Permutation Models for Set Theory

May 3, 2013

Contents

1 Overview

1.1 The problem of naming

Naming entities is an aspect of abstraction so basic that it is easy to forget its existence entirely. Good communication between humans requires good names to be chosen, but good mathematics behaves the same regardless of which names are used. This irrelevance is a red herring: the requirement that choice of name be irrelevant to semantics is a restriction that makes the *problem of naming* a critically relevant and somewhat subtle one.

A key property of naming is *referential transparency*, that when a name is replaced by its referent or vice versa, meaning is preserved: this ensures that names can be introduced to or eliminated from a context without compromising expressivity. With this in mind, one at-first appealing approach is to semantically *identify* names with their referents, giving them no character or identity of their own. But when names are used for abstraction, they need not have an associated concrete referent, indeed they often will not do so: after all, the purpose of abstraction is to work with fewer details and hence broader applicability than any concrete situation. Hence, names *cannot* be fully transparent, and must have some internal structure. But any inspection of that structure or identity from within the abstraction violates the irrelevance property!

Maintaining this discipline becomes a tedious burden on the paper author, or worse still, their readers. Barendregt's iconic book on the lambda calculus [1, p. 26] made use of the convention that "terms that are α-congruent [equal up to renaming] are identified", and wherever there is imprecision, there are casualties of it:

> "We thank T. Thacher Robinson for showing us on August 19, 1962 by a counterexample the existence of an error in our handling of bound variables."

The above quotation is from an article written by Kleene [9, p. 16], so clearly this is not only an issue of education or experience.

1.2 Purpose of this essay

The purpose of this essay is to describe a set-theoretic technique to treat names in such a way that renaming is guaranteed to preserve meaning. The principal idea is to restrict the universe to sets that are *symmetric* in the sense that they are invariant with respect to certain automorphisms, which correspond to renaming variables: such universes are the titular *permutation models*. To do so enables "badly-behaved" (non-name-invariant) functions to be excluded by construction.

This approach to naming was pioneered by Gabbay and Pitts [7], and has been covered in detail by several authors. However, many of them focus on describing the technique and its applications, leaving the set-theoretic foundations to just one or two central references. The set theory behind permutation models goes back to Fraenkel and Mostowski in the 1930s, with some improvements made by Specker. In more modern works, Jech [8] develops the theory in a somewhat heavyweight way, leaving major components as an exercise to the reader, while Felgner [4] takes a simpler but more laborious approach.

For the development of permutation models, I will be using Felgner's method, since it seems to me more elementary than Jech's, but not Felgner's proofs; an idea due to Thomas Forster shows that syntactic properties of formulae can be identified that make the proofs much lighter work.

For the application to the problem of naming, I will be following Gabbay and Pitts' original paper [7], although my presentation of their ideas will be somewhat different. In particular, where they appeal to category-theoretic ideas of initial algebras and their construction, I attempt to explain in purely set-theoretic terms.

The original contribution of this essay, then, is to provide a *sound foundation* for the solution of the naming problem using permutation models that is at once *simple* while not unduly *laborious*, and accessible to readers who are more at home with sets and relations than objects and morphisms.

First, however, I present a brief example of the problem, and an inadequate solution.

1.3 Nominal structures: the hard way

The lambda calculus is perhaps the simplest system in which careful treatment of naming is a necessity. Originally developed as a primitive model of computation, λ-terms (over an infinite collection of variables $a, b, c, \ldots$ [1]) are defined recursively as follows:

- Any *variable* forms a λ-term on its own,
- The *application* of two λ-terms s and t, denoted st, is a λ-term, and
- The λ-*abstraction* $\lambda n.t$, where n is a variable and t is a λ-term, is a λ-term.

A λ-abstraction $\lambda n.t$ is said to *bind* n in t, so an occurrence of a variable is called *bound* if it is inside a λ binding it. Any other occurrence is *unbound*.

Notionally, a λ-abstraction represents a function, taking a named parameter and returning a result that depends on the parameter somehow. Function evaluation is captured in the notion of β-reduction: $(\lambda n.E)F$ is said to β-reduce to the expression $E[n \mapsto F]$, that is to say, E with all *unbound* occurrences of n replaced with F. The reason for the unboundedness condition is so that variables always refer to their innermost binding λ: this allows subexpressions of a larger expression to be understood individually.

Under this interpretation, it is clear that $\lambda x.x$ and $\lambda y.y$ are meant to represent the same function, so terms are called *α-equivalent* if they can be obtained from one another by renaming variables in an appropriate sense. More precisely, α-equivalence, sometimes written $=_\alpha$, is the smallest equivalence relation that identifies each $\lambda x.E$ with $\lambda y.E[x \mapsto y]$ for each pair of variables x, y,[2] and inductively identifies terms whose subterms are equivalent. If the function interpretation is to hold true, operations defined on Λ should respect this equivalence, and so correspond to a function on $\Lambda/{=_\alpha}$, the set of λ-terms up to α-equivalence, also written Λ_α.

[1] Twenty-six is infinite enough for the present purpose.

[2] This is still not entirely precise, since I haven't given a precise interpretation of the substitution operation yet.

In order to formalise these notions, the set of lambda terms may be presented as a recursive data type over the constructors

$$\begin{aligned} \mathrm{var} &: A \to \Lambda \\ \mathrm{app} &: \Lambda \times \Lambda \to \Lambda \\ \mathrm{lam} &: A \times \Lambda \to \Lambda \end{aligned}$$

Set-theoretically, the constructors are "free" functions, say the function that pairs a token identifying itself with a sequence of all its arguments, and a recursive data type is the smallest set closed under application of the constructors, or equivalently the $\subseteq$-least fixed point of the set operation

$$L(X) = \mathrm{var}``A \cup \mathrm{app}``(X \times X) \cup \mathrm{lam}``(A \times \Lambda).$$

It is easy to check that this operation preserves set inclusion, so by Knaster-Tarski,[3] its least fixed point is

$$\Lambda = \bigcap \{X : L(X) \subseteq X\}.$$

This presentation gives rise to an induction principle (that any property that holds for a term whenever it holds for all its parts in fact holds for all terms) and the following recursion principle:

Theorem 1.1. *Given functions $f_{\mathrm{var}} : A \to R$, $f_{\mathrm{app}} : R \times R \to R$, $f_{\mathrm{lam}} : A \times R \to R$, there is a unique function $f : \Lambda \to R$, satisfying:*

$$\begin{aligned} f(\mathrm{var}(a)) &= f_{\mathrm{var}}(a) \\ f(\mathrm{app}(s,t)) &= f_{\mathrm{app}}(f(s), f(t)) \\ f(\mathrm{lam}(a,t)) &= f_{\mathrm{lam}}(a, f(t)) \end{aligned}$$

Proof. Let X be the subset of Λ on which f is uniquely defined. By the recursion equations, $L(X) \subseteq X$, but Λ was minimal with respect to this property, so $\Lambda \subseteq X$, so $\Lambda = X$. □

[3]Pedantically, this is really by an argument analogous to the Knaster-Tarski theorem, rather than Knaster-Tarski itself. Models of ZF set theory are not complete lattices, e.g. because the empty set has no intersection, and in general unions and intersections need only exist of what the model believes to be sets, rather than all subsets of the model. However, if the candidate for the lfp given by K-T exists, then it is indeed a least fixed point.

To define $f(E) = E[x \mapsto F]$, the following seem like good candidates:

$$
\begin{aligned}
f_{\mathrm{var}}(a) &= \begin{cases} F & \text{if } a = x \\ a & \text{otherwise} \end{cases} \\
f_{\mathrm{app}}(s,t) &= \mathrm{app}(s,t) \\
f_{\mathrm{lam}}(a,t) &= \begin{cases} \mathrm{lam}(a,t) & \text{if } a \neq x \\ \dots \end{cases}
\end{aligned}
$$

In the last clause there is a problem. Firstly, it is incomplete: if $a = x$ then the recursion should stop, since further occurrences of x will not be unbound. This is simple enough to fix, although it requires some tedious plumbing, defining a function that returns the pair $\langle E, E[x \mapsto F]\rangle$ so that the unmodified first component can be used when no recursion is necessary. Secondly, and more critically, it is incorrect: variables in F may "become bound" by λ-abstractions that the substitution recurses through; this behaviour depends on unconnected terms having equal names, so violates α-equivalence, e.g. with $F = y$,

$$\lambda y.xy \mapsto \lambda y.yy,$$

but

$$\lambda z.xz \mapsto \lambda z.yz.$$

No big deal, you might think. The example I have is only problematic because it contains two unconnected uses of the variable y. If we only put up with the annoyance of doing some global renames to avoid any duplication at the beginning, no further problems will be encountered, right? Unfortunately, this simply isn't true. Consider the following example:

$$
\begin{aligned}
&(\lambda z.zz)(\lambda xy.xy) \\
\xrightarrow{\beta}&(\lambda xy.xy)(\lambda xy.xy) \\
\xrightarrow{\beta}&\lambda y.(\lambda xy.xy)y
\end{aligned}
$$

Despite starting with no free variables and every variable binding unique, the evaluation *introduces* non-uniqueness, so that the final stage has an application that cannot be safely reduced without renaming variables. It seems name resolution and evaluation simply cannot be fully teased apart.

And yet, all the failure modes are somehow fragile: there are only finitely many variables in a term, after all, and infinitely many unused ones, so the problem seems to be confined to an insignificant fragment of all λ-terms. Gabbay and Pitts [7] recognised this with their use of the $\reflectbox{N}$ quantifier, which captures simultaneously the idea that *almost any* and in fact *nearly all* names are suitable for avoiding the above problems. The permutation model and its automorphisms allow reasoning in both these existential and universal contexts at once, which allows reasoning to automatically respect α-equivalence while remaining remarkably close to the informal pen-and-paper proofs that work with concrete names and terms.

