The problem of the consistency of Quine's New Foundations

M. Randall Holmes
Boise State University

December 8, 2015

The Backstory

New Foundations is a variation of a simple theory of types.

I would like to say a variation of *the* simple theory of types, but at least three different theories are called "the simple theory of types".

The simple theory of types that is relevant is the simple typed theory of sets (rather than the simple typed theory of relations due to Ramsey or the simple typed theory of functions due to Church).

The simple typed theory of sets (TST)

This is a first-order theory with equality and membership with sorts of object indexed by the natural numbers.

The rules of formation of atomic sentences are neatly summed up by the templates $x^n = y^n$; $x^n \in y^{n+1}$.

A question as to whether members of different types are equal or whether a member of a type is an element of a non-successor type is not answered in the negative but dismissed as ungrammatical.

Axioms...

The axioms of TST come in two sets.

Extensionality axioms say that objects of type $i+1$ are equal iff they have the same type i elements. This expresses the identity criterion we expect for sets.

Comprehension axioms assert that for any formula $\phi(x^i)$ of our language, the set $\{x^i : \phi(x^i)\}^{i+1}$ exists: any property of a type i object that we can express defines the extension of some type $i+1$ object.

One often adds axioms of Infinity and Choice.

A little more history

I often encounter misattributions of this to Russell. This is not the type theory of Whitehead and Russell, *Principia Mathematica*. Russell described something like TST informally in his Principles of Mathematics, 1903, but he could not implement it in his major work because he did not know how to represent ordered pairs as sets (this was discovered by Norbert Wiener in 1914). The type system of PM is complicated, both because it contains a quite complex system of types of n-ary relations with arguments with any combination of different types, and because it implements predicativity restrictions which are then awkwardly evaded by an Axiom of Reducibility.

It appears that TST was first described by Gödel, Tarski, and Carnap around the same time, c. 1930.

A little math in TST

We will do a little math in TST. This will help us to illustrate the point which caused Quine to suggest New Foundations.

We will define the natural numbers following Frege: the number three for example is the set of all sets with three elements.

We illustrate why this definition is not circular.

The set of all sets with zero elements is of course the singleton of the empty set.

$\emptyset = \{x : x \neq x\}$ is of course definable (we do not adorn expressions with type indices unless we have to: here it is clear that if x is of any type i, we can define an empty set in type $i + 1$).

$0 = \{x : x = \emptyset\}$ then defines 0 as the set of type $i+2$ whose only element is the empty set of type $i + 1$.

For any sets a and b of the same type, we define $a \cup b$ as $\{x : x \in a \vee x \in b\}$ as usual. For any x of a given type, we define $\{x\}$ of the next type as $\{y : y = x\}$.

For any set A, we define $\sigma(A)$ as

$$\{b : (\exists ax.a \in A \wedge x \notin a \wedge b = a \cup \{x\})\}$$

We might be called to account on this one, so we decorate it.

$$\sigma(A^{i+2}) = \{b^{i+1} : (\exists a^{i+1}x^i.a^{i+1} \in A^{i+2}$$

$$\wedge x^i \notin a^{i+1} \wedge b^{i+1} = a^{i+1} \cup \{x^i\}^{i+1})\}$$

Of course, if one had to write all these indices to do anything in TST, the theory would be too cumbersome to use (and this objection has been raised) but this is not really needed in practice.

Now observe that $\sigma(0)$ is the set of all sets with one element, which we may call 1, $\sigma(1)$ is the set of all sets with two elements which we may call 2, and $\sigma(2)$ is the set of all sets with three elements, which we may call 3. And so forth.

We say that a set I is inductive iff $0 \in I \wedge (\forall A.n \in I \rightarrow \sigma(n) \in I)$.

