus \{a\}) \right\}.$$
The first component $\pi(a)$ can assume any $c\#u$ or $c = a$, since $(c\ a) \in \mathsf{Fix}(\mathsf{S}u \setminus \{a\})$ for these c.

It remains to show this is functional, i.e. $\pi \cdot u = \pi' \cdot u$ if $\pi(a) = \pi'(a)$ and $\pi, \pi' \in \mathsf{Fix}(\mathsf{S}u \setminus \{a\})$. But then π and π' agree on $\mathsf{S}u$ and the result follows from the last part of Lemma 2.3.

(2) Expanding definitions,
$$[p]u = \left\{ \langle \pi \cdot p, \pi \cdot u \rangle \;\middle|\; \pi \in \mathsf{Fix}(\mathsf{S}u \setminus \mathsf{S}p) \right\}.$$
First we show this is functional: suppose $\pi, \pi' \in \mathsf{Fix}(\mathsf{S}u \setminus \mathsf{S}p)$ and $\pi \cdot p = \pi' \cdot p$. By the technical lemma below and Lemma 2.6, $\pi(c) = \pi'(c)$ for every $c \in \mathsf{S}u$. Functionality follows.

We now just need to show that for every $q\#\mathsf{S}u \setminus \mathsf{S}p$, there is some $\pi \in \mathsf{Fix}(\mathsf{S}u \setminus \mathsf{S}p)$ such that $\pi \cdot p = q$. Choose some fresh r (taking $r\#p, q, u$ suffices) and set $\pi = (q\ r) \circ (r\ p)$.

□

Lemma 2.9 (Technical Lemma)**.** *Suppose $p = [p_1, \ldots] \in \mathbb{L}$. If $\pi \cdot p = \pi' \cdot p$ then $\pi(p_i) = \pi'(p_i)$ for every i.*

Proof. Lemma 2.6 proves the permutation action is pointwise. The result is immediate. □

We shall write abstractions as $\hat{u}$, $\hat{v}$. We write the **value** or **concretion** of $\hat{u}$ at some c or q, when this is defined, as $\hat{u}@c$ or $\hat{u}@q$ respectively.

The terminology 'concretion' is consistent with existing terminology in the FM literature (Gabbay & Pitts 2001, Definition 5.3).

Concretion is very useful for specifying functions, since it allows us to 'choose the name of the bound variable'. We will use it heavily later.

Lemma 2.10. (1) *$[c](\hat{u}@c) = \hat{u}$ when $\hat{u}$ is an abstraction by an atom and $\hat{u}@c$ is defined, and $([a]u)@a = u$.*

(2) *$[p](\hat{u}@p) = \hat{u}$ when $\hat{u}$ is an abstraction by an element of $\mathbb{L}$ and $\hat{u}@p$ is defined, and $([p]u)@p = u$.*

Proof. (1) Suppose $\hat{u} = [a]u$. By Lemma 2.8, $\hat{u}@c = (c\ a)\cdot u$ and it must be the case that $c\#\hat{u}$. By definition

$$[c](\hat{u}@c) = [c](c\ a)\cdot u = (c\ a)\cdot([c]u).$$

We then use the fourth part of Lemma 2.3, recalling that by definition $a\#[a]u$ and that $c\#[a]u$.

We also observe that $(a\ a)\cdot u = u$, so using the previous result $([a]u)@a = u$ follows.

(2) Suppose $\hat{u} = [p]u$. By Lemma 2.8, $\hat{u}@q = (q\ a)\cdot u$ and it must be the case that $q\#\hat{u}$. By definitions

$$[q](\hat{u}@q) = \text{И}r.[q](q\ r)\circ(r\ p)\cdot u = \text{И}r.(q\ r)\circ(r\ p)\cdot([p]u).$$

We then use the fourth part of Lemma 2.3.

□

3. Inductive datatypes

Recall from §2.3 the definitions of cartesian products $X\times Y$, disjoint sums $X+Y$, lists X-list, the set of natural numbers $\mathbb{N}$, the classes of atoms and countable streams of distinct atoms $\mathbb{A}$ and $\mathbb{L}$, and so on (for any classes $X, Y \subseteq \mathcal{U}$).

Also write

$$[X]Y \quad \text{for} \quad \{[x]y \mid x\in X,\ y\in y\}.$$

Fix a countably infinite collection of **class variable symbols** V, V', X, and so on. Then expressions of a simple **class specification grammar** are defined by:

(3) $$E ::= X, V, V' \ldots \mid E\times E \mid [E]E \mid \mathbb{N} \mid \mathbb{A} \mid E+E \mid E\text{-list} \mid \ldots$$

Say E is **closed** when it has no class variables. A closed E denotes an actual class $[\![E]\!] \subseteq \mathcal{U}$ in a natural way. For example:

- $[\![\mathbb{A}]\!] = \mathbb{A}$.
- $[\![\mathbb{N}]\!] = \mathbb{N}$, where $\mathbb{N}$ is defined in §2.3.
- $[\![E+E']\!] = [\![E]\!] + [\![E']\!]$, where $+$ is defined in §2.3.
- $[\![[E]E']\!] = [\ [\![E]\!]\ ][\![E']\!]$ (in this paper we shall only really consider $E \in \{\mathbb{A}, \mathbb{L}\}$).
- and so on.

Say E is a **signature in** X when it mentions at most one class variable, and that variable is X. We abuse notation and write EX for a signature in X. Given EX and $E'X$ write $EE'X$ for a signature obtained by substituting $E'X$ for X in EX. For example, $(X\times X)(X\text{-list})X = (X\text{-list})\times(X\text{-list})$.

Say permutation **acts trivially** on $u \in \mathcal{U}$ when $\pi \cdot u = u$ for all π. Given a class $X \subseteq \mathcal{U}$, which may be countable and thus itself an element of $\mathcal{U}$, say X has a **trivial** permutation action when permutation acts trivially on every element of X.

Lemma 3.1. *If $X \subseteq \mathcal{U}$ has a trivial permutation action then $[\mathbb{A}]X \cong X$ and $[\mathbb{L}]X \cong X$ is a canonical way.*

Proof. We consider just $[\mathbb{A}]X$; the case of $[\mathbb{L}]X$ is similar. Map $\hat{x}$ to $\hat{x}@a$ for any a, and x to $[a]x$ for any a. This is well-defined by Lemma 2.10 and the fact that permutation acts trivially on x and $\hat{x}@a$. □

We need some notation: if $S \subseteq \mathcal{U}$ say S is **equivariant** when

$$\forall x \in \mathcal{U}, \pi \in P_{\mathbb{A}}.\ x \in S \Leftrightarrow \pi \cdot x \in S.$$

Much more on this later in §5.1.

Lemma 3.2.
(1) *$\mathbb{A}$ and $\mathbb{N}$ are equivariant.*
(2) *If X and Y are equivariant, so are $X \times Y$, $X + Y$, $[X]Y$, and X-list (and so on).*

Proof.
(1) If $a \in \mathbb{A}$ then $\pi \cdot a \in \mathbb{A}$, since a permutation $\pi \in P_{\mathbb{A}}$ *does* map atoms to atoms. The result follows since $P_{\mathbb{A}}$ is also a group. If $n \in \mathbb{N}$ then the permutation action is trivial (that is, $\pi \cdot n = n$ for all π) and the result is easy to prove.
(2) All results are by simple calculations. We only give one. Suppose $x \in X$ and $y \in Y$. Then $\pi \cdot [x]y = [\pi \cdot x]\pi \cdot y$ by Lemma 2.4 and since $\pi \cdot x \in X$ and $\pi \cdot y \in Y$, we are done.

□

Equivariant classes are easier to work with — we know we can rename atoms without moving out of the class, see also Lemma 3.4 below — and sufficient to model syntax-with-binding as we are in the process of demonstrating. All classes are assumed equivariant unless stated otherwise.

For some equivariant $S \subseteq \mathcal{U}$ write ES for the class denoted by EX interpreted according to the list above, where we additionally take X to denote S.

Lemma 3.3. *A signature induces a class ιE (ι for 'initial') which is the least class such that $E(\iota E) \subseteq \iota E$.*

Proof. Write $E^0 X \stackrel{\text{def}}{=} X$ and $E^{i+1} X \stackrel{\text{def}}{=} E(EX)$. ιE can be constructed as $\bigcup_i E^i X$ in a standard way (see (Paulson 1994) or (Gabbay 2000, Section 10)). We briefly look at the only unusual aspect of this construction, which is FM(G) abstraction (it turns out to be quite easy):

Suppose $X_0 \subseteq X_1 \subseteq X_2 \subseteq \ldots \subseteq \mathcal{U}$. Then:

(1) $[\bigcup_i X_i]Y = \bigcup [X_i]Y$: Suppose $[x]y \in [\bigcup_i X_i]Y$. Then $x \in X_i$ for some i and $y \in Y$, so $x \in [X_i]Y$. Conversely if $[x]y \in \bigcup [X_i]Y$ then $[x]y \in [X_i]Y$ for some i so $x \in X_i$ and $[x]y \in [\bigcup_i X_i]Y$.
(2) $[Y]\bigcup_i X_i = \bigcup_i [Y]X_i$: Similarly.

□

Call ιE an **inductively defined datatype**. The expression determines its **constructors**.

Three signatures are of particular interest to us by way of prototypical examples:

$$L_{nc}X \stackrel{\text{def}}{=} \mathbb{A} + X \times X + \mathbb{A} \times X \qquad LX \stackrel{\text{def}}{=} \mathbb{A} + X \times X + [\mathbb{A}]X$$
$$L_{db}X \stackrel{\text{def}}{=} \mathbb{N} + \mathbb{A} + X \times X + X \tag{4}$$

ιL_{nc} is an inductive datatype of terms of the λ-calculus without α-equivalence. ιL is an FM datatype of terms up to α-equivalence $=_\alpha$; the lemma

$$\iota L \;\cong\; \iota L_{nc}/{=_\alpha}$$

is easy to prove by induction, but we leave that for α which we construct in the next section. ιL_{db} is a de Bruijn datatype for λ-terms in the style of the seminal implementation of abstract syntax up to α-equivalence in Automath (de Bruijn 1972) by de Bruijn and others.

The next section discusses the relationships between these signatures in detail.

We recently mentioned that equivariant classes are easier to work with. Here is another reason (known already for FM (Gabbay 2000, Gabbay & Pitts 2001)):

Lemma 3.4. (1) *If X is equivariant and $\hat{x} \in [\mathbb{A}]X$ then $\hat{x}@b \in X$ for any b, where $\hat{x}@b$ is well-defined.*
(2) *If X is not equivariant then there may be $\hat{x} \in [\mathbb{A}]$ such that $\hat{x}@b$ is well-defined and not in X.*
Similarly for $[\mathbb{L}]X$.