All of these concepts and methods will be explained, but a full explanation of what a permutation model is, and how they work, must come first. The next section of this essay is dedicated to that purpose, beginning with an overview of how and why the models were originally developed.

2 Permutation models

2.1 The origin of the permutation model

Permutation models were developed in the 1930s as an attempt to tackle the question of the independence of the Axiom of Choice and related statements from the axioms of Zermelo-Fraenkel set theory. They attempt to defeat choice by containing many elements that are indistinguishable from the perspective of the model – that the model cannot choose between. The inability to distinguish between these elements corresponds to automorphisms of the model, each witnessing that the language of the theory behaves identically on its orbits.

I will demonstrate that such models cannot be built for the theories traditionally underlying set theory, but that a modest modification makes the construction possible, and indeed the Axiom of Choice can then be defeated.

Though it can be shown [8, p. 95] that this result can be transplanted back into ZF, so that permutation models are indeed a legitimate method of proving the independence of AC, historically speaking Cohen's method of forcing proved the key result by other means entirely before this transplantation was realised, and so for some time the method of permutation models was regarded as merely a curiosity.

However, the problem of naming once again has use for the notion of indistinguishable elements. Here, the irrelevance property of the names is captured again by automorphisms of the universe, that prevent definable operations and suitably symmetric sets from misbehaving in the way that the λ-term substitution operation did.

In recognition of this importance, I present the following well-known result, which motivates departure from ZF in pursuit of automorphisms:

Theorem 2.1. *Any automorphism of a model of ZF is trivial.*

Proof. By $\in$-induction: let f be an automorphism of the universe, and let y be a set. Assume that $\forall x \in y\,[f(x) = x]$. Since f is an automorphism,

$$\forall x\,[x \in y \iff f(x) \in f(y)].$$

But, combining this with the inductive hypothesis,

$$x \in y \implies x \in f(y),$$

so $y \subseteq f(y)$. But f is an automorphism, so in particular it has an inverse, and by the above argument $y \subseteq f^{-1}(y)$, so $f(y) \subseteq y$. So $y = f(y)$, which completes the proof. □

2.2 Defining a theory of atoms

Theorem 2.1 establishes that ZF is incapable of supporting a permutation model in the sense of the previous section. To make space for nontrivial automorphisms, we start instead from a version of ZF enriched with additional elements which are not sets. These elements are called *atoms*, and can belong to other sets but cannot contain sets or each other, and the corresponding theory is called ZFA, to mean 'ZF with atoms'[4]. It includes a new unary predicate symbol in its syntax, with the interpretation *is-an-atom*; this gives rise to the class of atoms, denoted A, so that $x \in A$ means 'x is an atom', and $x \notin A$ means 'x is a set'.

With these new atoms introduced, it becomes necessary to specify how they interact with the relational symbol $\in$ of traditional set theory. First-order logic cannot *syntactically* prevent the membership of an atom being queried, so instead I will make a meaningless but convenient choice of behaviour for the membership relation, with the understanding that any theorems that depend essentially on such behaviour are to be interpreted with appropriate skepticism.

The most obvious such choice is "empty atoms": simply take as an axiom

$$\forall x \in A\,[\neg(y \in x)]. \tag{EA}$$

However, in conventional set theory there is an Axiom of Extensionality

$$\forall x, z\,[\forall y\,[y \in x \iff y \in z] \implies x = z] \tag{Ext}$$

which in combination with (EA) would render all atoms equal to the empty set. There are two ways to proceed:

[4]This is the terminology used by Jech [8] and others [7]; the initials 'ZFA' have also been used by other authors (e.g. [10, section 2.3]) to refer to non-wellfounded set theories, where the A corresponds to 'antifoundation'. Felgner [4] calls it ZFU instead (U for 'urelements', another name for atoms), but it doesn't seem to have caught on.

- Weaken (Ext) so that it only applies to sets. The conceptual justification for (Ext) is that the identity of a set is its membership, so given that atoms are not sets, it is not clear why extensionality should apply to them.

- Populate the atoms with elements in such a way that they can be distinguished: in particular, taking $x \in x$ for each atom x prevents their collapse under extensionality. The conceptual meaning of this approach is less clear, since atoms notionally cannot contain, so in particular cannot contain themselves; however, the negation of membership is no less nonsensical than membership itself, and we have already committed to taking formulae involving membership in atoms not too seriously in any case. The advantage of this approach is that (Ext) survives unmodified; however, the Axiom of Foundation, whose various equivalent statements include that there exist no infinite descending membership chains, is violated. A weakened Foundation, e.g. that all infinite descending membership chains meet the class of atoms, can be used as a replacement.

The choice between weakening Extensionality or Foundation seems inescapable, since Theorem 2.1 is precisely a consequence of those two axioms (indeed, in either case Theorem 2.1 weakens precisely to the statement that any automorphism that acts trivially on the atoms is the identity on the entire model).

Implementing the atoms as so-called Quine atoms (that is to say, satisfying $x = \{x\}$) allows a particularly slick proof of the relative consistency with ZF, so I will weaken Foundation. However, Extensionality is useless for proving equality of Quine atoms, since the proof obligation for declaring two atoms equal is that they have equal elements, which is no use when the sets themselves are those elements. In light of this, weakening Extensionality too seems no additional burden. Moreover, once I have my model with Quine atoms, it is at least plausible (though I have not checked the details, because I do not need the result) that modifying the membership relation to exclude all the $x \in x$ pairs would give a model with Foundation intact and Extensionality weakened. So it really is unimportant which choice is made here.

Nevertheless, *some* choice is necessary[5], so my axioms of ZFA are precisely those of ZF, but with a nontriviality axiom $\exists a\,[a \in A]$, and with Foundation replaced with

$$\forall x\,[x \cap A = 0 \implies (x \neq \emptyset \implies \exists y \in x\,[y \cap x = \emptyset])], \tag{WF}$$

that is to say, every set disjoint from the atoms satisfies the usual Foundation.[6]

[5]That's choice with a small c. AC is optional.

[6]Felgner [4, p. 52] weakens Foundation in a way that is superficially distinct but equivalent to this one: I choose this presentation so that the connection with ZF-foundation is more explicit.

Having constructed this theory, it is natural to ask how its consistency compares with that of ZF. On the one hand, given a model of ZFA, the collection of sets that ignore the atoms altogether – have no atoms anywhere in their transitive closure – clearly form a model of ZF. Such sets are called *pure sets*, and this model is called the *kernel* of a model of ZFA. So the consistency of ZFA implies the consistency of ZF.

Conversely, given a model for ZF, can models be constructed for ZFA? The answer is yes, but the construction is not obvious; the next section describes one method in detail.

2.3 Existence of models with atoms

Given a model $\mathcal{M} = \langle M, \in \rangle$ of ZF set theory, I define a relation on M such that M with the new relation is a model of ZFA with countably many atoms, and hence prove that the universe of sets and atoms, in which permutation models are constructed, is at least no less reliable than the familiar universe of sets.

The new membership relation is parametrised by a permutation[7] σ of M, and given by:

$$x \in_\sigma y \iff x \in \sigma(y).$$

For example, if σ is the permutation that swaps a set y and its singleton, then we have that $x \in_\sigma y \iff x \in \{y\} \iff x = y$, i.e. y becomes a Quine atom under the new relation.[8] Similarly, the permutation that swaps each nonzero[9] finite (von Neumann) ordinal with its singleton gives rise to a countably infinite collection of Quine atoms; still bolder choices can give rise to a proper class of atoms, though this essay will not require such extravagance. I will call the structure created in this manner $\mathcal{M}^\sigma = \langle M, \in_\sigma \rangle$.

On the face of it, permuting the sets in this seemingly arbitrary way might be expected to cause havoc with the set of sentences satisfied by the structure, but in fact the situation is not so dire, thanks to the following observations:

- Applying a permutation to all instances of a bound variable preserves truth, since as x ranges over all sets, so does $\sigma(x)$.

- If σ is a permutation, then $\sigma^{\rightarrow}(x) = \sigma``x = \{\sigma(y) : y \in x\}$ is a permutation (so the previous observation applies to it). This operation is functorial, in the sense that $(\sigma \circ \tau)^{\rightarrow} = \sigma^{\rightarrow} \circ \tau^{\rightarrow}$ and $\mathrm{id}^{\rightarrow} = \mathrm{id}$; indeed, functorial operations always map bijections to bijections.

[7] Despite the fact that this process constructs a model of set theory using a permutation, it is not a permutation model in the sense of the title of this essay!

[8] This is similar to how Felgner [4] produces Quine atoms, but he uses $x \in_F y \iff F(x) \in y$; the difference is largely cosmetic.

[9] Including zero is problematic because its singleton is also an ordinal.

- Converting a formula containing $\in_\sigma$ to one containing $\in$ involves introducing applications of σ to variables on the right of $\in$, and the following equivalences give ways of introducing applications elsewhere in a formula:

$$\begin{array}{l} x \in y \iff \sigma(x) \in \sigma^{\to}(y) \iff \sigma^{\to}(\sigma(x)) \in \sigma^{\to\to}(\sigma^{\to}(y)) \iff \dots \\ x = y \iff \sigma(x) = \sigma(y) \iff \sigma^{\to}(\sigma(x)) = \sigma^{\to}(\sigma(y)) \iff \dots \end{array}$$

  If this can be done in such a way that each variable gets the same permutation prefix on all of its occurrences, by the first observation they can all be removed, so the formula with $\in_\sigma$ is equivalent to the formula with $\in$.