We define $\mathbb{N}$ as the intersection of all inductive sets:

$$\mathbb{N} =$$

$$\{n : (\forall I.[0 \in I \wedge (\forall A.m \in I \rightarrow \sigma(m) \in I)] \rightarrow n \in I\}$$

Hints about doing more math

The ordered pair (x, y) can be defined as $\{\{x\}, \{x, y\}\}$ as usual, and relations and functions can be defined as usual as sets of ordered pairs. If x and y are of type i, (x, y) [if defined in this way] is of type $i+2$, and so the typing of relation sentences $x\,R\,y$ and function expressions $f(x)$ would be $x^i\,R^{i+3}\,y^i$ and $f^{i+3}(x^i)$, which might seem a little odd. But if Infinity is assumed it is possible to define the ordered pair in such a way that it is the same type as its projections and change the types $i+3$ to $i+1$ above.

Sets being the same size (equinumerous) can be defined as usual in terms of the existence of bijections, and the cardinal of a set can be defined as its equivalence class (one type higher) under equinumerousness. Similarly, the ordinal numbers can be defined as equivalence classes of well-orderings under isomorphism.

The hall of mirrors

The feature of TST which led Quine to propose New Foundations has been pervasive in our work so far. This is the polymorphism of the system. Notice that we have succeeded in defining natural numbers 0,1,2... in each type $i+2$.

Russell noticed the same feature in his system of PM; he called it "systematic ambiguity". But in TST it takes an especially simple form.

For any formula ϕ, let ϕ^+ be obtained by replacing each variable in ϕ with a variable one type higher, in an injective manner.

For any axiom ϕ, ϕ^+ is also an axiom. The $+$ operation commutes with all our rules of inference. Thus, if ϕ is a theorem, so is ϕ^+.

If we define a mathematical object of type i using an expression $\{x^i : \phi(x^i)\}^{i+1}$, we can raise all types to get an object $\{x^{i+1} : \phi^+(x^{i+1})\}^{i+2}$ in the next type up.

Both of these effects mirror theorems or defined objects in a given type in all higher types.

New Foundations

In his paper "New foundations for mathematical logic" (1937) the American philosopher W. v. O. Quine proposed a radical cure for the hall of mirrors issue. He suggested that we should simply identify all the types, and take as our axioms the statements obtained from axioms of TST by dropping all type distinctions (without introducing identifications between variables).

So New Foundations (NF), as it is generally called after the title of the paper in which it was introduced, is the first-order theory with equality and membership whose axioms are

1. A single axiom of extensionality (when type distinctions between variables are dropped, all the extensionality axioms become the same assertion)

2. An axiom which may be informally stated "$\{x : \phi(x)\}$ exists", or more precisely

$$(\exists A.(\forall x.x \in A \leftrightarrow \phi(x))),$$

where A does not appear in ϕ, for each formula $\phi(x)$ obtained by dropping type distinctions between variables in a formula of TST without introducing new identifications between variables.

Notice that the axioms introduced in the second clause do *not* include contradictory assertions such as "$\{x : x \notin x\}$ exists", because $x \notin x$ is not obtainable from a well-formed formula of TST by dropping type distinctions between variables.

Stratification

It is usual to define the comprehension axiom scheme of NF in a way which does not appeal to the language of another theory.

We say that a formula ϕ is stratified iff there is a function σ from variables to natural numbers such that for each subformula $x = y$ of ϕ we have $\sigma(x) = \sigma(y)$ and for each subformula $x \in y$ of ϕ we have $\sigma(x) + 1 = \sigma(y)$. Such a function is called a stratification of ϕ.

The axiom scheme of stratified comprehension asserts that for each stratified formula $\phi(x)$, $\{x : \phi(x)\}$ exists.

It should be clear that this is exactly the same comprehension axiom proposed above.

Surprising (but harmless) features of NF

The formula $x \notin x$ whose extension is the Russell class is not stratified, but the formula $x = x$ is stratified and $V = \{x : x = x\}$ is a set. Statements like $V \in V$ are true (by convention, we can write this statement in the language of TST and say that it is true, by a sort of pun – we interpret the V's as the extensions of successive types, and the pun can be dispelled by writing $V^{i+1} \in V^{i+2}$, but in NF the two V's have the same reference). But $x \in x$ is not the sort of predicate which has an extension.