Proof. (1) If $\hat{x} \in [\mathbb{A}]X$ then by construction $\hat{x} = [a]x$ for some $a \in \mathbb{A}$ and $x \in X$. By Lemma 2.10 $\hat{x}@b = (b\ a) \cdot x$ where this is defined, and by equivariance this is in X.
(2) It suffices to provide a counterexample. Let $X = \{a\}$ and observe that $[a]a \in [\mathbb{A}]X$ and $[a]a@b = b$ is well-defined but $b \notin X$.
The case of $[\mathbb{L}]X$ is similar. □

Thus if we want to use classes as a sorting system for constructing abstract syntax trees (and we do) we had best use equivariant classes, since otherwise we cannot concrete an abstraction, i.e. 'choose a name for the abstracted variable name' at any atom for which the concretion is defined.

4. The relation between name-carrying-, FM-, and de Bruijn syntax

Call EX a **simple signature** when all its abstractions are of the form $[\mathbb{A}]$-. The examples of (4) are all simple.

Given a simple signature, for example $LX = \mathbb{A} + X \times X + [\mathbb{A}]X$, we derive two other signatures from it as follows:

(1) $E_{nc}X$ the **name-carrying** version is EX with every abstraction replaced by $\mathbb{A} \times$ -. For example $L_{nc}X = \mathbb{A} + X \times X + \mathbb{A} \times X$.
(2) $E_{db}X$ the **de Bruijn** version is EX with every abstraction replaced by -, and every instance of $\mathbb{A}$ replaced by $\mathbb{N} + \mathbb{A}$. For example $L_{db}X = \mathbb{N} + \mathbb{A} + X \times X + X$.

These give rise to three inductively defined sets of abstract syntax ιE_{nc}, ιE, and ιE_{db}.

We now define functions α and β

$$\iota E_{nc} \xrightarrow{\alpha} \iota E \qquad\qquad [\mathbb{L}]\iota E \xrightarrow{\beta} \iota E_{db} \tag{5}$$

as follows:

To define α, which is out of an inductively defined set, it suffices to define the action of α on its components. We let α be the identity *except* on components where EX has $[\mathbb{A}]X$ (so that E_{nc} has $\mathbb{A} \times X$), in which case α is the abstraction function $\lambda a, x.[a]x$.

For example $\alpha : \iota L_{nc} \to \iota L$ is defined as follows:

- $\alpha(a) \stackrel{\text{def}}{=} a$.
- $\alpha(tt') \stackrel{\text{def}}{=} \alpha(t)\alpha(t')$.

- $\alpha(\lambda a.t) \stackrel{\text{def}}{=} \lambda[a]\alpha(t)$ (of course the leftmost λ is a term-former of L_{nc} and the rightmost λ is a term-former of L).

We define

$$\beta t \quad \stackrel{\text{def}}{=} \quad \text{И} l.\beta'(l,t)$$

where β' is defined on $\mathbb{L} \times \iota E$ as follows:

- $\beta'(l,a) \stackrel{\text{def}}{=} i$ if $a = l_i$, and $\beta'(l,a) \stackrel{\text{def}}{=} a$ otherwise.
- $\beta'(l,tt') \stackrel{\text{def}}{=} \beta'(l,t)\beta'(l,t')$
- $\beta'(l,\lambda\hat{t}) \stackrel{\text{def}}{=} \text{И} a.\beta'(a :: l, \hat{t}@a)$.

It is easy to inductively characterise the kernel $\sim$ of α on ιL_{nc}:

(1) $a{\sim}a$.
(2) $s{\sim}t$ and $s'{\sim}t'$ implies $ss'{\sim}tt'$.
(3) $\text{И} c.(c\ a)\cdot t = \text{И} c.(c\ a')\cdot t'$ implies $\lambda a.t{\sim}\lambda a'.t'$.

A few examples suffice to convince us that the kernel of α is the relation which we would naturally call α-equivalence.

βx is defined as $\text{И} l.\beta'(l,x)$ where β' is defined as before. We also state what to do with countable powersets:

$$\beta'(l, U \in \mathcal{P}_{\leq\omega}(\mathcal{U})) \quad \stackrel{\text{def}}{=} \quad \left\{\beta'(l,u) \;\middle|\; u \in U\right\}. \tag{6}$$

Theorem 4.1. *$\beta : \iota E \to \iota E_{db}$ described above is an isomorphism.*

Proof. It suffices to exhibit the following isomorphisms and observe that β is obtained by applying them repeatedly left-to-right. We only do the first part:

$$\begin{aligned} [\mathbb{L}](X\times Y) &\cong [\mathbb{L}]X \times [\mathbb{L}]Y & [\mathbb{L}](X+Y) &\cong [\mathbb{L}]X + [\mathbb{L}]Y \\ [\mathbb{L}]\mathbb{N} &\cong \mathbb{N} & [\mathbb{L}]\mathbb{B} &\cong \mathbb{B} \\ [\mathbb{L}]\mathcal{P}_{\leq\omega}(X) &\cong \mathcal{P}_{\leq\omega}([\mathbb{L}]X) & [\mathbb{L}]\mathcal{P}_{<\omega}(X) &\cong \mathcal{P}_{<\omega}([\mathbb{L}]X) \\ [\mathbb{L}](X\text{-list}) &\cong ([\mathbb{L}]X)\text{-list} & [\mathbb{L}]\mathbb{A} &\cong \mathbb{N} + \mathbb{A} \\ && [\mathbb{L}][\mathbb{A}]X &\cong [\mathbb{L}]X \end{aligned} \tag{7}$$

So β distributes the abstraction $[\mathbb{L}]$- down until it reaches $\mathbb{A}$, at which point we obtain $\mathbb{N} + \mathbb{A}$.

We consider a selection of cases:

(1) *The case* $X \times Y$.

$$\begin{aligned} \hat{u} : [\mathbb{L}](X\times Y) &\longmapsto \text{И} l.\langle [l]\pi_1(\hat{u}@l), [l]\pi_2(\hat{u}@l)\rangle \\ \langle \hat{u}, \hat{u}'\rangle : ([\mathbb{L}]X)\times([\mathbb{L}]Y) &\longmapsto \text{И} l.[l]\langle \hat{u}@l, \hat{u}'@l\rangle \end{aligned}$$

The functions could also be written informally as $[l]\langle u,u'\rangle \mapsto \langle [l]u, [l']u\rangle$ and $\langle [l]u, [l]u'\rangle \mapsto [l]\langle u,u'\rangle$. This makes it intuitively obvious that they are well-defined and mutually inverse.

However, proving this now would be hard; best to wait for later when in FMG sets we will have powerful results about И (Corollary 6.7 and Theorem 6.3). For full proofs, see Theorem 6.10.

(2) *The case* $\mathbb{N}$. The functions are given by $\lambda\hat{n}.\text{И} l.(\hat{n}@l)$ and $\lambda n.\text{И} l.[l]n$. See Lemma 3.1.

(Note that $\hat{n} = \left\{\langle l,n\rangle \;\middle|\; l\in\mathbb{L}\right\}$ for some n, so $\hat{n}@l = \hat{n}@l'$ for any l and l' and $\text{И} l.(\hat{n}@l)$ is well-defined.)

(3) The case of $\mathbb{B}$ is similar; indeed, the result holds for any set all of whose elements have the trivial permutation action.

(4) *The case* $\mathcal{P}_{\leq\omega}(X)$.

$$\hat{U} : [\mathbb{L}]\mathcal{P}_{\leq\omega}(X) \longmapsto \text{И}l.\left\{[l]u \;\middle|\; u \in \hat{U}@l\right\}$$
$$U : \mathcal{P}_{\leq\omega}([\mathbb{L}]X)) \longmapsto \text{И}l.[l]\left\{\hat{u}@l \;\middle|\; \hat{u} \in U\right\}$$

It is worth discussing this in a little detail.

Suppose we have $\hat{U} : [\mathbb{L}]\mathcal{P}_{\leq\omega}(X)$. Choose $l\#\hat{U}$; we know $\hat{U}@l$ is well-defined and is a countable set of elements of $\mathcal{U}$. We also know that $l\#[l]u$ for any $u \in \hat{U}@l$ and so that $l\#\left\{[l]u \;\middle|\; u \in \hat{U}@l\right\}$. So the map above is well-defined (it does not depend on *which* $l\#\hat{U}$ we choose).

Now suppose we have $U : \mathcal{P}_{\leq\omega}([\mathbb{L}]X)$. This is a countable collection of elements so the support of U is the union of the support of its elements; choose some l such that $l\#\hat{u}$ for all $\hat{u} \in U$; in particular $\hat{u}@l$ is well-defined for all $\hat{u} \in U$. We know that $l\#[l]\left\{\hat{u}@l \;\middle|\; \hat{u} \in U\right\}$, so the map above is well-defined (it does not depend on which l we choose).

(5) *The case* $\mathbb{A}$.

$$\hat{a} : [\mathbb{L}]\mathbb{A} \longmapsto \text{И}l.\text{if } a = l_i \text{ then } i \text{ else } a$$
$$i : \mathbb{N} \longmapsto \text{И}l.[l]l_i$$
$$a : \mathbb{A} \longmapsto \text{И}l.[l]a.$$

So $[l]a$ corresponds to i if $a = l_i$ for some i and to a otherwise.

(6) *The case* $[\mathbb{A}]X$. The functions are given by

$$\hat{\hat{x}} : [\mathbb{L}][\mathbb{A}]X \longmapsto \text{И}l.\text{И}a.[a :: l](\hat{\hat{x}}@l@a) : [\mathbb{L}]X$$
$$\hat{x} : [\mathbb{L}]X \longmapsto \text{И}l.[tl(l)][hd(l)](\hat{x}@l) : [\mathbb{L}][\mathbb{A}]X.$$

This function is clearly a de Bruijn shift; the bound atom in $[\mathbb{L}][\mathbb{A}]X$ is pushed onto the head of the list of bound atoms, which is shifted up. Take $X = \mathbb{A}$. Then for example $[l][n]l_2$ maps to $[n :: l]l_2 = [l]l_3$.

□

We notice that ιE_{db} is *not quite* a de Bruijn datatype because it can have both dangling unbound atoms and unbound indices. Consider ιL_{db} as constructed above; then $\lambda 7$ and λa are both terms. Generally we allow only one or the other. A little extra notation solves the problem:

Given a class $V \subseteq \mathcal{U}$ write

$$\Gamma V \stackrel{\text{def}}{=} \left\{u \in V \;\middle|\; \mathsf{S}u = \emptyset\right\}.$$

Then we can prove by induction that $\Gamma\iota L_{db} = \iota(\mathbb{N} + X \times X + X)$. We have recovered a bona-fide de Bruijn datatype. $\Gamma[\mathbb{L}]\iota E$ consists of elements of ιE with free atoms bound in order. β restricts to an isomorphism between $\Gamma[\mathbb{L}]\iota E$ and $\Gamma\iota(E_{[\mathbb{A}\mapsto\mathbb{N},[\mathbb{A}]X\mapsto X]})$, using an informal notation.