This motivates the following definition:

Definition 2.2. A formula of the language of set theory is *stratified* if there exists an assignment of natural numbers to each variable[10] (a *stratification*) such that:

- whenever $x = y$ appears in the formula, x is assigned the same number as y, and
- whenever $x \in y$ appears in the formula, y is assigned a number one greater than the number assigned x.

A formula is *stratifiable* if it is equivalent to one which is stratified.

I will give numeric subscripts to variables to indicate stratifications or partial stratifications where illustrative. So, for example, $\exists x_1 [\forall y_0 [y_0 \notin x_1]]$ is stratified, but

$$\forall x [x \neq \emptyset \implies \exists y [y \in x \wedge y \cap x = \emptyset]]$$

is not, because $y \cap x = \emptyset$, shorthand for $\forall z_n [\neg(z_n \in y_{n+1} \wedge z_n \in x_{n+1})]$, requires y and x to be assigned the same number, while $y_m \in x_{m+1}$ requires them to be assigned different numbers.

The concept of stratification has a rich and interesting history, originating in an attempt to make type theory more palatable. Here, however, as Forster [5] realised, it is precisely the condition necessary to make use of the observations above:

Theorem 2.3. *If ϕ is a stratifiable sentence of the language of set theory, then $\mathcal{M} \models \phi \iff \mathcal{M}^\sigma \models \phi$ for any permutation σ.*

[10] Each variable, not occurrence: in $\forall x [x = y \vee x = z]$, both x must receive the same number, however in $\forall x [x = y] \vee \forall x [x = z]$, the x are allowed different numbers, because they are bound by different quantifiers.

Proof. Without loss of generality, ϕ is in fact stratified. It suffices to show that ϕ is equivalent to the formula ϕ_σ obtained by replacing each $\in$ in ϕ with $\in_\sigma$.

Define the permutations σ_n by $\sigma_0 = \mathrm{id}$, $\sigma_{n+1} = \sigma_n^{\rightarrow} \circ \sigma$, so that, for example, $\sigma_1 = \sigma$ and $\sigma_3 = \sigma^{\rightarrow\rightarrow} \circ \sigma^{\rightarrow} \circ \sigma$. Apply σ_n to each variable assigned n by the stratification; since the σ_n are permutations, this preserves truth. Now, by the definition of a stratification, each instance of equality in the formula looks like $\sigma_n(a) = \sigma_n(b)$, and each instance of membership looks like $\sigma_m(c) \in \sigma_{m+1}(d)$. But the former case is equivalent to $a = b$, since σ_n is a permutation, and the latter case is $\sigma_m(c) \in \{\sigma_m(z) : z \in \sigma(d)\}$, so is equivalent to $c \in \sigma(d)$, i.e. $c \in_\sigma d$. So the formula is equivalent to ϕ_σ. □

The ZF axioms of empty set, pair set, union, power set, infinity, and extensionality are all stratifiable, and hence all satisfied by $\mathcal{M}^\sigma$. The Axiom of Choice is stratifiable, and so holds in $\mathcal{M}^\sigma$ if it holds in $\mathcal{M}$. Foundation is not stratified, which is promising since it ought to fail, but Separation and Replacement, axiom schemas parametrised over formulae, will clearly also not be stratified if the formula parameter is not so. Fortunately, they still hold in ZFA; it is well-known that (e.g. using a partial identity function) Separation follows from Replacement, so it is sufficient to prove Replacement directly.

Theorem 2.4. $\mathcal{M}^\sigma \models$ Replacement

Proof. Let x be a set, and F be a function-class. We wish to find y so that

$$z \in_\sigma y \iff \exists w \in_\sigma x \, [z = F(w)],$$

or in other words

$$z \in \sigma(y) \iff \exists w \in \sigma(x) \, [z = F(w)].$$

Take $y = \sigma^{-1}(F``\sigma(x))$. □

To prove that $\mathcal{M}^\sigma$ is a model of ZFA with countably many atoms, it only remains to give an interpretation of the class of atoms A that is countably infinite, and with which $\mathcal{M}^\sigma$ satisfies the modified axiom of foundation (WF). At this point the choice of σ becomes relevant (since, after all, $\mathcal{M}^{\mathrm{id}}$ is not terribly interesting!). It suffices to take, as hinted above, the permutation swapping each nonzero finite ordinal with its singleton. Then let A be the set of nonzero finite ordinals of $\mathcal{M}$, each of which is a Quine atom in $\mathcal{M}^\sigma$, so A is countably infinite. It only remains to prove (WF):

Theorem 2.5. *With this choice of A and σ, $\mathcal{M}^\sigma \models WF$.*

The proof makes use of the following lemma:

Lemma 2.6. *If x has an $\in_\sigma$-element that is neither empty*[11] *nor an atom, then x is fixed by σ.*

Proof. If x is not fixed by σ, then x according to $\mathcal{M}$ is either a nonzero finite ordinal, or the singleton of one. If it is a nonzero finite ordinal, then $y \in_\sigma x \implies y \in \{x\} \implies y = x$, so the only $\in_\sigma$-element of x is x itself, which is an atom. Alternatively, $x = \{n\}$ for some finite nonzero ordinal n. Then $y \in_\sigma x \implies y \in n$, so y is a finite ordinal, so y is either empty or an atom. $\square$

Proof of Theorem 2.5. Let x be a set in $\mathcal{M}^\sigma$ disjoint from A and non-empty. It suffices to find an $\in_\sigma$-minimal element for x. Now, by the Lemma, either x $\in_\sigma$-contains the empty set, which suffices as an $\in_\sigma$-minimal element, or x is fixed by σ. Then, since $\mathcal{M} \models$ Foundation, it has an $\in$-minimal element, y, say. Either:

- $\sigma(y) = y$. Then both y and x have the same $\in$-elements as $\in_\sigma$-elements, so y is $\in_\sigma$-minimal for x.
- $\sigma(y) = \{y\}$. But this means y is an atom, which is impossible since y is an element of x, which is disjoint from A.
- $y = \{n\}$, $\sigma(y) = n$. Then the $\in_\sigma$-elements of y are atoms or empty, so y is $\in_\sigma$-disjoint from x. $\square$

Hence in fact all axioms of ZFA hold, and the consistency follows:

Theorem 2.7. *Let σ be the permutation swapping every nonzero finite ordinal with its singleton. If $\mathcal{M}$ is a model of ZF then $\mathcal{M}^\sigma$ is a model of ZFA with A countably infinite.* $\square$

Notice it is precisely the unstratified nature of Foundation that allows this construction to knock it out while leaving the other axioms undamaged: it is for this reason that I chose to weaken Foundation instead of Extensionality in defining ZFA.

By contrast, the proof of weak foundation leaves something to be desired in terms of clarity and elegance. Another choice of σ would lead to a different proof, so there is some potential for improvement here.

2.4 Automorphisms of models with atoms

As promised, Theorem 2.7 gives a model of a set theory with non-trivial automorphisms: in fact, the automorphisms of the model correspond exactly to permutations of the atoms. In order to prove this, it will be necessary to develop a principle

[11] Note that with this σ, the empty sets of $\mathcal{M}$ and $\mathcal{M}^\sigma$ are equal, so "z is empty" is unambiguous.

of $\in$-induction[12] from the weakened Foundation, and use it to make good on the earlier promise that Theorem 2.1 weakens to the condition that an automorphism that fixes the atoms fixes the model.

In fact, this development is somewhat backwards: I formulated the weakened Foundation precisely so that the induction principle I wanted would be true. It is as follows:

Theorem 2.8. *In a model of ZFA, if ϕ is a formula with a single free variable such that $\phi(a)$ holds whenever $a \in A$, and if $\forall y \in x\,[\phi(y)] \implies \phi(x)$, then $\phi(x)$ holds for all x.*

Proof. The proof is almost identical to that of $\in$-induction in ZF from the usual Axiom of Foundation.

Suppose ϕ is a property for which the antecedents hold. Suppose for a contradiction that $\exists x\,[\neg\phi(x)]$.

I will need a transitive set containing x. Let $TC(x)$ denote the $\in$-transitive closure of x, that is to say,

$$TC(x) = \bigcup\{x, \cup x, \cup\cup x, \dots\},$$

which exists by the axioms of Infinity, Replacement, and Union. $TC(\{x\})$ is the smallest transitive set containing x.

Consider $u = \{y \in TC(\{x\}) : \neg\phi(y)\}$. u is disjoint from A, since $\phi(a)$ holds for $a \in A$[13], and u is non-empty, since $x \in u$. Hence by (WF), u has an $\in$-minimal member z, say. But now, if $w \in z$ then $w \in TC(\{x\})$, and by minimality of z, $w \notin u$, so it must be the case that $\phi(w)$ holds. So $\forall w \in z\,[\phi(w)]$, but by assumption this means that $\phi(z)$, contradicting $z \in u$. ☐

An immediate application:

Theorem 2.9. *If f is an automorphism of a model of ZFA such that $\forall a \in A\,[f(a) = a]$, then $\forall x\,[f(x) = x]$. In particular, there is at most one automorphism for each permutation of the set of atoms.*

Proof. Let $\phi(x)$ be the property $f(x) = x$. ϕ holds for all the atoms and, by an argument identical to that of Theorem 2.1, holds for a set whenever it holds for its elements. Hence, by induction, ϕ holds for all sets.

[12] Hitherto $\in$ has meant ZF-membership and $\in_\sigma$ has meant ZFA-membership, but that distinction is no longer necessary: from now on, $\in$ means the membership relation of whichever model is under discussion.