The universe is a Boolean algebra in NF, as sets have complements as well as unions and intersections.

The natural numbers can be defined following Frege (using our exact development above) and the number 3 is genuinely the set of *all* sets with three elements.

Cardinal numbers can be defined as equivalence classes of sets under equinumerousness defined as usual. Ordinal numbers can be defined as equivalence classes of well-orderings under isomorphism.

We review the classical paradoxes of set theory.

The Russell paradox is blocked because $x \notin x$ is not a stratified formula.

The Cantor paradox is blocked in a subtler way. Clearly $\mathcal{P}(V) = V$, so it cannot be the case that $|A| < |\mathcal{P}(A)|$ in general. But notice that this is an ill-typed assertion, making no sense in the language of TST. The Cantor theorem of TST asserts that $|\iota``A| < |\mathcal{P}(A)|$, the collection of one element subsets of $A$ is strictly smaller than the power set of $A$. The proof is quite standard, with an adjustment for type discipline. The specialization of this proof to the case of the universe asserts that $|\iota``V| < |\mathcal{P}(V)|$, the collection of singletons is strictly smaller than the set of sets. The obvious external bijection $x \mapsto \{x\}$ is therefore not a set, and there is no paradox in this as the obvious comprehension axiom which would give its existence is not stratified, so our axioms do not require that it be a set.

The Burali-Forti paradox is handled similarly and has if anything an even more curious appearance. An ordinal is defined as an equivalence class of well-orderings under isomorphism. There is a natural well-ordering on the ordinals whcih has order type Ω. There is a natural order type on the ordinals less than Ω, which we might suppose is Ω, leading to paradox, as a well-ordering cannot have an initial segment similar to itself. But the assertion that the order type of the natural order on the ordinals restricted to ordinals less than α is α does not make sense in type theory. If α is the order type of $\leq$, let $T(\alpha)$ be the order type of $\{(\{x\},\{y\}) : x \leq y\}$: this is one type higher in TST. The theorem we can prove in TST (if we use the Kuratowski pair) is that the order type of the ordinals less than α is $T^4\alpha$ (this can be changed to $T^2\alpha$ if we use a type level pair). The specialization to Ω tells us that $T^4\Omega < \Omega$. T respects order, so it follows that there is a “descending sequence” $\Omega, T^4(\Omega), T^8(\Omega), T^{12}(\Omega) \ldots$ of ordinals which is not a contradiction because the map T cannot be shown to be a function in any obvious way (and indeed by these considerations is proved not to be a function).

Nothing here shows that NF does not fall prey to some argument of the same general type. However, it can be shown that it does not. A “minor modification” of NF exhibits all these features and is demonstrably as reliable as familiar set theories. This is the next chapter in our story.

A useful and not surprising theorem of Specker (1962)

Specker showed in 1962 that NF is equiconsistent with TST + the ambiguity scheme, which asserts $\phi \leftrightarrow \phi^+$ for each sentence ϕ. This seems natural given the motivation of NF, but it was nice to have it nailed down. A model of TST + Ambiguity gives us a model of NF satisfying the same sentences with types dropped.

The consistency of NFU (1969)

Jensen showed in 1969 that if one modifies NF by weakening extensionality so as to allow urelements, the resulting theory NFU is consistent.

Here is a version of the argument. Let α_i be any increasing sequence of ordinals indexed by the natural numbers. Interpret type i as V_{α_i} (the α_i'th level of the cumulative hierarchy). Interpret $x^i \in y^{i+1}$ as $x \in V_{\alpha_i} \wedge y \in V_{\alpha_i+1} \wedge x \in y$. Notice that this makes all of the elements of $V_{\alpha_{i+1}} \setminus V_{\alpha_i+1}$ into urelements.