Infinitary syntax: There are various directions in which the signature E (and its interpretation in $\mathcal{U}$) may be made infinitary. We can extend signatures with E-streams, which in $\mathcal{U}$ are interpreted by **countably infinite streams** (also called lazy lists) and which give rise to sets of abstract syntax with infinitely broad branching. We can add just $\mathbb{A}$-streams, which permits syntax to mention infinite lists of atoms. We can add $[\mathbb{L}]E$, which permits syntax to contain bindings by infinite lists of distinct atoms.

Our results carry through to these extensions. Furthermore in the rest of this paper we extend our theory powerfully to sets not necessarily of abstract syntax trees, and which in full FMG may be much larger and more complex than anything contained in $\mathcal{U}$.

5. α-EQUIVALENCE AS A MATHEMATICAL NOTION

We have related different kinds of inductive datatype for syntax, using datatypes of abstract syntax constructed in $\mathcal{U}$ by means of proofs based on induction on the signatures defining the datatypes. This made things nice and concrete, but it ties us to a concrete notion of signature and to working only on sets which happen to be abstract syntax trees.

We can do better: we can jettison both the signatures and their sets of abstract syntax trees, and define useful functions on general elements $u \in \mathcal{U}$ with no assumption of inductive structure. We call this type of reasoning 'syntax-free'.

5.1. **Equivariance.** Recall that $\mathcal{U}$ was defined with an inherently 'syntax-free' notion of 'free names of u' given by $\mathsf{S}u$. When $\mathsf{S}u = \emptyset$, u 'mentions no names'.

Because this is an important property we give it a fancy name: say u is **equivariant** when $\mathsf{S}u = \emptyset$.

We introduced a notion of 'equivariance' in §3, for classes (not elements) of $\mathcal{U}$. The two notions coincide in a suitable sense:

Lemma 5.1. (1) *u is equivariant precisely when $\forall \pi.\ \pi \cdot u = u$.*
(2) *u is equivariant precisely when $\forall y, \pi.\ y \in u \Rightarrow \pi \cdot y \in u$.*

Proof. The second part follows from the first, for suppose $\pi \cdot u = u$ always and $y \in u$. Recall that $\pi \cdot u = \{\pi \cdot y \mid y \in u\}$. Then obviously $\pi \cdot y \in u$. Suppose conversely $y \in u$ implies $\pi \cdot y \in u$. Then $\pi \cdot u \subseteq u$. But note that $P_{\mathbb{A}}$ is a group, so that $y = \pi^{-1} \cdot (\pi \cdot y) \in u$ and $u \subseteq \pi \cdot u$.

Now we prove the first part. Suppose u is equivariant. Then $\pi \# u$ always, and by the fourth part of Lemma 2.3, $\pi \cdot u = u$. Now suppose $\pi \cdot u = u$ always. Then by the second part of Lemma 2.3, $\pi \cdot \mathsf{S}u = \mathsf{S}u$ always, and the only way this can happen is if $\mathsf{S}u = \emptyset$. □

Call a function f **equivariant** when for all π and x, $f\pi \cdot x$ is defined and

$$\pi \cdot fx = f\pi \cdot x. \tag{8}$$

Lemma 5.2. *Suppose $f \in \mathcal{U}$ is the* graph *of a function. Then $\mathsf{S}f = \emptyset$ if and only if $\pi \cdot fx = f\pi \cdot x$ always — that is, the two notions of equivariance just described, agree where they overlap.*

Proof. Suppose $f = \{\langle x, fx\rangle \mid x \in dom(f)\}$. Then

$$\pi \cdot f = \{\langle \pi \cdot x, \pi \cdot fx\rangle \mid x \in dom(f)\}.$$

Suppose $\pi \cdot f = f$. Then we read $f(\pi \cdot x) = \pi \cdot fx$ directly off the equation above. Conversely suppose $f\pi \cdot x$ is always defined and $\pi \cdot fx = f(\pi \cdot x)$. Then $\langle \pi \cdot x, \pi \cdot fx\rangle \in f$. Thus f is defined at $\pi \cdot x$, and $f(\pi \cdot x) = \pi \cdot fx$ as required. □

Lemma 5.3. *If f is injective (and thus bijective between $imgf$ and $domf$) then f is equivariant if and only if f^{-1} is equivariant.*

Proof. Choose some $y = fx$. Then $\pi \cdot y = f\pi \cdot x$ and $(f^{-1}\pi \cdot y) = \pi \cdot x = \pi \cdot f^{-1}y$. □

How does the support of fx relate to the support of x?

Lemma 5.4. (1) *If f is equivariant then $\mathsf{S}fx \subseteq \mathsf{S}x$ and $\mathsf{Fix}(\mathsf{S}x) \subseteq \mathsf{Fix}(\mathsf{S}fx)$ always.*
(2) *If f is equivariant* and injective *then $\mathsf{S}fx = \mathsf{S}x$.*
(3) *If $f(x, y)$ is equivariant and injective, then $\mathsf{S}f(x, y) = \mathsf{S}x \cup \mathsf{S}y$.*

Proof. (1) Suppose f is equivariant, so $\pi \cdot fx = f(\pi \cdot x)$ always. Using Lemma 2.3 it is also the case that $\pi \cdot \mathsf{S}fx = \mathsf{S}f(\pi \cdot x)$. Now if $\pi \# \mathsf{S}x$ (that is, if $\pi \in \mathsf{Fix}(\mathsf{S}x)$) then $\pi \cdot x = x$ so $\pi \cdot \mathsf{S}fx = \mathsf{S}fx$. Since this is valid for *any* $\pi \# \mathsf{S}x$, it must be that $\mathsf{S}fx \subseteq \mathsf{S}x$.

(2) Direct from the first part of this lemma, and by the previous result.

(3) Suppose $\pi \cdot f(x,y) = f(x,y)$. By the various properties we assumed, this happens precisely when $\pi \cdot x = x$ and $\pi \cdot y = y$. The result follows.

□

It is generally useful to assume a function is equivariant: if not we parameterise until we obtain one that is.

A fundamental theorem of model theory is that if we systematically permute the underlying sets of a model, we obtain a new model in a natural way. Thus we know from very general principles (Truss 1994) that all functions, if sufficiently parameterised, must be equivariant.

5.2. **A word on support.** The reader familiar with FM theory will no doubt recall that support has another characterisation in terms of freshness and $\mathsf{И}$. This is indeed the case here:

Lemma 5.5. (1) *$a \# x$ if and only if $\mathsf{И}b.(b\ a) \cdot x = x$ ('a is not in x precisely when if we rename it to be a fresh b, nothing happens').*

(2) *$p \# x$ if and only if $\mathsf{И}q.(q\ p) \cdot x = x$ ('p is not in x precisely when if we rename it to be a fresh q, nothing happens'). .*

(3) *As a corollary of part 1, x is equivariant precisely when $\forall a.\ a \# x$.*

(4) *As a corollary of part 2, x is equivariant precisely when $\forall p.\ p \# x$.*

Proof. (1) Suppose $\mathsf{И}b.(b\ a) \cdot x = x$. By part 2 of Lemma 2.3, also $\mathsf{И}b.(b\ a) \cdot \mathsf{S}x = \mathsf{S}x$. Since $\mathsf{S}x$ is countable and all but countably many b satisfy $b \notin \mathsf{S}x$, it follows by elementary properties of sets and part 1 of Lemma 2.2 that $a \notin \mathsf{S}x$.

Conversely suppose $a \notin \mathsf{S}x$. Then for each $b \notin \mathsf{S}x$ it is the case that $(b\ a) \cdot x = x$ by part 4 of Lemma 2.3. The result follows.

(2) Suppose $\mathsf{И}q.(q\ p) \cdot x = x$. That means that there is some countable set S such that for all $q \# S$, $(q\ p) \cdot x = x$. By part 2 of Lemma 2.6 we know $\mathsf{S}q = \{q_1, q_2, \ldots\}$, and by part 2 of Lemma 2.2 $\mathsf{S}S = S$. Since all sets concerned are countable and $\mathbb{A}$ is uncountable, there is one (actually uncountably many) $q \# S, p, x$ such that $(q\ p) \cdot x = x$. By part 2 of Lemma 2.3 also $(q\ p) \cdot \mathsf{S}x = \mathsf{S}x$ and it follows by part 1 of Lemma 2.2 that $p \# x$.

Conversely suppose $p \# \mathsf{S}x$. By part 3 of Lemma 2.6, $\{p_1, p_2, \ldots\} \cap \mathsf{S}x = \emptyset$ and we can use that part again along with part 4 of Lemma 2.3 to verify that for any other $q \# \mathsf{S}x$ we have $(q\ p) \cdot x = x$. The result follows.

(3,4) Left-to-right is immediate since it follows that $\mathsf{S}x = \emptyset$. Right-to-left follows since the intersection of any set with the empty set, is empty.

□

Strictly as an aside for the reader who is not surprised by part 3 of the result above, because they have seen it in FM, we now show that $\mathcal{U}$ is in fact rather special and that in full FMG we cannot take this result for granted.

Define an elementwise permutation action on *uncountable* sets $C \subseteq \mathcal{U}$ as $\pi \cdot C = \{\pi \cdot u \mid u \in C\}$ and let $a \# C$ be defined as above by $\mathsf{И}b.(b\ a) \cdot C = C$ (as is the case in FM). Then we semi-formally (because we have not defined FMG) state:

Lemma 5.6. *Part 3 of Lemma 5.5 fails in full FMG; there exist C and π such that $\forall a.\ a \# C$ is* true, *but $\pi \cdot C = C$ is* false.

(Of course we have not yet defined FMG set theory, but being a set theory of significant expressive power, it includes uncountable sets and in particular the ones we construct in the proof below.)

Proof. Choose any $p \in \mathbb{L}$ and define

$$C \stackrel{\text{def}}{=} \{\pi \cdot p \mid \mathsf{S}\pi \text{ is finite}\}.$$

(Recall that $\mathsf{S}\pi$ being finite is equivalent to $\pi(a) = a$ holding for all but finitely many atoms. The canonical example of such a π is $(a\ b)$, and the canonical example of a π which is *not* so is $(q\ p)$ for some $q\#p$.)

We can easily verify that $a\#C$ for any a, but also for any fresh $q\#p$ it is the case that $(q\ p) \cdot C \neq C$. □

In fact, on of our reasons for considering $\mathcal{U}$ is that we can build and investigate $\mathbb{L}$ and $\mathsf{И}$ in it, and *still* have part 1 of Lemma 5.5. We return to this issue later in §6.2 (*fuzzy support* in FMG sets); a suitable version of part 4 of Lemma 5.5 remains valid, see Lemma 6.11.

First, we return to $\mathcal{U}$ because even without fuzzy support, it is by no means without novelty…

5.3. **Abstractive functions or, syntax-free "quotient by α-equivalence".** Call an equivariant function f **purely abstractive** when

$$\forall x, x' \in dom(f).\ fx = fx' \Rightarrow \exists \pi \in \mathsf{Fix}(\mathsf{S}fx).\ x' = \pi \cdot x.$$

Write $\mathbf{Id} : \mathcal{U} \to \mathcal{U}$ for the **identity function** mapping u to itself. Later it will be useful to write $\mathbf{Id}_X$ for $\mathbf{Id}$ restricted to being a partial function on some $X \subseteq \mathcal{U}$.