[13] This is the sole difference from the proof of induction in ZF. In fact, arguably this proof is just the well-known fact that a relation that is well-founded relative to a class (say, the non-atoms) satisfies induction relative to that class. But expressing that is scarcely more concise than just presenting the proof.

In particular, suppose f and g are two automorphisms that permute the atoms in the same way. Then $f^{-1} \circ g$ is an automorphism that fixes the atoms, so it is the identity, so $f = g$. $\square$

In fact (unsurprisingly), "at most one" can be strengthened to "exactly one": any permutation π of the atoms can be extended to a permutation π^* of the model by setting $\pi^*|_A = \pi$ and, for each $x \notin A$, $\pi^*(x) = \{\pi^*(y) : y \in x\}$. The proof that this defines a function is by $\in$-induction; it is a homomorphism by definition, and bijective since its inverse is $(\pi^{-1})^*$.

The following facts are generally true of automorphisms, but benefit from being stated explicitly, so I can refer to them later:

Theorem 2.10. *If ϕ is a formula of the language of ZFA, and π an automorphism of a model $\mathcal{M}$, then*

$$\mathcal{M} \models \phi(x_1, \ldots, x_n) \iff \mathcal{M} \models \phi(\pi(x_1), \ldots, \pi(x_n))$$

Proof. This follows since it is true of all atomic formulae: $x = y \iff \pi(x) = \pi(y)$, $x \in y \iff \pi(x) \in \pi(y)$, and $x \in A \iff \pi(x) \in A$. $\square$

Theorem 2.11. *Automorphisms commute with definable operations. That is to say, if ϕ is a function-class in free variables $\vec{x} = x_1, \ldots, x_n$ and y, $f(\vec{x})$ denotes the unique y such that $\phi(\vec{x}, y)$, and π is an automorphism, then*

$$f(\pi(x_1), \ldots, \pi(x_n)) = \pi(f(\vec{x})).$$

Proof. This follows immediately from Theorem 2.10 applied to ϕ. $\square$

This complete characterisation of automorphisms of the universe will be essential in the following section, which (at last!) sets up the notion of a permutation model, by restricting attention to sets invariant under some collection of automorphisms: in particular, any function that "chooses between the atoms" cannot be invariant in this sense.

2.5 Construction and consistency of permutation models

As previously hinted, a permutation model is a submodel of a model of ZFA, taking only those sets invariant under certain permutations. The choice of which permutations is constrained by the requirement that the axioms of ZFA continue to be satisfied – to make sense of these constraints, some more definitions are needed. In what follows, $\mathcal{M}$ is a model of ZFA with a class of atoms A.

Definition 2.12. If G is a group of automorphisms of the model, then $\mathrm{stab}_G(x)$ is the stabiliser of x under the G-action of automorphism application, i.e. the subgroup $\{f \in G : f(x) = x\}$.[14]

Definition 2.13. A collection $\mathcal{F}$ of subgroups of a group G is a *normal filter* if:

$$H \in \mathcal{F}, H \leq K \leq G \implies K \in \mathcal{F}, \tag{$\uparrow$}$$
$$H, K \in \mathcal{F} \implies H \cap K \in \mathcal{F}, \tag{$\cap$}$$
$$H \in \mathcal{F}, g \in G \implies gHg^{-1} \in \mathcal{F}. \tag{N}$$

For example, the set of all subgroups containing a specified normal subgroup is a normal filter, and so is the set of all nontrivial subgroups of $\mathbb{Z}$ (the key behind this example being that the intersection of finitely many nontrivial subgroups of $\mathbb{Z}$ is nontrivial, even though the intersection of infinitely many distinct subgroups won't be).

Now, let G be a group of automorphisms of $\mathcal{M}$,[15] and $\mathcal{F}$ a non-empty normal filter of subgroups of G. Write $\mathrm{stab}(x)$ for $\mathrm{stab}_G(x)$.

Definition 2.14. A set is *symmetric* (with respect to G and $\mathcal{F}$) if $\mathrm{stab}(x) \in \mathcal{F}$. A set x is *hereditarily symmetric* if every member of $TC(\{x\})$ is symmetric (i.e. x is symmetric, and all its members are symmetric, and all their members, etc.).

Definition 2.15. A *permutation model* is the collection of all hereditarily symmetric sets (relative to G and $\mathcal{F}$) in a model of ZFA, together with the inherited membership relation.

The rest of this section is devoted to proving that a permutation model $\mathcal{N} = \langle \mathcal{M}, G, \mathcal{F} \rangle$ satisfies the axioms of ZFA.

Theorem 2.16. *Any pure set of $\mathcal{M}$ is a set of $\mathcal{N}$.*

Proof. Every automorphism fixes all pure sets. Since $\mathcal{F}$ is upward-closed and non-empty, it contains G, so $\mathrm{stab}(x) \in \mathcal{F}$ for any pure x. □

Corollary 2.17. $\mathcal{N} \models$ *Empty Set and Infinity.* □

Theorem 2.18. $\mathcal{N}$ *is transitive, that is to say, if $x \in y \in \mathcal{N}$ then $x \in \mathcal{N}$.*

Proof. If $x \in y$ then $TC(\{x\}) \subseteq TC(\{y\})$, so by the definition of hereditarily symmetric, $y \in \mathcal{N} \implies x \in \mathcal{N}$. □

[14] Felgner [4, p. 53] calls this $H[x]$; Jech [8, p. 46] calls it $\mathrm{sym}_{\mathcal{G}}(x)$.

[15] *A* group, not *the* group! G is permitted to be a proper subgroup of the full automorphism group.

Corollary 2.19. $\mathcal{N} \models$ *Extensionality and* (WF).

Proof. Extensionality and weak foundation are both *bounded* formulae, that is to say, all their internal quantifiers range only over elements of sets of the model, rather than over the whole model. It is well-known that such a formula is equivalent to the same formula relativised to a transitive submodel: e.g. in Extensionality, the antecedent is that the hereditarily symmetric members of x and y are the same, but since all members of hereditarily symmetric sets are hereditarily symmetric, this is equivalent to all the members of x and y being the same. $\square$

Lemma 2.20. *If $x \subseteq \mathcal{N}$ and x is symmetric, then $x \in \mathcal{N}$.*

Proof.
$$TC(\{x\}) = \{x\} \cup \bigcup_{y \in x} TC(\{y\}).$$
Hence, if all members of x are hereditarily symmetric and x is symmetric, x is hereditarily symmetric. $\square$

Theorem 2.21. *If f is a definable operation in the sense of Theorem 2.11, then if $\vec{x} = x_1, \ldots, x_n$ are symmetric, $f(\vec{x})$ is symmetric.*

Proof. Since the $\vec{x}$ are symmetric, $\mathrm{stab}(x_i) \in \mathcal{F}$ for each x_i. Then by ($\cap$), $\bigcap_{i=1}^n \mathrm{stab}(x_i) \in \mathcal{F}$. I claim that this intersection is contained in $\mathrm{stab}(f(\vec{x}))$, so by ($\uparrow$) the latter is in $\mathcal{F}$.

Proof of claim: let π be an automorphism that stabilises all the x_i. Then, by Theorem 2.11, $\pi(f(\vec{x})) = f(\pi(x_1), \ldots, \pi(x_n))$, but $\pi(x_i) = x_i$ for all i, so this is $f(\vec{x})$. $\square$

The above result depends upon and hence motivates conditions ($\cap$) and ($\uparrow$). It shows the class of symmetric sets is closed under any definable operation; however, the *hereditarily* symmetric sets are a bit more difficult, and require use of Lemma 2.20:

Corollary 2.22. $\mathcal{N} \models$ *Pairing and Union.*

Proof. If $x_1, x_2 \in \mathcal{N}$ then $\{x_1, x_2\}$ and $\bigcup x_1$ are the result of definable operations, so symmetric by Theorem 2.21, and subsets of the model (in the latter case by transitivity) so in the model by Lemma 2.20. $\square$

The following theorem motivates condition (N):

Theorem 2.23. *Any automorphism f of $\mathcal{M}$ restricts to an automorphism of $\mathcal{N}$.*

Proof. It suffices to show that if $x \in \mathcal{N}$ then $f(x) \in \mathcal{N}$. By $\in$-induction, it suffices to prove that if x is symmetric then so is $f(x)$ (for both x an atom and x a set). But if x is symmetric, then $\operatorname{stab}(x) \in \mathcal{F}$, so $\operatorname{stab}(f(x)) = f\operatorname{stab}(x)f^{-1} \in \mathcal{F}$ by (N). $\square$

Theorem 2.24. $\mathcal{N} \models$ *Power Set.*

Proof. $\mathcal{P}x$ is symmetric, by Theorem 2.21. It would then be convenient if it were a subset of $\mathcal{N}$, but unfortunately it isn't in general. However, by Theorem 2.23, automorphisms map $\mathcal{N}$ to itself, so an automorphism that stabilises $\mathcal{P}x$ will also stabilise $\mathcal{P}x \cap \mathcal{N}$, i.e. $\operatorname{stab}(\mathcal{P}x) \leq \operatorname{stab}(\mathcal{P}x \cap \mathcal{N})$, and so by ($\uparrow$), the latter is symmetric. It's trivially a subset of $\mathcal{N}$, and collects all the sets that $\mathcal{N}$ believes to be subsets of x, so it is a powerset for x in $\mathcal{N}$. $\square$

Theorem 2.25. $\mathcal{N} \models$ *Replacement (and therefore Separation).*

Proof. Take ϕ a formula in free variables x, y, and some auxiliary variables $\vec{t} = t_1 \dots t_n$. Suppose ϕ is functional on $\mathcal{N}$. Then, if $\psi = \phi^{\mathcal{N}} \wedge x \in \mathcal{N} \wedge y \in \mathcal{N}$, then ψ is functional on $\mathcal{M}$, so by replacement in $\mathcal{M}$, $\psi``x$ exists; by definition of ψ it is a subset of $\mathcal{N}$. It only remains to show that it is symmetric.