Let Σ be a finite set of formulas of the language of TST(U). Let n be a strict upper bound on the type indices appearing in Σ. Partition the n element sets A of ordinals by considering the truth values of the sentences in Σ in models determined by a sequence s where

$s\lceil n$ has range A. This partition has an infinite homogeneous set H: choose an increasing sequence h whose range is included in H, and the model of TSTU determined by h as above will satisfy ambiguity for the formulas in Σ.

By compactness the full ambiguity scheme is consistent, and by Specker's results NFU is consistent (they do adapt to TSTU). Because the model of NFU will satisfy the same well typed sentences as the model of TSTU, it follows that mathematics in NFU is really quite normal. It is consistent with such things as Infinity and Choice, for example.

NFU exhibits all of the surprising features of NF which I exhibited above. This is why I say they are harmless: all of these phenomena can be seen to occur in a theory which is consistent and whose model theory is well understood.

Things get weird (1954)

Specker showed (before his very nice result of 1962) the much more unpleasant and disturbing result that NF disproves the Axiom of Choice (and incidentally proves Infinity). This rather caused the bottom to fall out of the NF market (it is rather sad that in the same year Rosser published his *Logic for mathematicians*, a really nice book on the foundations of mathematics which unfortunately used NF...)

We outline the proof.

Define the exponential map exp so that $\exp(|\iota``A|) = |\mathcal{P}(A)|$. (this is a way to get a map that makes sense in TST and so in NF). $\exp(\kappa) > \kappa$ for the usual reasons.

We consider the set M of all cardinals κ such that for some natural number n, $\exp^n(\kappa)$ is undefined (exp is undefined for cardinals greater than $|\iota``V|$). This is certainly nonempty so by Choice it has a smallest element μ.

For any cardinal $\kappa = |A|$, define $T\kappa$ as $|\iota``A|$.

Let $\exp^m(\mu)$ be the largest power of μ that is defined. $T(\exp^m(\mu)) = \exp^{Tm}(T\mu)$ is less than $|\iota``V|$ and greater than $|\iota^2``V|$, so it has either one or two more defined iterated exponentials (because $\exp(|\iota^{n+1}``V|) = |\iota^n``V|$ for each n – these are internal representations of the cardinalities of successive types) – and belongs to the same set M, so $\mu \leq T(\mu)$. Thus $T^{-1}(\mu) \leq \mu$, and a similar argument shows that $T^{-1}(\mu)$ has finitely many defined images under exp, so $T^{-1}(\mu) \leq \mu$, so $\mu = T\mu$. But then $m = Tm + 1$ or $Tm + 2$, which is absurd. It is not necessarily the case that $Tm = m$ (that is another story) but certainly they have the same remainder mod 3.

I don't necessarily expect my Audience to follow this argument. But I do want you to see that it is very strange.

This argument hinges on the fact that in NF, $\exp(|\iota``V|) = |V|$. In NFU with choice, $\exp(|\iota``V|) = |\mathcal{P}(V)|$, the cardinality of the set of sets, which is much smaller than $|V|$ (most of the universe consists of urelements).

Jensen's argument adapted to NF – tangled type theory

We present a theory which we presented in 1995 and showed to be equiconsistent with NF.

TTT is a first order theory with linearly ordered sorts (we may suppose the natural numbers) with the rules of formation $x^m = y^n$ well formed iff $m = n$ and $x^m \in y^n$ well-formed iff $m < n$.

Let s be an increasing sequence of types and let ϕ be a formula of TST. ϕ^s is the wff of TTT one gets if one replaces each type n in ϕ with s_n. The axioms of TTT are exactly the formulas ϕ^s such that ϕ is an axiom of TST.

If NF is consistent, we get a model of TTT by using the model of NF as each type (or disjoint copies of the model of NF if you prefer) and the membership of the model of NF for membership of each type in each higher type.