Lemma 5.7. (1) ***Id*** *is abstractive.*
(2) *The* ***abstraction function*** $\lambda a, x.[a]x : \mathbb{A} \times \mathcal{U} \to \mathcal{U}$ *is purely abstractive.*
(3) *The unique function* ! *from* $\mathbb{A}$ *to the unit set* $1 = \{*\}$ *is purely abstractive.*
(4) *The tail-of and head-of functions* $tl : \mathbb{L} \to \mathbb{L}$ *and* $hd : \mathbb{L} \to \mathbb{A}$ *are purely abstractive.*
(5) *The 'remove atom' function* $\lambda U, a.U \setminus \{a\}$ *defined for* $U \in \mathcal{P}_{\leq\omega}(\mathbb{A})$*, is purely abstractive.*
(6) *The tail-of and head-of functions* tl : $\mathbb{A}$*-stream* $\to$ $\mathbb{A}$*-stream and* hd : $\mathbb{A}$*-stream* $\to \mathbb{A}$ *are* not *purely abstractive.*

Proof. (1) If $\mathbf{Id}x = \mathbf{Id}y$ then $x = y$ and we take π equal to $\mathbf{Id}$ (the identity *permutation*; we shall always make it clear which $\mathbf{Id}$ we mean).
(2) Suppose $[a]x = [a']x'$. Choose some fresh $c\#a, x, a', x'$. Then $[a]x@c = (c\ a) \cdot x = (c\ a') \cdot x' = [a']x'$ by Lemma 2.8. It follows using Lemma 2.3 that $(a, x) = (a\ c) \circ (c\ a') \cdot (a', x')$, and we observe by calculations that $(a\ c) \circ (c\ a') \in \mathsf{FixS}([a]x)$.
(3) Suppose $!a = !b$. Well, that always happens, and $\mathsf{S}* = \emptyset$ so it suffices to observe that $(b\ a) \cdot a = b$.
(4) Suppose $tl(p) = tl(q)$. By definitions we know that $hd(p)\#tl(p)$ and $hd(q)\#tl(q)$, so $(hd(p)\ hd(q)) \in \mathsf{S}tl(p)$. Now observe that $(hd(p)\ hd(q)) \cdot p = q$.
Suppose $hd(p) = hd(q)$. Choose some $r\#p, q$. Then observe that $(r\ tl(p)) \cdot p = (r\ tl(q)) \cdot q$. It follows by calculations that $p = (r\ tl(p)) \circ (r\ tl(q)) \cdot q$, and $(r\ tl(p)) \circ (r\ tl(q)) \# hd(p), hd(q)$.
(5) Suppose $U \setminus \{a\} = U' \setminus \{a'\}$. Choose some fresh $c\#a, U, a', U'$ and observe that $(c\ a) \cdot (U \setminus \{a\}) = (c\ a') \cdot (U' \setminus \{a'\})$. The result follows similarly to the case for abstractions.

(6) It suffices to provide a counterexample. Consider $s = [a, a, a, a, a, \ldots]$ and $s' = [b, a, a, a, a, \ldots]$. Then $tl(s) = tl(s')$ but there is no π such that $\pi \cdot s' = s$. Similarly consider s and $s'' = [a, b, b, b, b, \ldots]$. Then $hd(s) = hd(s'')$ but there is no π such that $\pi \cdot s = s''$.

□

We shall assume basic properties of permutations, freshness, and support (most notably Lemma 2.3) **without comment henceforth**.

α from (5) is not purely abstractive in general. For the example of L and L_{nc}, observe that $\alpha(\lambda a.\lambda b.b) = \alpha(\lambda a.\lambda a.a)$ but no π exists to make them equal.

This example motivates the following definition:

Call f **Barendregt abstractive** when for every y in $img(f)$ there is a unique $\mathsf{Fix}(\mathsf{S}y)$-orbit in $f^{-1}y$ whose support has maximal cardinality. Call an element of this orbit a **Barendregt representative**.

Lemma 5.8. (1) $\mathbb{A} \times \mathbb{A} \to 1$ *is Barendregt abstractive with a Barendregt representative* $\langle a, b\rangle$ *for (any)* $a \neq b$.
(2) $\mathbb{A} + \mathbb{A} \to 1$ *is not Barendregt abstractive because* $\boldsymbol{Inl}(a)$ *and* $\boldsymbol{Inr}(a)$ *both map to* $*$ *and are not related by a permutation* π.
(3) α *from §4 is Barendregt abstractive. A Barendregt representative of* $\lambda[a]\lambda[a]a$ *is* $\lambda a.\lambda b.b$.

Proof. By simple calculations. A λ-term satisfying the Barendregt variable convention (Barendregt 1984) (that all bound variable names be distinct from each other and from the free variable names) is precisely what we here call a Barendregt representative. □

It may be useful to recall that $\lambda[a]\lambda[b]b = \lambda[a']\lambda[b']b' = \lambda[a]\lambda[a]a \in \iota L$. However $\lambda a.\lambda b.b \neq \lambda a'.\lambda b'.b' \neq \lambda a.\lambda a.a \in \iota L_{nc}$ — all these expressions map to the same element under α.

Given classes $X, X', Y, Y' \subseteq \mathcal{U}$ and $f : X \to X'$ and $g : Y \to Y'$, write $f \times g : X \times Y \to X' \times Y'$ and $f + g : X + Y \to X' + Y'$ for the obvious functions on products and sums. For convenience, write $\mathbf{Inl}(X) \subseteq X + Y$ for the left-hand component of the disjoint sum, and similarly for $\mathbf{Inr}(Y)$.

Theorem 5.9. *Continue the notation just established. Then:*
(1) *If* f *and* g *are purely/Barendregt abstractive then* $f + g$ *mapping* $\boldsymbol{Inl}(x)$ *to* $\boldsymbol{Inl}(fx)$ *and* $\boldsymbol{Inr}(y)$ *to* $\boldsymbol{Inr}(gy)$ *is purely/Barendregt abstractive.*
(2) *If* f *and* g *are purely/Barendregt abstractive then* $f \times g$ *mapping* $\langle x, y\rangle \in dom(f) \times dom(g)$ *to* $\langle fx, gy\rangle$ *is Barendregt abstractive.*
(3) *The map* $\lambda x, y.[x]y$ *is Barendregt abstractive.*

As a corollary, the usual α*-equivalence quotient on inductive syntax is Barendregt abstractive.*

Proof. (1) The Barendregt representative of $fx \in \mathbf{Inl}(X')$ is x, similarly for fy.
(2) We consider just the case of products. Given $z = \langle z_x, z_y\rangle \in X' \times Y'$ let x and y be Barendregt representatives. Let π_x be a permutation in $\mathsf{FixS}z_x$ mapping any difference between $\mathsf{S}x$ and $\mathsf{S}z_x$ to some fresh set of atoms, and similarly let $\pi_y \in \mathsf{FixS}z_y$ map $\mathsf{S}y \setminus \mathsf{S}z_y$ to some *other* fresh set of atoms. The Barendregt representative of z is $\langle \pi_x \cdot x, \pi_y \cdot y\rangle$.

If we accept that the usual α-equivalence quotient on inductive syntax is the identity on most components, and the abstraction function $\lambda a, x.[a]x$ on $\mathbb{A} \times X$ where appropriate, then the result follows by induction on the datatype. □

Corollary 5.10. *Second projection on a pair whose first element is in $\mathbb{A}$, write this $\pi_2 = \lambda a, x.x$, is Barendregt abstractive. Similarly for second projection on a pair whose first element is in $\mathbb{L}$.*

Proof. Combining the previous theorem with the observations about ! which precede it. The Barendregt representatives are tuples $\langle a, z\rangle$ such that $a\#z$. □

(Recall that $a\#z$ means $a \notin \mathsf{S}z$.)

There is a simple connection between Barendregt abstractive and purely abstractive functions:

Lemma 5.11. *A Barendregt abstractive map gives rise to a purely abstractive map on the subset of its domain consisting of elements with support of maximal cardinality.*

Proof. By unpacking definitions. □

So results of purely abstractive functions can usually be extended to ones of Barendregt abstractive functions, modulo the 'junk' of non-Barendregt representatives. Purely abstractive functions are a little easier to work with, so we may prefer to consider them, but this is just for convenience unless stated otherwise.

Something quite similar to all this is discussed (in the framework of ZF sets) in well-known work by McKinna and Pollack (McKinna & Pollack 1999). The apparatus of abstractive functions can be viewed as an abstract account of a phenomenon of which an instance is treated in (McKinna & Pollack 1999) as a concrete methodology.

We have defined a new abstract class of functions and shown that a concrete class of functions are a subset of them. Barendregt abstraction corresponds to 'quotient by α-equivalence'.

For this to be most useful, we need a notion of 'pick a representative'.

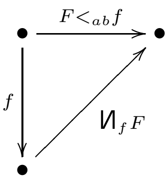
5.4. **Generalised И and Barendregt representatives.** Suppose f is a purely abstractive equivariant function and suppose F is another equivariant function with $dom(F) = dom(f)$. Write

$$F <_{ab} f \quad \text{when} \quad \text{for all } x \in dom(f),\ \mathsf{S}Fx \subseteq \mathsf{S}fx.$$

We argued in the previous subsection that f is a general α-equivalence quotient map. We shall now show that $F <_{ab} f$ is the condition 'F respects (f)-α-equivalence'.

Theorem 5.12. *For fixed f there is a one-to-one correspondence between $F <_{ab} f$ and equivariant h such that $dom(h) = img(f)$ and $img(h) = img(F)$. Write $\text{И}_f F$ for the unique h corresponding to F:*

(9)



(If f is Barendregt abstractive, the result still holds, modulo the 'junk' of non-Barendregt representatives in $dom(f)$.)

Proof. Suppose we have F as in the statement of the theorem. We now define $\text{И}_f F$: choose any $y \in img(f)$ and pick a representative of y, that is, some $x \in dom(f)$ such that $fx = y$. Let $\text{И}_f F(y) \stackrel{\text{def}}{=} Fx$.

We must show this is well-defined. So suppose x' is any other element such that $fx' = y$. By the purely abstractive property there exists $\pi \in \mathsf{FixS}y$ such that $x' = \pi \cdot x$. By equivariance of F we know $Fx' = F\pi \cdot x = \pi \cdot Fx$. Also by $F <_{ab} f$ we know $\mathsf{S}Fx \subseteq \mathsf{S}fx = \mathsf{S}y$ and so $\mathsf{FixS}y \subseteq \mathsf{FixS}Fx$. So $\pi \in \mathsf{FixS}Fx$ so by Lemma 5.4 we know $Fx' = \pi \cdot Fx = Fx$ as required.

The reverse map is given by composition: given h such that $dom(h) = img(f)$ the corresponding F is $h \circ f$.