Take f an automorphism of $\mathcal{N}$ that fixes all the $\vec{t}$. Then

$$\psi(x, y, \vec{t}) \iff \psi(f(x), f(y), f(\vec{t})) \iff \psi(f(x), f(y), \vec{t}),$$

so in particular, f commutes with the function defined by ψ. Then

$$f(\psi``x) = f``\psi``x = \{f(\psi(y)) : y \in x\} = \{\psi(f(y)) : y \in x\}.$$

Now suppose f fixes x too. Then

$$y \in x \iff f(y) \in f(x) \iff f(y) \in x,$$

so the previous set is precisely $\{\psi(y) : y \in x\}$, which is $\psi``x$, so $f$ fixes $\psi``x$. Then

$$\operatorname{stab}(x) \cap \bigcap_{i=1}^{n} \operatorname{stab}(t_i) \leq \operatorname{stab}(\psi``x),$$

so $\psi``x$ is symmetric. $\square$

Combining all these results:

Theorem 2.26. $\mathcal{N}$ *is a model of ZFA.* $\square$

2.6 A model in which Choice fails

The choice of normal filter $\mathcal{F}$ determines the properties of the resultant model $\mathcal{N}$. Although the Axiom of Choice is largely irrelevant to the problem of naming, I will demonstrate this flexibility by proving that there is a permutation model in which the Well-ordering Theorem, and hence the Axiom of Choice, fails, so that $\neg AC$ is consistent with ZFA. In fact, the well-orderable sets of a permutation model have a simple characterisation in terms of the normal filter.

Definition 2.27. Given a group G of automorphisms of a model, and a set x of the model, the pointwise stabiliser $\mathrm{fix}_G(x)$ (or just $\mathrm{fix}(x)$)[16] is defined by

$$\mathrm{fix}_G(x) = \{f \in G : \forall y \in x\,[f(y) = y]\}.$$

Notice that $\mathrm{stab}(x) = \mathrm{fix}(\{x\})$, and that $\mathrm{fix}(x) \leq \mathrm{stab}(x)$.

Lemma 2.28. *Let x be a set in a model of ZFA, σ an automorphism, and f a function from a pure set α to x. Then $\sigma(f)$ is the function that maps $a \in \alpha$ to $\sigma(f(a))$.*

Proof. Since σ is an automorphism, $\sigma(\langle a, b\rangle) = \langle \sigma(a), \sigma(b)\rangle$. Now, $f = \{\langle a, f(a)\rangle : a \in \alpha\}$, so $\sigma(f) = \{\langle \sigma(a), \sigma(f(a)) : a \in \alpha\}$. But $\sigma(a) = a$ since a is pure, so $\sigma(f)$ maps a to $\sigma(f(a))$. ☐

Theorem 2.29. *A set $x \in \mathcal{N}$ that is well-orderable in $\mathcal{M}$ is well-orderable in $\mathcal{N}$ if and only if $\mathrm{fix}(x) \in \mathcal{F}$.*

Proof. Let f be a bijection from an ordinal α to x. I claim that $\mathrm{stab}(f) = \mathrm{fix}(x)$. Since ordinals are pure sets, by Lemma 2.28 $\sigma(f)$ maps $a \in \alpha$ to $\sigma(f(a))$. Hence if $\sigma(b) = b$ for all $b \in x$, certainly $\sigma(f) = f$, so $\mathrm{fix}(x) \leq \mathrm{stab}(f)$. Conversely, if $\sigma(f) = f$, then any b is $f(a)$ for some $a \in \alpha$, since f is a bijection, and $\sigma(f(a)) = f(a)$, so $\sigma(b) = b$. So $\mathrm{stab}(f) \leq \mathrm{fix}(x)$.

This proves the claim. Now, if x is well-orderable in the model, $\mathrm{stab}(f) \in \mathcal{F}$, so $\mathrm{fix}(x) \in \mathcal{F}$. Conversely, if $\mathrm{fix}(x) \in \mathcal{F}$, then since x is well-orderable in $\mathcal{M}$, there exists a bijection from an ordinal to it, and then this will be symmetric, and a subset of $\alpha \times x$ so a subset of $\mathcal{N}$, so by Lemma 2.20 it will be in the model. ☐

Corollary 2.30. *Let $\mathcal{M}$ be a model of ZFA with a countably infinite[17] set A of atoms, G the group of all permutations of A, and $\mathcal{F}$ generated by the set of stabilisers of finite sets of atoms, i.e. $H \in \mathcal{F}$ if and only if there is some finite $E \subseteq A$ such that H contains $\mathrm{fix}(E)$. Then the corresponding permutation model $\mathcal{N}$ contains A but no well-ordering of A, and hence $\mathcal{N} \models \neg AC$.*

[16] Jech's notation, [8, p. 46]; Felgner [4, p. 56] calls this $K[x]$.

[17] And therefore well-orderable!

Proof. First, the given $\mathcal{F}$ is indeed a normal filter: it is upward-closed by definition; suppose $H \supseteq \text{fix}(E)$ and $K \supseteq \text{fix}(F)$, then $H \cap K \supseteq \text{fix}(E \cup F)$; if $H \supseteq \text{fix}(E)$ then $fHf^{-1} \supseteq \text{fix}(f``E)$.

$\mathcal{N}$ contains A: A itself is symmetric, since $\text{stab}(A) = G = \text{fix}(\emptyset)$. All atoms are symmetric: $\text{stab}(a) = \text{fix}(\{a\})$. So A is hereditarily symmetric, hence in the model.

$\mathcal{N}$ does not contain $\text{fix}(A)$: $\text{fix}(A) = \{\text{id}\}$, which contains no subgroup other than itself, and certainly is not the pointwise stabiliser of a finite set. □

This completes the construction of a permutation model in which Choice fails.

3 Nominal structures via permutation models

3.1 The new improved recipe

Having shown the consistency and the power of permutation models, the objective is now to modify the set-theoretic construction of nominal structures given in section 1.3 to give it the symmetry properties required. I leave the guidance of Jech and Felgner for that of Gabbay and Pitts [7], although several aspects of the presentation are my own.

The construction will take place in a permutation model $\langle \mathcal{M}, G, \mathcal{F} \rangle$ with a countably infinite set of atoms A and G all permutations of A; I will leave $\mathcal{F}$ unspecified until the construction makes demands of it. It will be necessary to refer to the automorphisms of $\mathcal{M}$ more specifically; I will use the notation $(a\ b)$ to mean the permutation swapping the atoms a and b, and $(a\ b)(x)$ to mean the image of the set x under the automorphism induced by that permutation.

Revisiting the construction of Λ, given an name a and an expression x, one can either construct the term $\text{app}(\text{var}(a), x)$ or $\text{lam}(a, x)$. But from the perspective of naming, the latter is very different from the former! Clearly the constructor signature must be modified, and the modification I will use is to change the signature of lam to make it clear that rather than taking an arbitrary pair of name and expression, there is some interaction between the components: the name on the left is bound in the expression on the right.

To that end, I construct the *abstraction set* $[A]X$, consisting of *atom-abstractions* $a.x$, which satisfy the condition that if b is not otherwise bound in x, then $a.x = b.(a\ b)(x)$, so the name a inside x has no discernible identity – the only property of occurrences in x that is true under every such substitution is that it is distinct from the other bound variables, and equal to its binder. Once I have set up this context in which name-invariance holds as desired, the aim will be to find a recursion principle that allows informal reasoning, operating on concrete terms and names, to be safely transported into the abstract setting.

The first ingredient is a sufficiently abstract treatment of the notion of free variables and binding: in particular, the binding constructs will need to make the variables they bind no longer free and hence subject to the necessary renaming symmetries.

3.2 Free names and support

The idea that $\lambda a.a$ and $\lambda b.b$ should be considered the same, but a and b distinct, motivates recognising free variables as those whose identity forms part of the identity of the term: precisely those which, when permuted, change the identity of a term. Conversely, when some permutation of names fixes the free variables, the corresponding automorphism should fix the term itself.

Definition 3.1. A set $S \subseteq A$ is said to *support* a set x if $\mathrm{fix}(S) \leq \mathrm{stab}(x)$, that is to say, if some $\pi \in G$ fixes every element of S, then it fixes x too.

The idea is that the intersection of all supporting sets will be precisely the atoms 'relevant' to a set, those which, when moved, move x. However, that's not quite right: if x has only a single atom a in its transitive closure, then both $\{a\}$ and $A \setminus \{a\}$ will support x, and they are disjoint; there doesn't seem to be a 'least support' for this term. In general, terms and supports which are 'almost as large as A' suffer from pathology relating to the lack of sufficiently many unused variables, so that things can be fixed merely by virtue of there being nowhere to send them.

In my applications, terms do not need to be so large, so supporting sets are expected to be finite; restricting attention to only these finite cases finds them much better-behaved:

Lemma 3.2. *(a) If E and F are finite and support x, then so does $E \cap F$.*

(b) Hence, if a set has any finite supporting set, it has a least such.

Proof. (a) Suppose f fixes $E \cap F$. The idea is that f can be written as the product of permutations that alternately fix E and F, and so all fix x, so f fixes x. Suppose I can find a permutation g a product of permutations fixing E or F such that gf^{-1} fixes $E \cup F$. Then fg^{-1} fixes $E \cup F$, so $f = fg^{-1}g$ is of the required form.