Suppose we have a model of TTT.

Let Σ be a finite set of formulas of the language of TST. Let n be a strict upper bound on the type indices appearing in Σ. Partition the n element sets A of types by considering the truth values of the sentences in Σ in interpretations of TST in our model of TTT determined by a sequence s where $s\lceil n$ has range A. This partition has an infinite homogeneous set H: choose an increasing sequence h whose range is included in H, and the model of TST determined by h as above will satisfy ambiguity for the formulas in Σ.

By compactness the full ambiguity scheme is consistent, and by Specker's results NF is consistent.

TTT unfolded – tangled webs of cardinals

TTT is an extremely strange theory. Each type is being interpreted as a power set of each lower type simultaneously. These power sets cannot be honest. But it is not any longer asserting that a certain set is a power set of *itself* (as NF seems to do), and with a little work it can be adapted to something we might imagine could be implemented in ordinary set theory (necessarily without choice).

Partly by examining the way TTT sees its own internal type structure, I developed (in the same 1995 paper) the following formulation of a property of a system of cardinals which if it could be realized in (say) Zermelo set theory would entail consistency of NF.

A natural model of TST is a model in which each type $i+1$ is implemented by a set the same size as the power set of the set implementing type i and every subset of the set implementing type i is the extension of an element of type $i+1$ in the model. The first order theory of a natural model of TST is completely determined by the cardinality of type 0.

A tangled web of cardinals is a function τ from nonempty finite subsets of a limit ordinal λ to cardinals, with the following properties:

1. $2^{\tau(A)} = \tau(A \setminus \{\mathtt{min}(A)\}$ (if A has at least two elements).

2. The first-order theory of the first n types of a natural model with base type of size $\tau(A)$ is determined by the smallest n elements of A.

Let Σ be a finite set of formulas of the language of TST. Let n be a strict upper bound on the type indices appearing in Σ. Partition the n element sets A of λ by considering the truth values of the sentences in Σ in natural models of TST with base type of size $\tau(A)$. This partition has an infinite homogeneous set H: choose a subset B of size $n+1$ of H, and a natural model of TST with base type of size $\tau(B)$ will satisfy ambiguity for the formulas in Σ.

By compactness the full ambiguity scheme is consistent, and by Specker's results NF is consistent.

NF itself doesn't prove the existence of tangled webs, though it proves the existence of as large a concrete fragment as one might want of a system of cardinals which is tangled for a concrete finite set of formulas. Moderately strong extensions of NF do show the existence of closer approximations to tangled webs.

My reason for defining these was that they represent a kind of structure one might suppose to exist in ZF without choice (eight types of TTT disprove choice by a modification of Specker's method): if one can show them to exist in a model of ZF, one has solved the problem of consistency of NF, which has been open since 1937 and a matter of some concern since Specker's strange results of 1954.

Preparations for the proof of consistency

We will work in ZFA: this is Zermelo-Fraenkel set theory with extensionality weakened to allow a set of atoms. In our ambient ZFA we will assume the axiom of choice, but we will carry out a Fraenkel-Mostowski construction of a class submodel of our ambient ZFA in whcih there will be a tangled web of cardinals. It is worth noting that nothing like the full strength of ZFA is needed to carry out this argument.

A parameter of the construction

We fix an uncountable regular cardinal κ for the rest of the paper. We refer to all sets of size $< \kappa$ as small and all other sets as large.

The Fraenkel-Mostowski method

Any permutation π of the set of atoms is extended to all sets by the rule $\pi(A) = \pi``A$.