It remains to show that these maps are inverse, that is

$$(10) \qquad (\text{И}_f F) \circ f = F \quad \text{and} \quad \text{И}_f(h \circ f) = h.$$

(1) For any x, by definition $(\text{И}_f F)(fx) = F(x')$ for some x' such that $fx' = (fx)$. Since f is assumed purely abstractive, there is some $\pi \# fx$ such that $x' = \pi \cdot x$. But since $F <_{ab} f$ also $\pi \# Fx$. So $Fx' = \pi \cdot Fx = Fx$.
(2) For any y, by definition $\text{И}_f(h \circ f)(y) = (h \circ f)(x)$ for some x such that $fx = y$. So this is just hy.

□

If $img(f)$ is a two-element set we can interpret it as truth values. We may use this result for predicates without comment.

For example:

- There is a 1-1 correspondence between maps f out of $\mathbb{A} \times X$ such that $a \# f(a, x)$ always, and maps out of $[\mathbb{A}]X$. This is a known principle of FM techniques (Gabbay & Pitts 2001, Lemma 6.3).
- If $F = \lambda a, z.a \# z$ (restricted to those a and z such that $a \# z$) and $f = \pi_2 = \lambda a, z.z$ then h is the (always true) function $\lambda z.\text{И}a.a \# z$. This is the known axiom (Fresh) (Gabbay & Pitts 2001, page 8).
- There is a 1-1 correspondence between maps f out of $\mathbb{L} \times X$ such that $p \# f(p, x)$, and maps out of $[\mathbb{L}]X$.
- ... and so on.

The construction above is useful because it systematises the treatment of choosing representatives and calculating on them, also for sets which may be much more complicated than products and projections.

We made a simplifying assumption that f, F, and h be all equivariant. This becomes false the moment they are parameterised over z', which may contain atoms. Those parameters can simply be incorprated into the arguments, or (if the reader likes) they can verify that the proofs above do generalise to this more general situation.

5.5. **Freshness.** We observed in the last subsection that:

(1) $! : \mathbb{A} \to 1$ is purely abstractive and
(2) $\pi_2 : \mathbb{A} \times Z \to Z$ is Barendregt abstractive.

The corresponding generalised И quantifier $\text{И}_! F$ and $\text{И}_{\pi_2} F$ are the well-known FM И-quantifier, choosing a fresh atom (for a parameter $z \in Z$).

In the rest of this section we concentrate on $! : X \to 1$. In fact $\pi_2 : X \times Z \to Z$ is more useful in practice because there the atom is chosen fresh *for a parameter*; the arguments below will (obviously) work for π_2 as well, just with a small overhead of writing 'Z' everywhere which we prefer to avoid.

For what sets of X does $\text{И}x \in X.\phi(x)$ have proper meaning?

Theorem 5.13. *$! : X \to 1$ is purely abstractive if and only if for all $x, x' \in X$ there exists π such that $x' = \pi \cdot x$.*

Proof. Suppose ! is purely abstractive. $\mathsf{S}* \in 1 = \emptyset$ and $\mathsf{Fix}\emptyset = P_{\mathbb{A}}$. Also $!(x) = !(x')$ always. Unpacking definitions, for all x and x' there is some π such that $\pi \cdot x = x'$.

Conversely suppose for all x and x' that there is some π such that $\pi \cdot x = x'$. Clearly this validates the condition for ! to be purely abstractive. □

Thus ! above is purely abstractive when X consists of a single orbit under $P_{\mathbb{A}}$.

We shall write $\mathsf{И}x \in X.F(x,z)$ for $\mathsf{И}_{\pi_2}F$, and we say that X has a $\mathsf{И}$-quantifier. In fact it suffices for $!\colon X \to 1$ to be Barendregt abstractive (e.g. $X = \mathbb{A}^\omega$), but then we can restrict attention to the elements of maximal support in X (e.g. $\mathbb{L} \subseteq X$, see below) from which generalised $\mathsf{И}$ chooses its representative. On this subset ! is purely abstractive, thus the situation of the theorem above is indeed canonical and useful.

We now show that $\mathbb{L}$ satisfies this condition. Recall that $p \in \mathbb{L}$ is a countably infinite stream of distinct atoms.

Lemma 5.14. *For all $l, l' \in \mathbb{L}$ there is a π such that $l' = \pi \cdot l$.*

Proof. π is given by $(l\ r) \circ (r\ l')$ for any $r \# l, l'$. □

(Recall $r \# l, l'$ means $r_i \neq l_j$ and $r_i \neq l'_{j'}$ for *all* i, j, j'. Recall from Lemma 2.7 cannot just use $(l'\ l)$.)

$\mathbb{A}$ and $\mathbb{L}$ are minimal and maximal sets for which a $\mathsf{И}$-quantifier exists, in the following sense:

Lemma 5.15. *If X is equivariant and such that for $x, x' \in X$ there always exists some π with $\pi \cdot x = x'$, then X is a quotient of $\mathbb{L}$ under some subgroup of $P_{\mathbb{A}}$.*

Proof. Pick any $x \in X$. This has countable support, so put the atoms in that support in some order and, if the set is finite, pad it with countably many distinct fresh atoms. Write this stream $p \in \mathbb{L}$ and declare $\phi(\pi \cdot p) = \pi \cdot x$. It is not hard to verify that provided X is equivariant then ϕ is well-defined and surjective, and that its kernel is a subgroup of $P_{\mathbb{A}}$. □

(Recall we stated we would only be concerned with equivariant classes, in §3.)

We used this technology to study the π-calculus (Gabbay 2003b) and models of process calculi with dynamic allocation (Gabbay 2003a). FreshML is based on a universe with finite support, so $\mathbb{L}$ does not appear; in a lazy programming version based on a language such as Haskell, $\mathbb{L}$ might be useful for foundations.

A benefit of these FM(G) techniques is that they not only account for binding in the syntax, but also in the *semantics*, which may very well include functions and consider infinite behaviour with infinite allocation of names and thus binding by $\mathbb{L}$ or variants of it.[4]

5.6. **Scope extrusion for $\mathsf{И}$.** Generalised $\mathsf{И}$ exhibits **generalised scope-extrusion**. This is the characteristic property of most notions of freshness, that if we choose a fresh name in some context, we can also choose 'fresher' for a wider context.

Theorem 5.16. *Suppose f, F, and G are equivariant. Suppose f is abstractive, that $domF = domf = X$, that $imgf = X'$, that $imgF = Y$, that $domG = Z \times Y$, and that $imgG = U$.*

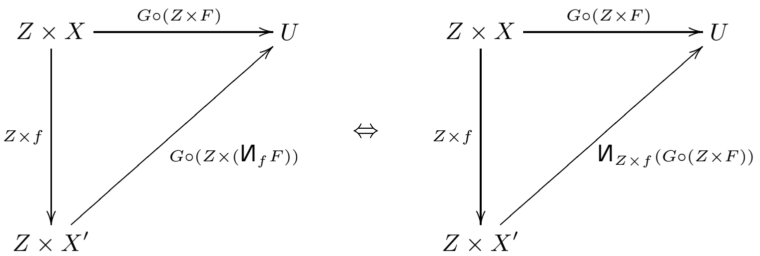
Suppose $F <_{ab} f$. Then $\lambda z, x.G(z, Fx) <_{ab} \boldsymbol{Id}_Z \times f$, $\mathsf{И}_f F$ and $\mathsf{И}_f(\lambda z, x.G(z, Fx))$ are both well-defined, and

$$G(z, \mathsf{И}_f Fx) = \mathsf{И}_{\boldsymbol{Id}_Z \times f} G(z, Fx). \tag{11}$$

[4]A variant of this not using $\mathbb{L}$ was developed and applied by Shinwell and Pitts to give semantics to FreshML, see (Shinwell & Pitts 2005).

In pictures, the left-hand diagram commutes if and only if the right-hand diagram commutes:

(12)



Here the Z annotating the arrows actually are the *identity* $\mathbf{Id}_Z$. We use this notation without comment henceforth. $\circ$, as before, denotes function composition.

Proof. The proof is just by unpacking definitions. Since these have accumulated we give the process in a little detail.

The conditions on domains and images just ensure that all functions are well-defined.

$\lambda z, x.G(z, Fx) <_{ab} \mathbf{Id}_Z \times f$ when for all z and x, $\mathsf{S}G(z, Fx) \subseteq \mathsf{S}\langle z, fx\rangle$. By Lemma 5.4 we know $\mathsf{S}\langle z, fx\rangle = \mathsf{S}z \cup \mathsf{S}fx$. Since G is equivariant we know by Lemma 5.4 that $\mathsf{S}G(z, Fx) \subseteq \mathsf{S}z \cup \mathsf{S}Fx$, and we are done. Therefore the functions in question are well-defined.

Now $G(z, \text{И}_f Fy)$ is the value $G(z, Fx)$ at any x such that $fx = y$ and x has maximal support. $\text{И}_f \lambda z, x.G(z, Fx)(z, y)$ is the value of $G(z, Fx')$ at any x' such that $fx' = y$ and x' has maximal support *and* such that $\mathsf{S}x' \cap \mathsf{S}z$ is minimal. In particular $Fx = Fx'$ and we are done. □

When we work with representatives we do not worry about generating names fresh for *all possible* contexts: this is impossible since we do not know a priori what atoms those contexts may contain.

So the above tells us what we always knew; that it does not matter because we can always extrude the scope of our choice of fresh name ad hoc. It is not unusual to see results of the form "$(\nu[a]P) \mid Z$ and $\nu[a](P \mid Z)$ are equivalent up to bisimilarity" (this example is of course taken from the π-calculus and we have written Z instead of Q to connect the notation to the theorem above). Usually such results are proved on a case-by-case equivalence-by-equivalence basis, perhaps one day they might be stated as 'general nonsense' consequences of theorems like the ones above.

The following lemma is useful in Theorem 6.10, and we also used it 'secretly' when we informally wrote some isomorphisms in Theorem 4.1.

Lemma 5.17. *Suppose F, and G are equivariant. Suppose that $domF = \mathbb{L} \times X$, $imgF = Y$, $domG = \mathbb{L} \times Y$, and $imgG = U$. Suppose further that $F, G <_{ab} \pi_2$ (that is if $l\#x$ and $l\#y$ then $l\#F(l, x)$ and $l\#G(l, y)$).*

Then $\lambda l, x.G(l, F(l, x)) <_{ab} \pi_2$ and $\text{И}l.G(l, \text{И}l.F(l, x)) = \text{И}l.G(l, F(l, x))$ (and this is all well-defined).