To find g, apply f^{-1}, then apply transpositions to return elements of $E \cup F$ one by one. Let $x \in E \cup F$, say $x \in E$. Suppose $f^{-1}(x) \neq x$, so $x \notin E \cap F$, so $x \notin F$. Then if $f^{-1}(x) \notin F$, just apply the transposition $(x\ f^{-1}(x))$, which fixes F, to return x to its home. If, on the other hand, $f^{-1}(x) \in F$, then it is not in E, so pick an element of $A \setminus (E \cup F)$, say a, and apply the transposition $(f^{-1}(x)\ a)$, which fixes E, followed by $(a\ x)$, which fixes F.

(b) Since there is a finite supporting set, there is one of minimal size: let it be S. Let T be another finite supporting set. Then $S \cap T$ is a supporting set, and $S \cap T \subseteq S$, but S is of minimal size, so $S = S \cap T$, so $S \subseteq T$. □

I conjecture that if large terms were needed, then making the set of atoms still larger would suffice; I can offer no advice to those wishing to bind a proper class of variables, apart from possibly some fresh air and exercise.

In any case, the previous lemma justifies the following notation:

Definition 3.3. The *support*, denoted $\mathrm{supp}(x)$, of a finitely-supported set x is the $\subseteq$-least set among all finite sets that support x.[18]

This leaves one question unanswered: how to deal with the pathological sets with large support? Recall the permutation model in which Choice failed: it contained sets whose setwise stabiliser contained $\mathrm{fix}(E)$ for some finite set E. But $\mathrm{fix}(E) \leq \mathrm{stab}(x)$ is precisely the condition that x is supported by E, so in fact the model can be characterised as the sets hereditarily of finite support: it's the largest transitive model on which supp can be everywhere defined. This is the natural home for what follows, so the previously-unspecified $\mathcal{F}$ should be taken to be the normal filter generated by $\{\mathrm{fix}(E) : E \subseteq A, E \text{ finite}\}$.

It is worth mentioning that even though at no point have I required that the model believe A is a set (it is included as a unary predicate, a class, although it's entirely reasonable for the model to believe it is a set too), it is still possible to make sense of formulae like $x \in \mathcal{P}(A \times A)$ (x is a set of pairs such that all of its first and second projections satisfy the *is-an-atom* predicate) and hence to quantify over permutations of A. By appealing to $\in$-induction, the theory can even formalise its own automorphism action, and so it is possible to define supp as a function-class, although without further axioms it will not be possible to prove it is total. The axiom "supp is total" would suffice, but since I mostly deal in models rather than theory, the precise details are unimportant.

The following observation aids in the calculation of supports: the theory of ZFA is (by necessity!) incapable of talking about single specific atoms, so in particular, any operation definable in the theory cannot introduce new free variables:

Theorem 3.4. *If f is a definable operation in the sense of Theorem 2.11 on $\vec{x} = x_1, \ldots, x_n$, then*

$$\mathrm{supp}(f(\vec{x})) \subseteq \bigcup_{i=1}^{n} \mathrm{supp}(x_i).$$

[18] It's worth mentioning that this definition of supp is distinct from that given by Gabbay and Pitts, but proven equivalent in [7, Prop. 3.4].

Proof. The RHS supports $f(\vec{x})$, because if an automorphism fixes all the supports of the $\vec{x}$, then by Theorem 2.11 it will fix $f(\vec{x})$. Hence, by minimality of supp, the inclusion holds. ☐

3.3 Freshness and abstraction

Having established the notion of free variables via supports and a model in which this notion is applicable to every set, it is now possible to say what it means for a variable to be *not* free in a term.

Definition 3.5. For $a \in A$ and x a set, the predicate $a \# x$ ("a is *fresh* for x") means $a \notin \operatorname{supp}(x)$.[19]

The universal finite support property means that for any finite collection of variables, infinitely (indeed cofinitely many) atoms will be fresh for all of them – and since formulae are preserved under automorphisms, and any two fresh atoms may be swapped by an automorphism, all of them are theory-wise indistinguishable. Hence:

Theorem 3.6. *For ϕ a formula with free variables $a, \vec{x}$, the following are equivalent:*

(i) $\forall a \in A\, [a \# \vec{x} \implies \phi(a, \vec{x})]$,

(ii) $\phi(a, \vec{x})$ *for all but finitely many* a,

(iii) $\exists a \in A\, [a \# \vec{x} \wedge \phi(a, \vec{x})]$.

Proof. (i) $\implies$ (ii) since the precondition of (i) will be satisfied for a not in the union of the support of the $\vec{x}$, which is a finite union of finite sets, so finite.

(ii) $\implies$ (iii) since the set of atoms fresh for $\vec{x}$ is infinite.

(iii) $\implies$ (i): let b be an arbitrary atom satisfying $b \# \vec{x}$. Then $(a\ b)$ fixes the supports of all the $\vec{x}$, so it fixes all the $\vec{x}$. But by Theorem 2.10,

$$\phi(a, \vec{x}) \implies \phi((a\ b)(a), (a\ b)(\vec{x})) = \phi(b, \vec{x}).$$ ☐

This striking result led Gabbay and Pitts to invent a new quantifier:

Definition 3.7. The quantifier И means "for all but finitely many", so that for example Theorem 3.6(ii) can be written $\text{И}a \in A\, [\phi(a, \vec{x})]$.

[19] Again, this is equivalent but not identical to the definition originally given by Gabbay and Pitts; it is given as a characterisation of $\#$ in [7, Prop. 4.5], modulo the previously-mentioned equivalent definition of supp.

Corollary 3.8. *The* И *quantifier commutes with logical connectives, e.g. for* ϕ *and* ψ *formulae in free variables* $\vec{x}$*, ZFA proves*

$$\text{И}a \in A\,[\phi \iff \psi] \iff (\text{И}a \in A\,[\phi] \iff \text{И}a \in A\,[\psi])$$

Proof. Each logical connective commutes with at least one of $\forall$ and $\exists$, and by Theorem 3.6, the И quantifier can serve as either. $\square$

In any case, the promising theme is that concrete reasoning on a single atom can be lifted to abstract reasoning over all atoms, as long as the freshness precondition is checked.

With this concept of freshness, it's now possible to say a little more about the properties desired of the abstraction $a.x$: it should be the case that $a \# a.x$, and so for all $b \# x$, $(a\ b)(a.x) = a.x$. But if $a.x$ is to be a definable operation, then by the automorphism property, $(a\ b)(a.x) = b.(a\ b)(x)$, so it must be the case that $b \# x \implies a.x = b.(a\ b)(x)$. This suggests a couple of ways to proceed:

- Take the orbit of $\langle a, x\rangle$ under a suitable part of the permutation action, ensuring invariance, or
- Implement $a.x$ as a partial function on A that takes a fresh atom b and returns $(b\ a)(x)$ (thus 'forgetting' the identity of a).

Both of these approaches work; indeed both produce the same set, namely:

Definition 3.9. The *atom-abstraction* $a.x$ is defined by the equation:

$$a.x = \{\langle b, (b\ a)(x)\rangle : b \# x \vee b = a\}.$$

This is a definable operation, so automatically commutes with automorphisms by Theorem 2.11. It can be viewed as an equivalence class, relating two elements if they are mutually reachable by automorphism, or as a partial function, which is the view I tend to prefer. To avoid an excess of parentheses, I offer the following syntax for application in this case:

Definition 3.10. The *concretion* $a.x@b$ is the result of evaluating the partial function $a.x$ at b.

The next results show that it satisfies the properties expected of it with respect to support and invariance:

Lemma 3.11.

$$c \# x \implies a.x = c.(a\ c)(x)$$

Proof. Aim to show that $\langle b, y\rangle \in a.x \iff \langle b, y\rangle \in c.(a\ c)(x)$, for any atom b and set y. Split into three cases:

- $b = a$. In this case,

$$\begin{aligned} \langle a, y\rangle \in a.x &\iff & y = (a\ a)x &= x \\ \langle a, y\rangle \in c.(a\ c)(x) &\iff & y = (a\ c)(a\ c)(x) &= x. \end{aligned}$$

- $b = c$. Exactly as above.

- Otherwise, $(a\ c)(b) = b$. Now

$$b \# x \iff (a\ c)(b) \# (a\ c)(x) \iff b \# (a\ c)(x),$$

  and $b \# x \implies (b\ c)(x) = x$, so if $b \# x$ then

$$y = (b\ a)(x) \iff y = (b\ c)(c\ a)(b\ c)(x) \iff y = (b\ c)(c\ a)(x).$$

  So

$$\begin{aligned} \langle b, y\rangle \in a.x &\iff b \# x \wedge y = (b\ a)(x) \\ &\iff b \# (a\ c)(x) \wedge y = (b\ c)(a\ c)(x) \\ &\iff \langle b, y\rangle \in c.(a\ c)(x) \end{aligned}$$

  as required. □

Theorem 3.12. *The atom-abstraction $a.x$ satisfies* $\mathrm{supp}(a.x) = \mathrm{supp}(x) \setminus \{a\}$.

Proof. First, the given set supports $a.x$. By Theorem 3.4, $\mathrm{supp}(x) \cup \{a\}$ supports $a.x$. Let π be an automorphism that fixes $\mathrm{supp}(x) \setminus \{a\}$. If π also fixes a then it fixes $\mathrm{supp}(x) \cup \{a\}$, so it fixes $a.x$.

Suppose instead that π does not fix a. Then π does not fix $\pi(a)$, so $\pi(a) \# x$, so $\sigma = (a\ \pi(a)) \circ \pi$ is an automorphism that fixes $\mathrm{supp}(x) \cup a$, so $\sigma(a.x) = a.x$, so

$$\pi(a.x) = (a\ \pi(a))(\sigma(a.x)) = (a\ \pi(a))(a.x) = \pi(a).(a\ \pi(a))(x),$$

which is equal to $a.x$ by Lemma 3.11. So $\mathrm{supp}(x) \setminus \{a\}$ supports $a.x$.