Let G be a group of permutations of the atoms. Let Γ be a subset of the collection of subgroups of G with the following properties:

1. The subset Γ contains all subgroups J of G such that for some $H \in \Gamma$, $H \subseteq J$.

2. The subset Γ includes all subgroups $\bigcap C$ of G where $C \subseteq \Gamma$ and C is small.

3. For each $H \in \Gamma$ and each $\pi \in G$, it is also the case that $\pi H \pi^{-1} \in \Gamma$.

4. For each atom a, $\mathtt{fix}_G(a) \in \Gamma$, where $\mathtt{fix}_G(a)$ is the set of elements of G which fix a.

A nonempty Γ satisfying the first three conditions is what is called a κ-complete normal filter on G.

We call a set A Γ-symmetric iff the group of permutations in G fixing A belongs to Γ. The major theorem which we use but do not prove here is the assertion that the class of hereditarily Γ-symmetric objects (including all the atoms) is a class model of ZFA (usually not satisfying Choice: showing independence of Choice was the original application). The assumption that the filter is κ-complete is not needed for the theorem ("finite" usually appears instead of "small"), but it does hold in our construction.

Overview of clans, litters, and allowable permutations

For certain large sets P we will postulate the existence of sets $\texttt{clan}(P)$ of atoms (called clans). Every atom will belong to a clan. It is important to notice that elements of sets P are not necessarily pure sets.

- There will be a bijection f_P from $P \times \kappa$ to $\texttt{clan}(P)$. We introduce the notation a^P_α for $f_P(a, \alpha)$. Mention of P might be omitted from the notation when understood from context.

- We define $\texttt{litter}^P(a)$ as $\{a^P_\alpha : \alpha < \kappa\}$ and call such sets litters. We define a near-litter as a subset of a clan which has small symmetric difference from a litter. Mention of P might be omitted from the notation when understood from context. The anomalous elements for a near-litter are the elements of the small symmetric difference between it and a litter (they are not necessarily elements **of** the near-litter!).

• We stipulate that if P and Q are distinct and $\mathtt{clan}(P)$ and $\mathtt{clan}(Q)$ exist, these clans are disjoint, and that each atom belongs to a clan. Note that P and Q may overlap, so there might be distinct atoms a^P_α and a^Q_α and disjoint litters $\mathtt{litter}^P(a)$ and $\mathtt{litter}^Q(a)$ for fixed a, α with $a \in P \cap Q$.

• P is called the parent set of $\mathtt{clan}(P)$. Details of which sets are parent sets will be revealed below. $a \in P$ is called the parent of a_α as an atom, of $\mathtt{litter}^P(a)$ as a litter, and of any near-litter with small symmetric difference from $\mathtt{litter}^P(a)$.

• The group G of allowable permutations consists of all permutations π of the atoms [extended to all sets by the rule $\pi(A) = \pi``A$]

with the property that π fixes each clan and for each near-litter N (included in any clan) with small symmetric difference from a litter $\texttt{litter}(a)$, $\pi(N)$ has small symmetric difference from $\texttt{litter}(\pi(a))$. An exception for an allowable permutation π is an atom x in a $\texttt{litter}(a)$ such that either $\pi(x) \notin \texttt{litter}(\pi(a))$ or $\pi^{-1}(x) \notin \texttt{litter}(\pi^{-1}(a))$.

• A support set S is a small set of atoms and near-litters (possibly from many clans). A object A has support S iff every allowable permutation which fixes each element of S also fixes A. Clearly an atom has support its own singleton.

General features of the main construction

In the main construction, we build a system of infinitary notation intended to represent the atoms in the clans we postulate and selected sets in selected iterated power sets of those clans. We define a notion of equivalence on these infinitary notations, which on each type of notation coincides with the relation of having the same referent, although the equivalence relation is defined by recursion on the structure of the notation independently of the semantics. At the same time, the referents of notations of various types are stated.

Types of infinitary notation

• Fix a limit ordinal λ and a large transitive pure set X (κ would do).

•A clan index is a finite subset of λ. if A is a nonempty clan index, define A_1 as $A \setminus \{A\}$. Define A_0 as A and A_{n+1} as $(A_n)_1$ where this is defined. We say that B downward extends A when $A \subseteq B$ and all elements of $B \setminus A$ are less than all elements of A. We say $B << A$ when B downward extends A and is distinct from A.