In pictures, the left-hand diagram commutes if and only if the right-hand diagram commutes:

$$(13)\qquad \begin{array}{ccc} \mathbb{L}\times\mathbb{L}\times X & \xrightarrow{G\circ(\mathbb{L}\times F)} & U \\ {\scriptstyle \pi_2\circ\pi_2}\big\downarrow & \nearrow_{\text{И}_{\pi_2} G\circ(\mathbb{L}\times \text{И}_{\pi_2} F)} & \\ X & & \end{array} \quad\Leftrightarrow\quad \begin{array}{ccc} \mathbb{L}\times X & \xrightarrow{\lambda l,x.G(l,F(l,x))} & U \\ {\scriptstyle \pi_2}\big\downarrow & \nearrow_{\text{И}_{\pi_2} \lambda l,x.G(l,F(l,x))} & \\ X & & \end{array}$$

Proof. By the properties of И we know it does not matter which Barendregt representatives we use. So suppose we used the representative (l, x) to calculate $F(l, x) = y$ so that $l\#x$. Then $l\#y$ and we can use (l, y) to calculate $G(l, y)$. The result follows. □

6. The underlying theory of sets

6.1. **Axioms. FMG sets** is a set theory similar to the original FM sets (Gabbay 2000), but considerably more general. So far we have worked in $\mathcal{U}$, a concrete model which is an initial fragment of a standard cumulative hierarchy model of FMG. We now define the theory.

The logic of FMG sets is First-Order Logic (Barwise 1977, van Dalen 2002) with a constant symbol $\mathbb{A}$ the **set of atoms** and a binary predicate $\in$ **set inclusion**.

The axioms of FMG sets are given in Figure 1.

We write $\mathcal{P}(x)$ for the powerset of x and $wo(x)$ for the predicate "there exists a set which is a well-ordering of the elements of x".

(A well-ordering is a well-founded trichotomous relation. A relation $R \subseteq x \times x$ is trichotomous when for all $z, z' \in x$, precisely one of zRz', $z = z'$, or $z'Rz$ holds. A relation is well-founded when there exist no infinite strictly descending chains. The canonical example of a well-ordering is the numerical 'less than or equal' relation on natural numbers $0, 1, \ldots$. This ceases to be well-founded if we add negative numbers, and is not trichotomous if we move to the complex integers $a + ib$, ordered by the size of their modulous.)

Where does FMG stand with respect to FM sets?

FM sets has a notion of a 'small set of atoms'. The И quantifier means precisely 'for all but a small set of atoms, . . . '.

Originally (Gabbay 2000, Gabbay & Pitts 1999) 'small' was equal to 'finite'. We also took $\mathbb{A}$ to be countably infinite, so the $\text{И}a.\phi(a)$ means 'for countably many a, $\phi(a)$'. It was folklore that we could just have well have taken $\mathbb{A}$ to be uncountable; the important point was what we might call the 'size of small'.

$\mathcal{U}$ took 'small' to mean 'countably infinite'. Of course we also insist $\mathbb{A}$ be uncountable, so that И is not degenerate.

FMG set theory generalises this further to:

'small' means 'internally well-orderable'.

This is very nice because (rather unexpectedly, perhaps) it makes no commitment to actual size; FMG sets is consistent with any of the following axioms: "Small is finite", "small is countable", and "small is ω^ω" (here ω is the first uncountable ordinal).

If we add the first as an axiom in FMG sets we recover FM sets. The second possibility gives a set theory suitable for containing $\mathcal{U}$ in its standard cumulative hierarchy model. Further possibilities give theories with increasingly large small sets of atoms. Or, we can add no extra axiom and consider FMG with no further assumptions.

(Sets)	$\forall x, y.\ x \in y \Rightarrow y \notin \mathbb{A}$
(Extensionality)	$\forall x, y \notin \mathbb{A}.\ (\forall z.\ z \in x \Rightarrow z \in y) \Rightarrow x = y$
(Collection)	$\forall x.\ \exists y \notin \mathbb{A}.\ \forall z.\ z \in y \Rightarrow (z \in x \wedge \phi) \quad (y \text{ not free in } \phi)$
($\in$-Induction)	$(\forall x.\ (\forall y \in x.\ [y/x]\phi) \Rightarrow \phi) \Rightarrow \forall x.\ \phi$
(Replacement)	$\forall x.\ \exists z.\ \forall y.\ y \in z \Rightarrow \exists x'.\ (x' \in x \wedge y = F(x'))$
(Pairset)	$\forall x, y.\ \exists z.\ x \in z \wedge y \in z$
(Union)	$\forall x.\ \exists y.\ \forall z.\ z \in y \Rightarrow (\exists w \in x.\ z \in w)$
(Powerset)	$\forall x.\ \exists y.\ \forall z.\ z \in y \Rightarrow \forall w \in z.\ w \in x$
(Infinity)	$\exists x.\ \exists y.\ y \in x \wedge \forall y \in x.\ \exists w \in x.\ y \in w$
(AtmLrg)	$\neg wo(\mathbb{A})$
(Fresh)	$\forall x.\ \exists S \in \mathcal{P}(\mathbb{A}).\ wo(S) \wedge S \text{ supports } x$

F any function-class, ϕ any predicate defined in the logic of FMG. $wo(x)$ is a predicate expressing that there exists a set which is a well-ordering of x.

FIGURE 1. Axioms of FMG

As we mentioned in the Introduction, 'small equals well-orderable' seems the right choice, because it gives enough isomorphisms to rename atoms:

Lemma 6.1. *For any two well-orderable sets α and β, one is in bijection with a subset of the other.*

Proof. See (Johnstone 1987, Lemma 6.3). ☐

We obtain permutations from these injections and we can use them to rename atoms to be fresh. This turns out in practice to be *the* property we need a small set to satisfy: that the set can always be renamed to be fresh (and *the* property of a large set is that this is not the case).

Write **ZF** for Zermelo-Fraenkel set theory (a classic mathematical foundation (Bell & Machover 1977, Johnstone 1987)).

Theorem 6.2. *FMG set theory is consistent relative to ZF. In addition, for any ordinal α FMG set theory with an additional axiom 'small is α' is consistent relative to ZF.*

Proof. The first part is a corollary of the second, so we concentrate on the second part.

FMG is just a theory in first-order logic (whose axioms are presented for the delectation of future generations in Figure 1). To prove consistency of *any* theory in first-order logic it suffices to exhibit a model. Because of fundamental limits to logic (Gödel 1931, Proposition VI) a model, i.e. consistency, cannot be exhibited in any absolute sense, but only relative to assuming the consistency, i.e. the existence of a model, of another theory.

So suppose we have a model of Zermelo-Fraenkel set theory. We construct a cumulative hierarchy model of FMG set theory (with 'small = countable'; the generalisation to any α is very easy) as a subclass of the elements of that model. This suffices to prove relative consistency. We only sketch the construction, since this is an entirely standard method found often in the literature (Brunner 1996, Kunen 1980).

Choose any countable ZF set for $\mathbb{A}$. Define a hierarchy of increasing subclasses by:

(1) $V_0 = \mathbb{A}$.
(2) $V_{\alpha+1}$ is the set of subsets of V_α with small support, union with V_α.

A set x has small support when there exists a countable set of atoms S such that if $\pi \in \mathsf{Fix}S$ then $\pi \cdot x = x$.

Then the model of FMG is $V = \bigcup_i V_i$, the union of the V_i. Set-membership in the model is interpreted by set-membership in V.

The proof that this *is* a model of the axioms of FMG sets given in Figure 1 is long but routine. We only sketch it here:

(1) (Sets) is valid by construction; everything in V is an atom or a set of (sets of (sets of . . .)) sets of atoms.
(2) (Extensionality) is inherited from ZF.
(3) (Collection) is proved by induction on the size of ϕ, a formula in the logic of FMG (the same logic in which the axioms are expressed).
(4) ($\in$-Induction) is by the inductive nature of the construction.
(5) (Replacement) is inherited from ZF.
(6) (Pairset) and (Union) likewise.
(7) (Powerset) is inherited from ZF, bearing in mind that we only take the sets with countable support.
(8) (Infinity) is inherited from ZF.
(9) (AtmLrg) is by the fact that any set encoding a well-ordering of $\mathbb{A}$ is invariant *only* under the identity permutation on $\mathbb{A}$.
(10) (Fresh) by the construction of the cumulative hierarchy, which only takes sets with small support.

□

In the presence of an axiom that 'small' is (at least) 'countable' we can interpret $\mathcal{U}$ in this cumulative hierarchy and abstractions, as promised, are implementable as the collection defined in §2.1, which is now also a set.

The permutation action $\pi \cdot x$ is defined by ϵ-induction starting at atoms and pointwise on sets, thus:

$$\pi \cdot a = \pi(a) \qquad \pi \cdot x = \{\pi \cdot x' \mid x' \in x\}$$

The following result corresponds in FMG to the same result in FM (see for example (Gabbay & Pitts 2001, Lemma 4.7) and (Pitts 2003, Proposition 2)):

Theorem 6.3 (Equivariance)**.** (1) *If $\phi(x_1, \dots, x_n)$ is a predicate in FMG then*

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

is always provable.

(2) *If $f(x_1, \dots, x_n)$ is a function-class specified in FMG then*

$$\pi \cdot f(x_1, \dots, x_n) = f(\pi \cdot x_1, \dots, \pi \cdot x_n).$$

Proof. (1) By induction on the language of FMG (just like the similar result for FM (Gabbay 2000, Gabbay & Pitts 2001)). The important base cases are $\pi \cdot \mathbb{A} = \mathbb{A}$ and $x \in y$ if and only if $\pi \cdot x \in \pi \cdot y$ by the pointwise definition of permutation.

(2) A function-class is merely a predicate specifying its graph. We use the first part to deduce

$$\phi(x_1, \dots, x_n, z) \Leftrightarrow \phi(\pi \cdot x_1, \dots, \pi \cdot x_n, \pi \cdot z)$$

and the result follows immediately.

□

In words:

> Any predicate specified in (the logic of) FMG sets is equivariant over its arguments, and so is any function-class.

FMG set theory is a set theory so models of FMG sets are closed under powersets, not just countable powersets as $\mathcal{U}$ was (*nota bene:* FMG powersets do not take all the subsets of the ambient foundation, only those with small support). FMG sets displays a richness of structure far exceeding $\mathcal{U}$.

6.2. **Lifting constructions in $\mathcal{U}$ to the set theory.** $\mathbb{A}$ is a set in FMG, identified by the predicate - $\in \mathbb{A}$. So are the natural numbers, pairs, cartesian products, disjoint sums, lists, streams, and their constructors such as $\langle x, y\rangle$, $\mathbf{Inl}(x)$, $\mathbf{Inr}(x)$, and so on; the constructions are just as in §2.3.

In particular we recall that $\langle x, y\rangle$ can be implemented in sets as $\{\{x\}, \{x, y\}\}$.

$\mathbb{L}$ and $\mathbb{A}$ are now sets. Function-sets are built as their graphs as is standard. $P_{\mathbb{A}}$ is identified by:

$$P_{\mathbb{A}} \stackrel{\text{def}}{=} \left\{ f : \mathbb{A} \to \mathbb{A} \;\middle|\; f \text{ bijective} \right\}.$$

Lemma 6.4. (1) $\pi' \cdot \pi = \pi^{\pi'}$.
(2) *Any permutation $\pi \in P_{\mathbb{A}}$ is the identity off some well-orderable set of atoms.*

In the first part, $\pi' \cdot \pi$ denotes π' (the permutation) acting on π (the function-set). $\pi^{\pi'}$ denotes the function-set expressing the graph of the permutation $\pi' \circ \pi \circ {\pi'}^{-1}$.