Second, it is in fact the least support: by Theorem 3.4, the support of $a.x@a$ is contained in the union of the supports of $a.x$ and a; since $a.x@a = x$, $\mathrm{supp}(x) \subseteq \mathrm{supp}(a.x) \cup \mathrm{supp}(a)$, so $\mathrm{supp}(a.x)$ must contain all of $\mathrm{supp}(x)$ except possibly a. □

Having defined atom-abstractions as above, I will want to collect them in the set $\{a.x : a \in A, x \in X\}$. However, there is a problem with this set: take e.g. $X = \{a\}$, then the abstraction $a.a$ evaluated at $b \in A$ need not fall back in X. The problem is that abstraction followed by concretion results in permutation of the variables, so if it is to be well-behaved, X should be symmetric with respect to the relevant atoms. Hence the following definition:

Definition 3.13. The *abstraction set* $[A]X$ is the set of all atom-abstractions $a.x$ with $a \in A$, $x \in X$, and $a \# X$.

This forbids taking $a.a$ in the above example, but allows $b.a$ for any $b \neq a$, whose concretion at any atom gives $a \in X$ as expected. In practice I will often have X be a set with empty support, in which case this detail is irrelevant.

In order to 'unpack' elements of $[A]X$, there is the following useful lemma:

Lemma 3.14.

$$z \in [A]X \implies \text{И}a \in A\,[z = a.(z@a)]$$

Proof.

$$z \in [A]X \implies \exists a \in A\,[\exists x \in X\,[z = a.x]].$$

Since $a.x@a = x$ and $a \# a.x$, it follows that

$$z \in [A]X \implies \exists a \in A\,[a \# z \wedge z = a.(z@a)],$$

from which the conclusion follows by Theorem 3.6. $\square$

This abstraction set is the final ingredient necessary to revisit the construction of Λ in section 1.3.

3.4 Induction and recursion modulo alpha-equivalence

Tentative definition. Λ_α is the recursive data type over the constructors

$$\begin{aligned}
\mathrm{var} &: A \to \Lambda_\alpha \\
\mathrm{app} &: \Lambda_\alpha \times \Lambda_\alpha \to \Lambda_\alpha \\
\mathrm{lam} &: [A]\Lambda_\alpha \to \Lambda_\alpha
\end{aligned}$$

This definition poses two questions:

- How can we be sure that this constructor signature forms a set? Can we safely put an abstraction set in the domain of a constructor? Previously, I glossed over this issue: in the case where the domains are only sums and products of fixed sets and recursive occurrences of the data type, the existence question is 'obvious' or at least outside the scope of this essay. But if I were to put, say, $\mathcal{P}(\Lambda_\alpha)$ in the domain of a constructor, then there would be an injection from the power set of the type to the type, which is impossible by Cantor's theorem.[20] In order to characterise what sorts of operations are allowable, a more detailed construction will be necessary.

[20] Note that it is provable that there is no injection from $\mathcal{P}X$ to X without using either Choice or Foundation, so the proof goes through in ZFA, too.

- How do we define functions by recursion on this type? The natural recursion principle uses $f_{\mathrm{lam}} : [A]X \to X$, but how do we define functions on atom-abstractions? We'd much rather work with concrete names and terms, with functions $A \times X \to X$.

I'll address the second problem first, since it is simpler. The following lemma gives a condition under which concrete functions can be lifted to the abstraction sets.

Theorem 3.15. *If f is a function $A \times X \to X$, satisfying*

$$\text{И}a \in A\, [\forall x \in X\, [a \# f(a,x)]], \tag{F}$$

then there exists a unique function $\hat{f} : [A]X \to X$ satisfying

$$\text{И}a \in A\, [\forall x \in X\, [\hat{f}(a.x) = f(a,x)]].$$

Proof. Define $\hat{f}(z)$ by choosing an atom a fresh for f, z, and X; by Lemma 3.14 $z = a.x$ (where $x = z@a$); set $\hat{f}(z) = f(a,x)$. This gives:

$$\langle z, y\rangle \in \hat{f} \iff \exists a \in A\, [a \# f, z, X \wedge \exists x \in X\, [z = a.x \wedge y = f(a,x)]].$$

Then by (F) and Theorem 3.6, it follows that

$$\langle z, y\rangle \in \hat{f} \iff \text{И}a \in A\, [\exists x \in X\, [z = a.x \wedge y = f(a,x)]].$$

Now, since $a.x = a.x' \implies x = x'$ (take the concretion of both at a),

$$\exists x \in X\, [z = a.x \wedge \phi(x)] \implies \forall x \in X\, [z = a.x \implies \phi(x)],$$

so

$$\langle z, y\rangle \in \hat{f} \iff \text{И}a \in A\, [\forall x \in X\, [z = a.x \implies y = f(a,x)]].$$

Since И commutes with $\iff$ (Corollary 3.8), this is equivalent to

$$\text{И}a \in A\, [\forall x \in X\, [\hat{f}(a.x) = f(a,x)]],$$

as required (note that by moving from $\exists$ to $\forall$ and И, the arbitrary choices made initially have been eliminated, so $\hat{f}$ is indeed well-defined). □

The condition (F) is not so surprising: any f of the form $f(a,x) = a.g(a,x)$ will automatically satisfy it, and any *other* f is removing the binder for a, so it certainly ought to remove all free occurrences of a in x too: $f(\lambda a.a) = a$ is certainly not allowed!

With this result, functions on $[A]X$ are not too taxing to define, so the recursion principle for Λ_α is usable – if, of course, it is true! All that remains for me to do is prove that Λ_α exists and has the properties required. Let $L_\alpha(X)$ be the set operation that applies all constructors to all elements of X:

$$L_\alpha(X) = \mathrm{var}``A \cup \mathrm{app}``(X \times X) \cup \mathrm{lam}``[A]X$$

Λ_α, if it exists, had better be a fixed point of this operation (applying constructors to a lambda term gives a lambda term, and every lambda term is a constructor applied to a lambda term), and moreover should satisfy the following induction principle:

Theorem 3.16. *(Induction principle for Λ_α)*
If X is a set closed under applications of the constructors for Λ_α, then X contains all lambda terms. In other words,

$$L_\alpha(X) \subseteq X \implies \Lambda_\alpha \subseteq X$$

The following single characterisation of these two properties will be useful:

Lemma 3.17. *X is a fixed point of L_α satisfying the induction principle if and only if X is the least pre-fixed point of L_α (where X is a pre-fixed point of F means $F(X) \subseteq X$).*

Proof. The induction principle is equivalent to X being a lower bound for the class of pre-fixed points of L_α.

If X is a fixed point for L_α, then certainly it is pre-fixed, so since it is a lower bound, it is the least pre-fixed point.

If X is the least pre-fixed point, then it is a lower bound for the pre-fixed points, and moreover it is a fixed point: observe that L_α is monotonic, so that not only is $L_\alpha(X) \subseteq X$ true, but $L_\alpha(L_\alpha(X)) \subseteq L_\alpha(X)$, so $L_\alpha(X)$ is pre-fixed, so $X \subseteq L_\alpha(X)$, so they are equal. □

So I am looking for a least pre-fixed point of L_α. Rather than making a vague appeal to Knaster-Tarski as before, I will take a more hands-on approach.[21] Define the first attempt $X_0 = \emptyset$; it at least satisfies the "least" part of "least pre-fixed point". However, it's not pre-fixed, because the $\mathrm{var}``A$ part of $L_\alpha$ throws in all the atoms as variables, and $X_0$ doesn't have those. Okay, second attempt, $X_1 = \mathrm{var}``A = L_\alpha(X_0)$. But X_1 contains terms, so it should contain the applications of those terms: try $X_2 = L_\alpha(X_1)\ldots$

It's clear that each λ-term can be assigned a *rank*, the minimal n for which it is found in X_n, and it's also clear that the application of two terms of rank n will

[21] This approach is very standard in computer science, but perhaps less familiar to set theorists.

be of rank $n+1$. So no X_n will ever be sufficient. What about taking the union, in the hope of gathering the set of all elements of all ranks?

$$X_\omega = \bigcup_{n<\omega} X_n$$

For constructor signatures that are in some sense "finitary", this is exactly the right thing. Here's why:

Lemma 3.18. *Suppose F is a monotonic set operation such that F commutes with F^ω, defined by $F^\omega(X) = \bigcup_{n<\omega} F^n(X)$. Then the least pre-fixed point $\mu(F)$ of F is $F^\omega(\emptyset)$.*

Proof. $\mu(F)$ is indeed a fixed point: F commutes with F^ω, so

$$\begin{aligned} F(\mu(F)) &= F(\textstyle\bigcup_{n<\omega} F^n(\emptyset)) \\ &= \textstyle\bigcup_{n<\omega} F^{n+1}(\emptyset) \\ &= \textstyle\bigcup_{0<n<\omega} F^n(\emptyset) \end{aligned}$$

The $n = 0$ term is empty, so omitting it makes no difference:

$$\begin{aligned} &= F^\omega(\emptyset) \\ &= \mu(F). \end{aligned}$$

$\mu(F)$ is least: I show by induction on n that any pre-fixed point must contain $F^n(\emptyset)$, so then it must contain them all, and so must contain their union. The case $n = 0$ is trivial. If Z is a pre-fixed point of F that contains $F^k(\emptyset)$, then by monotonicity, $F^{k+1}(\emptyset) \subseteq F(Z) \subseteq Z$, as required. $\square$

The "finitary" condition to which I referred before corresponds to the preservation of F^ω: if each constructor uses only finitely many terms from $\bigcup_{n<\omega} F^n(\emptyset)$, then there is some maximum rank N that it uses, and then it will have rank $N+1$ and appear in the union itself.