Proof. (1) We observe that π' acts on the graph of π by acting on the atoms within it; it is a fact of group theory that conjugation does the same.
(2) If S supports π then S contains every atom such that $\pi(a) \neq a$ (since otherwise we could permute $\pi(a)$ to something else). Since π is a set, it has well-orderable support. The result follows.

□

If $S \subseteq \mathbb{A}$ write

$$\mathsf{Fix}(S) \quad \text{for} \quad \{\pi \in P_{\mathbb{A}} \mid \forall a \in S.\ \pi(a) = a\}.$$

We say

$$S \textbf{ supports } x \quad \text{when} \quad \forall \pi \in P_{\mathbb{A}}.\ \pi \in \mathsf{Fix}(S) \Rightarrow \pi \cdot x = x.$$

Finally define

$$\mathsf{S}x \quad \text{by} \quad \bigcap \left\{ S \subseteq \mathbb{A} \;\middle|\; \forall \pi.\ \pi \in \mathsf{Fix}(S) \Rightarrow \pi \cdot x = x \right\},$$

(though the next result shows this is not as useful as we might imagine).

Lemma 6.5. (1) *If S and T support x and are well-orderable, then so is $S \cap T$.*
(2) $\mathsf{S}x$ *does* not *necessarily support x.*

Proof. (1) Suppose S and T support x and suppose π' fixes just $S \cap T$. Let $U = S \backslash T$. By (AtmLrg) $\mathbb{A}$ is not well-orderable and since U certainly is, we can choose some U' disjoint from S, T, U, and a supporting set for π'.

Since U and U' are well-orderable and the same size, there is an idempotent permutation $\pi = \pi^{-1}$ bijecting U with U' and fixing all other atoms (swap the ith elements of U and U', for arbitrary but fixed well-orderings). Now π fixes T so $\pi \cdot x = x$. Now π' fixes $S \cap T$ and also U' so $\pi \circ \pi' \circ \pi$ which is equal to ${\pi'}^{\pi}$ (since $\pi = \pi^{-1}$) fixes $\pi \cdot (S \cap T) \cup \pi \cdot U'$ which includes S. Therefore $\pi \circ \pi' \circ \pi \cdot x = x$. Applying π to both sides and simplifying we deduce that $\pi' \cdot x = x$ as required.
(2) It suffices to consider the example of Lemma 5.6.

□

So here, at least, FMG is not as tractable as FM (or $\mathcal{U}$)! The notion of *a supporting set* is still valid and useful, and we may still intersect supporting sets to obtain smaller supporting sets — but if we do this infinitely often we may 'miss an infinitesimal bit of support'.

When $\mathsf{S}x$ does not support x, say x **has fuzzy support**. When $\mathsf{S}x$ *does* support x, say it **has sharp support**.

$\mathcal{U}$ identifies a convenient subset of the FMG universe which exhibits inductive datatypes of abstract syntax with infinitary binding, but still has sharp support.

Lemma 6.6. (1) *Permutations have sharp support, and* $\mathsf{S}\pi = \{a \mid \pi(a) \neq a\}$.
(2) $a \in \mathbb{A}$ *and* $p \in \mathbb{L}$ *have sharp support, and* $\mathsf{S}a = \{a\}$ *and* $\mathsf{S}p = \{p_1, p_2, \dots\}$. *(As before)* $\pi \cdot p = [\pi \cdot p_1, \dots]$.
(3) *If* x *has sharp support, then so does* $\pi \cdot x$.
(4) *If* S *supports* x *then* $\pi \cdot S$ *supports* $\pi \cdot x$. *As a corollary,* $\mathsf{S}\pi \cdot x = \pi \cdot \mathsf{S}x$.

Proof. (1) A permutation is represented as a set by its graph; e.g. $(a\ b) = \{\langle a, b\rangle, \langle b, a\rangle, \langle c, c\rangle, \langle d, d\rangle, \dots\}$. It is not hard to verify the requisite properties, recalling that π acts pointwise on sets and $\langle x, y\rangle$ is just $\{\{x\}, \{x, y\}\}$.
(2) By concrete calculations.
(3) The property of being a supporting set can be expressed by a predicate ϕ in FMG. The result follows by part 1 of Theorem 6.3 (a proof by concrete calculations is also possible). The second part follows by calculations, or directly by part 2 of Theorem 6.3.

□

As a matter of notation write

$$x\#y \qquad \text{when} \qquad \exists S, T.\ S \text{ supports } x \ \wedge\ T \text{ supports } y \ \wedge\ S \cap T = \emptyset.$$

The reader may like to compare this with the corresponding definition from §2.1, which was $x\#y$ when $\mathsf{S}x \cap \mathsf{S}y = \emptyset$. The difference is of course due to the possibility of fuzzy support; if x and y have sharp support the two versions become equivalent.

The following result is a useful corollary of Equivariance (it also generalises Lemma 5.4, which there was proved for $\mathcal{U}$ by concrete calculation):

Corollary 6.7. (1) *If* S *supports* $x_1, \dots, x_n$ *and* f *is a function-class specified in the language of FMG with parameters amongst the* x_i*, then* S *supports* $f(x_1, \dots, x_n)$.
(2) *As a corollary, if* $f(x_1, \dots, x_n)$ *is a function-class in FMG with parameters* $x_1, \dots x_n$*, then if* $p\#x_1, \dots, x_n$ *then* $p\#f(x_1, \dots, x_n)$.

Proof. It suffices to apply the second part of Theorem 6.3. □

In words:

> Any function-class specified in (the logic of) FMG sets, does not create support.

We verify that a suitable version of Lemma 2.3 is still valid:

Lemma 6.8. (1) *If* S *supports* x *then* $\pi \cdot S$ *supports* $\pi \cdot x$. *Also,* $\pi \cdot (\mathsf{S}x) = \mathsf{S}(\pi \cdot x)$.
(2) $x\#[x]y$.
(3) *If* $\pi\#x$ *then* $\pi \cdot x = x$.
(4) *If* $\pi(a) = \pi'(a)$ *for all* $a \in S$ *for some* S *supporting* x*, then* $\pi \cdot x = \pi' \cdot x$.

Proof. (1) Direct from Theorem 6.3 for the predicate 'supports' and the function 'S'.
(2) From Lemma 6.9.
(3) $\pi\#x$ when π fixes a set supporting x. The result follows.

(4) By definition of supporting set.

□

Abstractions $[x]y$ are defined much as in §2.1, but we must assume x has sharp support, and protect against the possibility that y has fuzzy support by *not* using $\mathsf{S}y$:

$$[x]y \stackrel{\text{def}}{=} \{\langle \pi \cdot x, \pi \cdot y\rangle \mid \exists S.\ S \text{ supports } y \wedge \pi \# S \backslash \mathsf{S}x\}.$$

Lemma 6.9. (1) $\pi \cdot ([x]y) = [\pi \cdot x](\pi \cdot y)$ *holds.*

(2) *If x has sharp support then S supports y if and only if $S\backslash \mathsf{S}x$ supports $[x]y$.*

Proof. (1) By part 2 of Theorem 6.3.

(2) In essence identical to the proof of the case for abstractions of part 4 of Lemma 2.3, we sketch half of it in full:

We can prove by calculation that $\{\pi \mid \exists S.\ S \text{ supports } y \wedge \pi \# S\backslash \mathsf{S}x\}$ is a group.

Suppose x has sharp support and S supports y. Suppose $\pi \in \mathsf{Fix}(S\backslash \mathsf{S}x)$. Then

$$\begin{aligned}\pi \cdot [x]y &= \big\{\langle \pi' \cdot (\pi \cdot x), \pi' \cdot (\pi \cdot y)\rangle \ \big|\ \exists S', \pi'.\ S' \text{ supports } \pi \cdot y \wedge \pi' \in \mathsf{Fix}(S' \backslash \pi \cdot \mathsf{S}x)\big\} \\ &= \big\{\langle \pi'' \cdot x, \pi'' \cdot y\rangle \ \big|\ \exists S', \pi''.\ S' \text{ supports } y \wedge \pi'' \in \mathsf{Fix}(S' \setminus \mathsf{S}x)\big\} = [x]y.\end{aligned}$$

Conversely if we suppose S supports y then the result follows similarly.

□

(It does not seem fruitful to consider $[x]y$ when x does not have sharp support.)

The maps between ιE, ιE_{nc}, and ιE_{db}, can also be constructed as before.

It is not hard to verify that Lemma 2.10 transfers unchanged to FMG. The discussions of purely abstractive and Barendregt abstractive functions, И, and freshness, transfer almost word-for-word.

6.3. **Abstractions commute with pairsets, powersets, and more.** Theorem 4.1 generalises beautifully in FMG. We give just the case for cartesian product, and also full function spaces and powersets (not just countable ones, as we did in Theorem 4.1).

Theorem 6.10. (1) $[\mathbb{L}](X \times Y) \cong [\mathbb{L}]X \times [\mathbb{L}]Y$.

(2) $[\mathbb{L}](Y^X) \cong [\mathbb{L}]Y^{[\mathbb{L}]X}$.

(3) $[\mathbb{L}]\mathcal{P}(X) \cong \mathcal{P}([\mathbb{L}]X)$.

Similarly for $\mathbb{A}$, e.g. $[\mathbb{A}](Y^X) \cong [\mathbb{A}]Y^{[\mathbb{A}]X}$.

Proof. We consider just the cases for $\mathbb{L}$; the cases of $\mathbb{A}$ is simpler.

(1) We map

$$\begin{array}{lll} \hat{u} \in [\mathbb{L}](X \times Y) & \text{to} & \text{И}l.\langle [l]\pi_1(\hat{u}@l), [l]\pi_2(\hat{u}@l)\rangle \\ \langle \hat{u}_1, \hat{u}_2\rangle \in ([\mathbb{L}]X) \times ([\mathbb{L}]Y) & \text{to} & \text{И}l.[l]\langle \hat{u}_1@l, \hat{u}_2@l\rangle \end{array}$$

These maps are well-defined since it is easy to use Corollary 6.7 and part 2 of Lemma 6.8 to deduce $l\#\langle [l]\pi_1(\hat{u}@l), [l]\pi_2(\hat{u}@l)\rangle$ and $l\#[l]\langle \hat{u}_1@l, \hat{u}_2@l\rangle$.

We now verify that these maps are self-inverse. By Lemma 5.17 it suffices to check that (for one suitably fresh l)

$$\begin{aligned}\hat{u} &= [l]\langle ([l]\pi_1\hat{u}@l)@l, ([l]\pi_2\hat{u}@l)@l\rangle \\ \langle \hat{u_1}, \hat{u_2}\rangle &= \langle [l]\pi_1\big([l]\langle \hat{u}_1@l, \hat{u}_2@l\rangle\big)@l, [l]\pi_2\big([l]\langle \hat{u}_1@l, \hat{u}_2@l\rangle\big)@l\rangle.\end{aligned}$$

This follows easily, using Lemma 2.10.

(2) We map

$$\hat{f} \in [\mathbb{L}](Y^X) \qquad \text{to} \qquad \lambda\hat{x}.\mathsf{И}l.[l]((\hat{f}@l)(\hat{x}@l))$$
$$f \in ([\mathbb{L}]Y)^{[\mathbb{L}]X} \qquad \text{to} \qquad \mathsf{И}l.[l](\lambda x.f([l]x)@l)$$

We must check these maps are well-defined. Suppose $\hat{f}$ and $\hat{x}$ are given. Choose some $l\#\hat{f},\hat{x}$. Then $\hat{f}@l$ and $\hat{x}@l$ are well-defined and $l\#[l](\hat{f}@l)(\hat{x}@l)$. Conversely suppose f is given. Choose some $l\#f$. Then $l\#f([l]x)$ so $f([l]x)@l$ is well-defined and $l\#[l]\lambda x.f([l]x)@l$.

We now verify that these maps are self-inverse. By Lemma 5.17 it suffices to check that (for one suitably fresh l)

$$(\hat{f}@l)x = \big([l](\hat{f}@l)(([l]x)@l)\big)@l \qquad f\hat{x} = [l]([l](\lambda x.f([l]x)@l)@l)(\hat{x}@l)$$

This follows easily, making heavy use of Lemma 2.10.

(3) This is a corollary of the previous lemma, taking $Y = \mathbb{B}$ and observing that $[\mathbb{L}]\mathbb{B}$ is naturally isomorphic to $\mathbb{B}$, see Theorem 4.1.

In full, the maps are given by:

$$\hat{U} \in [\mathbb{L}]\mathcal{P}(X) \qquad \text{to} \qquad \hat{U}' \stackrel{\text{def}}{=} \mathsf{И}p.\left\{[p]u \;\middle|\; u \in \hat{U}@p\right\}$$
$$U \subseteq [\mathbb{L}]X \qquad \text{to} \qquad U' \stackrel{\text{def}}{=} \mathsf{И}p.[p]\left\{\hat{u}@p \;\middle|\; \hat{u} \in U \wedge p\#\hat{u}\right\}.$$

□

This result is important because it enables us to drop the condition implicit in the construction of $\mathcal{U}$ of considering just *countable* powersets. Since $\mathcal{P}(X)$ is strictly larger than X, and similarly Y^X is strictly larger than Y and X (provided Y and X have more than two elements), and they *not* inherit any inductive structure, this theorem is one way of making formal that our treatment of names and bindings in FMG is strictly more general than the standard model given by abstract syntax trees.

This extension to function-sets is also useful, e.g. for the existence of a monad in the semantics of FreshML used to prove the language correct, and for equalities used in understanding models of behaviour (Shinwell, et al. 2003, Gabbay 2003a).

6.4. **An algebraic version.** We give an alternative presentation of $\mathcal{U}$ and FMG in algebraic style. Fix a particular uncountable set $\mathbb{A}$. Let $P_{\mathbb{A}}$ be permutations of $\mathbb{A}$ with countable support. Then a **Nominal FMG set** (where small=countable) is:

$$\begin{gathered}\forall x.\ \mathbf{Id}\cdot x = x \quad \forall \pi,\pi',x.\ \pi\cdot\pi'\cdot x = \pi\circ\pi'\cdot x\\ \forall x.\ \mathsf{И}p.\mathsf{И}q.(p\ q)\cdot x = x \quad \forall x.\ \mathsf{И}a.\mathsf{И}b.(a\ b)\cdot x = x.\end{gathered} \tag{14}$$

Here $\mathsf{И}a.\phi$ means that ϕ holds of all but a countable set of atoms, and $\mathsf{И}p.\phi$ means that ϕ holds of countable lists of distinct atoms whose atoms may be all but a countable set of atoms.

It is easy to verify that an equivariant FMG set (for any given model of FMG sets), or an equivariant subclass of $\mathcal{U}$, is a Nominal FMG set. If we need more atoms we just take $\mathbb{A}$ to be larger and $\pi \in P_{\mathbb{A}}$ to have any support strictly smaller than the size of $\mathbb{A}$.

Lemma 6.11. (1) *$\mathbb{L}$ is a nominal FMG set with small=countable.*
(2) *If $x \in X$ is such that $l\#x$ for all $l \in \mathbb{L}$ then $\pi \cdot x = x$ always.*

Proof. (1) By part 2 of Lemma 6.6.

(2) By part 1 of Lemma 6.6 we know that π permutes only countably many atoms. Let S support x (we use it in a moment). We choose some l' suitably fresh for π, S and deduce that $\pi \cdot (l'\ l) \cdot x = \pi \cdot x$. By part 3 of Lemma 6.6 we conclude that $\pi \cdot (l'\ l) \cdot x = x$.

□

It is now easy to see that if we know that small=X where X is some cardinality, then similar results hold for streams of distinct atoms of length an ordinal with cardinality X. This is the correct generalisation of the FM principle that $\forall a.\ a\#x$ implies x is equivariant.

7. Future work

Infinite binding arises in behavioural models of reactions: for example, if $P \stackrel{\text{def}}{=} \nu[a]\overline{a}a$ then the π-calculus process $!P$ is behaviourally equivalent to the 'infinitary' process $P \mid P \mid \ldots$ (Milner, et al. 1992). Also, if $\omega_3 \stackrel{\text{def}}{=} \lambda x.xxx$ then the λ-calculus process $\omega_3\omega_3$ is $\alpha\beta$-equivalent to the 'infinitary' process $(((\ldots\omega_3)\omega_3)\omega_3)\omega_3$. This equivalence can be made formal by considering a behavioural notion of model, which can display infinite structure even though the trace of any particular behaviour may be finite. One such structure is considered by Montanari et al (Montanari & Pistore 1998) which we can view as a de Bruijn-based model of behaviour which may generate fresh names, whose main interest is in powerful optimisations which merge nodes identical except for permutations of names.

If name-binding is modelled by FM-style abstractions, there is a need for infinite FM-style abstraction in the model of behaviour.[5] We have proposed one method of obtaining this effect. We have examined the proof-principles which seem useful on these objects, and by progressive abstraction we have formalised and generalised them to obtain notions such as 'abstraction and concretion by infinitely many atoms', 'abstractive functions', 'Barendregt representatives', 'generalised И', and we have described some nice results describing their properties.

$\mathcal{U}$ has an interesting capture-avoiding substitution action given as follows:

$$a\{a\mapsto u\} = u \quad b\{a\mapsto u\} = b \quad V\{a\mapsto u\} = \big\{v\{a\mapsto u\}\ \big|\ v \in V\big\}$$
$$([v']v)\{a\mapsto u\} = [v'](v\{a\mapsto u\}) \quad \mathsf{S}v' \cap \mathsf{S}u = \emptyset \tag{15}$$

(Recall that an element of $\mathcal{U}$ is either an atom a, a countable set V, or an abstraction $[v']v$, and that for any abstraction there is a choice of v' and v forming that abstraction such that $\mathsf{S}v' \cap \mathsf{S}u = \emptyset$; thus this is a total function.)

Since β-reduction is given in the λ-calculus, this suggests that we might use the FM model of abstraction to obtain models of the λ-calculus that are not Scott domains, and with a set-theoretic flavour. The obstacle to this idea is that we do not know what application is. This is current work.

References

H. P. Barendregt (1984). *The Lambda Calculus: its Syntax and Semantics (revised ed.)*, vol. 103 of *Studies in Logic and the Foundations of Mathematics*. North-Holland.

J. Barwise (1977). 'An Introduction to First-Order Logic'. In J. Barwise (ed.), *Handbook of Mathematical Logic*, pp. 5–46. North Holland.

J. Bell & M. Machover (1977). *A course in mathematical logic*. North-Holland.

N. Brunner (1996). '75 Years of Independence Proofs by Fraenkel-Mostowski Permutation Models'.

[5]Since this paper was submitted for publication, a paper has been published describing a semantics for FreshML which approximates this in a category with a FM-style monad representing finitely but unboundedly many abstractions (Shinwell & Pitts 2005).

N. G. de Bruijn (1972). '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.

J. B. Fraleigh (1994). *A First Course in Abstract Algebra*. Addison-Wesley.

M. J. Gabbay (2000). *A Theory of Inductive Definitions with alpha-Equivalence*. Ph.D. thesis, Cambridge, UK.

M. J. Gabbay (2003a). 'Fresh Graphs'. Submitted.

M. J. Gabbay (2003b). 'The π-calculus in FM'. In F. Kamareddine (ed.), *Thirty-five years of Automath*. Kluwer.

M. J. Gabbay & A. M. Pitts (1999). 'A New Approach to Abstract Syntax Involving Binders'. In *14th Annual Symposium on Logic in Computer Science*, pp. 214–224. IEEE Computer Society Press, Washington.

M. J. Gabbay & A. M. Pitts (2001). 'A New Approach to Abstract Syntax with Variable Binding'. *Formal Aspects of Computing* **13**:341–363.

K. Gödel (1931). 'On Formally Undecidable Propositions in Principia Mathematica and Related Systems I' Available online in translation by Martin Hirzel.

P. T. Johnstone (1987). *Notes on logic and set theory*. Cambridge University Press.

K. Kunen (1980). *Set Theory: An Introduction to Independence Proofs (Studies in Logic and the Foundations of Mathematics)*. Elsevier.

J. McKinna & R. Pollack (1999). 'Some Lambda Calculus and Type Theory Formalized'. *Journal of Automated Reasoning* **23**(3-4):373–409.

R. Milner, et al. (1992). 'A Calculus of Mobile Processes, II'. *Information and Computation* **100**(1):41–77.

U. Montanari & M. Pistore (1998). 'History-Dependent Automata'. Tech. Rep. TR-98-11.

L. C. Paulson (1994). 'A Fixedpoint Approach to Implementing (Co)Inductive Definitions'. In *CADE-12: Proceedings of the 12th International Conference on Automated Deduction*, pp. 148–161, London, UK. Springer-Verlag.

A. M. Pitts (2003). 'Nominal Logic, A First Order Theory of Names and Binding'. *Information and Computation* **186**:165–193.

M. R. Shinwell & A. M. Pitts (2005). 'On a Monadic Semantics for Freshness'. *Theoretical Computer Science* **342**:28–55.

M. R. Shinwell, et al. (2003). 'FreshML: Programming with Binders Made Simple'. In *Eighth ACM SIGPLAN International Conference on Functional Programming (ICFP 2003), Uppsala, Sweden*, pp. 263–274. ACM Press.

J. Truss (1994). 'Permutations and the axiom of choice'. In H. R.Kaye (ed.), *Automorphisms of first order structures*, pp. 131–152. OUP.

D. van Dalen (2002). 'Intuitionistic Logic'. In D. Gabbay & F. Guenthner (eds.), *Handbook of Philosophical Logic, 2nd Edition, Volume 5*, pp. 1–114. Kluwer.