The crucial question then becomes: is L_α monotonic, and does it preserve the union of its finite iterates? The answer, fortunately, is yes:

Lemma 3.19. *If F and G are monotonic set operations, then so is the operation $X \mapsto F(X) \cup G(X)$.*

Proof. Let $X \subseteq Y$. Then $F(X) \subseteq F(Y)$ and $G(X) \subseteq G(Y)$, so $F(X) \cup G(X) \subseteq F(Y) \cup G(Y)$. $\square$

Lemma 3.20. *The abstraction-set operation is monotone:*

$$X \subseteq Y \implies [A]X \subseteq [A]Y.$$

Proof. Let $z \in [A]X$.

$$\begin{aligned} & \text{И} a \in A\,[\exists x \in X\,[z = a.x]] \\ \Longrightarrow\ & \text{И} a \in A\,[\exists x \in Y\,[z = a.x]] \\ \Longrightarrow\ & \exists a \in A\,[a \# Y \wedge \exists x \in Y\,[z = a.x]] \\ \Longrightarrow\ & z \in [A]Y. \end{aligned}$$

where the first implication uses the fact that И commutes with $\Rightarrow$ (Corollary 3.8). □

Lemma 3.21. *L_α is monotone, and preserves L_α^ω.*

Proof. Monotonicity follows from Lemma 3.19 and Lemma 3.20, and the monotonicity of $X \mapsto \text{var}``A$ (trivial) and $X \mapsto \text{app}``(X \times X)$ (not much harder).

As for the preservation, let $x \in L_\alpha(\bigcup_n X_n)$, where $X_n = L_\alpha^n(\emptyset)$. Then either:

- $x = \text{var}(a)$ for $a \in A$. Then $x \in X_1$.
- $x = \text{app}(y, z)$ for $y, z \in \bigcup_n X_n$, say $y \in X_{n_y}$, $z \in X_{n_z}$. Then $x \in X_{\max\{n_y, n_z\}+1}$.
- $x = \text{lam}(z)$ for $z \in [A](\bigcup_n X_n)$. Then

$$\begin{aligned} & \text{И} a \in A\,[\exists x \in \bigcup_n X_n\,[z = a.x]] \\ \Longrightarrow\ & \text{И} a \in A\,[\exists n < \omega, x \in X_n\,[z = a.x]] \end{aligned}$$

  so $z \in X_{n+1}$.

In any of these cases, $x \in \bigcup_n X_n$. □

Theorem 3.22. *L_α has a least pre-fixed point*

$$\Lambda_\alpha = \mu(L_\alpha) = L_\alpha^\omega(\emptyset)$$

Proof. Lemma 3.18 applied to Lemma 3.21. □

So there exists a set closed under the constructors and satisfying induction! Recursion holds too:

Theorem 3.23. *Given functions*

$$\begin{aligned} f_{\text{var}} &: A \to R \\ f_{\text{app}} &: R \times R \to R \\ f_{\text{lam}} &: [A]R \to R, \end{aligned}$$

there is a unique function $f : \Lambda_\alpha \to R$, satisfying:

$$\begin{aligned} \forall a \in A\,[f(\mathrm{var}(a)) = f_{\mathrm{var}}(a)] \\ \forall s, t \in \Lambda_\alpha\,[f(\mathrm{app}(s,t)) = f_{\mathrm{app}}(f(s), f(t))] \\ \text{И}a \in A\,[\forall t \in \Lambda_\alpha\,[f(\mathrm{lam}(a.t)) = f_{\mathrm{lam}}(a.f(t))]] \end{aligned}$$

Proof. Exactly as for Λ in Theorem 1.1, except that it is necessary to show that the recursion equation for lam specifies a unique behaviour for f. The key is that any partial attempts to define f will be finitely supported, and hence $f((a\ b)(t)) = (a\ b)(f(t))$ for all but finitely many a, b pairs. Hence the above specification does not depend on the choice of atom to represent the atom-abstraction, except that it avoids a finite collection of finite supports. $\square$

Now we can finally complete the definition of $f(E) = E[x \mapsto F]$:

$$\begin{aligned} f_{\mathrm{var}}(a) &= \begin{cases} F & \text{if } a = x \\ a & \text{otherwise} \end{cases} \\ f_{\mathrm{app}}(r_1, r_2) &= \mathrm{app}(r_1, r_2) \\ f_{\mathrm{lam}}(z) &= \mathrm{lam}(z) \end{aligned}$$

This definition is the trophy of this essay and the technique of nominal sets: vastly simpler than the previous attempt, and *automatically* correct: there simply *cannot* be any problematic interactions between x and lambda-bound variables, since all lambda-bound variables have no single atom as their identity.

As an example that requires inspecting lambdas, the set of free variables of a term:

$$\begin{aligned} f_{\mathrm{var}}(a) &= \{a\} \\ f_{\mathrm{app}}(r_1, r_2) &= r_1 \cup r_2 \\ f_{\mathrm{lam}}(z) &= \hat{g}(z) \text{ where } g(a, v) = v \setminus \{a\} \end{aligned}$$

The hatting of g here is as given by Theorem 3.15, and is legitimate because $v \setminus \{a\}$ supports itself and does not contain a, so $a \# g(a, v)$ for all v.

3.5 Closing remarks

The highly symmetric structure of the permutation model, giving rise to the И quantifier and its uses, makes sound and formally correct what computer scientists have informally known all along, that reasoning about individual members of an α-equivalence class can be easily and safely transported to reasoning about the entire class, provided certain simple conditions are checked. The reasoning is transported by the rich collection of automorphisms made available by working in set theory modified to include atoms, and applies particularly to the submodel of such a theory given by a Fraenkel-Mostowski permutation model where every set is symmetric with respect to all but finitely many of the atoms.

3.5.1 Further reading

The following are resources that the scope of my essay or the time I had to write it prevented me from including:

The topic of FM sets has many pure-mathematical applications that predate those in computer science discussed here, and has been discussed at length by e.g. J.K. Truss in [12], Andreas Blass in [2], and many others.

Extending the work on nominal sets, Gabbay has developed a theory of infinitary nominal structures in [6], replacing "finite" as a criterion for good behaviour with "well-orderable". Meanwhile, Pitts is shortly to publish a book [11] titled 'Nominal Sets'.

There has been work to formalise the nominal techniques in theorem-proving software: [13] describes a modification that makes the technique compatible with the Axiom of Choice, rendering it amenable to formalisation in Isabelle/HOL.

FM models have found further unexpected applications in database theory and automata theory; [3] features applications where the atoms have relational structure, in contrast to the strictly uniform atoms of this essay.

3.5.2 Acknowledgements

I am grateful to Dr. Thomas Forster, for starting the reading group on FM models which introduced me to them, and more specifically for the observations on stratification and definable operations that made the proofs of existence for models of ZFA and permutation models much less painful.

I am grateful, too, to all the speakers of that reading group for their insights, and particularly to Steffen Lösch for introducing the group to the nominal sets technique.

I am particularly grateful to Andrew Pitts at the Computer Laboratory, for responding promptly and helpfully to the enquiries of a confused reader of a paper he wrote over a decade ago.

References

[1] H.P. Barendregt. *The lambda calculus: its syntax and semantics.* Studies in logic and the foundations of mathematics. North-Holland Pub. Co., 1981.

[2] Andreas Blass. Partitions and permutation groups. *Model Theoretic Methods in Finite Combinatorics: AMS-ASL Joint Special Session, January 5-8, 2009, Washington, DC*, 558:453, 2011.

[3] Mikołaj Bojańczyk and Sławomir Lasota. Fraenkel-Mostowski sets with non-homogeneous atoms. In Alain Finkel, Jérôme Leroux, and Igor Potapov, editors, *Reachability Problems*, volume 7550 of *Lecture Notes in Computer Science*, pages 1–5. Springer Berlin Heidelberg, 2012.

[4] U. Felgner. *Models of ZF-Set Theory*. Lecture Notes in Mathematics. Springer, 1971.

[5] T.E. Forster. Fränkel-Mostowski models: Notes for a reading group. Unpublished, circulated via the internet, October 2012.

[6] Murdoch J. Gabbay. A general mathematics of names. *Information and Computation*, 2007.

[7] Murdoch J. Gabbay and Andrew M. Pitts. A new approach to abstract syntax with variable binding. *Formal Aspects of Computing*, 13(3-5):341–363, 2002.

[8] T.J. Jech. *The Axiom of Choice*. Dover Books on Mathematics Series. Dover Publications, 2008.

[9] S. C. Kleene. Disjunction and existence under implication in elementary intuitionistic formalisms. *The Journal of Symbolic Logic*, 27(1):pp. 11–18, 1962.

[10] Lawrence S. Moss. Non-wellfounded set theory. In Edward N. Zalta, editor, *The Stanford Encyclopedia of Philosophy*. Fall 2009 edition, 2009.

[11] Andrew M. Pitts. *Nominal Sets*. Cambridge Tracts in Theoretical Computer Science. Cambridge University Press, 2013.

[12] John Truss. Permutations and the axiom of choice. *Automorphisms of first order structures*, pages 131–152, 1994.

[13] Christian Urban and Christine Tasson. Nominal techniques in isabelle/hol. In Robert Nieuwenhuis, editor, *Automated Deduction – CADE-20*, volume 3632 of *Lecture Notes in Computer Science*, pages 38–53. Springer Berlin Heidelberg, 2005.