• We will define a system of codes, indicate the nature of their intended referents, and define a relation of equivalence $\sim$ on codes. Codes will have types correlated with the kinds of referents they are intended to have. It will be important to notice that the relation of equivalence we define will turn out to coincide for codes of the same type with the relation of having the same intended referent, but is not in fact defined in terms of the referents of codes (where the relation holds between codes of different types, these codes will have the same referents as well). We use the notation $\delta(c)$ for the referent of a code c of whatever type: this is safe because the various types of codes are disjoint. All codes will be pure sets.

• For each clan index A, there will be a type $\texttt{clan}_*(A)$ (codes of this type are intended to refer to atoms in the clan indexed by A), a type $\texttt{clan}^\circ_*(A)$ (codes of this type are intended to refer to near-litters included in the clan indexed by A), and types $C^n(A)$ for each natural number $n \leq |A|$ (codes of this type are intended to refer to elements of the nth iterated power set of the clan indexed by A: not all elements of these iterated power sets will be referents of codes). Each type as an object is viewed as the class of codes of that type. These classes are in fact sets, but this requires demonstration.

Codes for atoms

A code of type $\texttt{clan}_*(A)$ is of the form $(1, p, A, \alpha)$, where $p \in X$ if A is empty and otherwise p is a code of type $\texttt{clan}_*(A_1)$ or else a code of a type $C^{|B|-|A|+1}(B)$ for some $B << A$. The intended referent of $(1, p, A, \alpha)$ is an atom denoted by $\delta(p)^A_\alpha$ (note that we are using the index A to indicate which clan the atom is in rather than the parent set of this clan which we do not yet know how to describe): the intention is that $\delta(p)^A_\alpha = \delta(p')^{A'}_{\alpha'}$ iff $\delta(p) = \delta(p')$ and $A = A'$ and $\alpha = \alpha'$. The equivalence $(1, p, A, \alpha) \sim (1, p', A, \alpha)$ holds iff $p \sim p'$, $A = A'$ and $\alpha = \alpha'$. For $p \in X$, $\delta(p) = p$ and $p \sim q$ holds iff $p = q$ for $p, q \in X$. The notation p is called the formal parent of the notation $(1, p, A, \alpha)$.

All codes are pure sets. For each type $\texttt{clan}_*(A)$ which is a set and each $\sim$-equivalence class $[c]$

for $c \in \texttt{clan}_*(A)$ we provide an atom $\delta(c)$ correlated with the $\sim$-equivalence class $[c]$ ($\delta(c) = \delta(d)$ iff $c \sim d$). This stipulation is made in this form to ensure that no more than a set of atoms are postulated in any case. It does turn out that all of these types are sets so we get atoms correlated with all equivalence classes of codes in types $\texttt{clan}_*(A)$. All atoms in our ambient ZFA are of this kind.

It is a useful observation that it is immediately evident that there is a large collection of mutually inequivalent codes in each of these types: consider atoms with iterated formal parents in X.

Clans and litters introduced officially

The class $\delta``\texttt{clan}_*(A)$ is called $\texttt{clan}[A]$, and such classes of atoms are called clans. These classes will turn out to be sets, but this requires demonstration. The set $\{p^A_\alpha : \alpha < \kappa\}$ (for appropriate p) is called $\texttt{litter}^A(p)$, and such sets are called litters. A set which is a subset of a clan and has small symmetric difference from a litter is called a near-litter. The class of near-litters included in $\texttt{clan}[A]$ will be denoted by $\texttt{clan}^\circ[A]$. The object p is called the parent of p^A_α as an atom, of $\texttt{litter}^A(p)$ as a litter, and of any near-litter with small symmetric difference from $\texttt{litter}^A(p)$ as a near-litter. The notation $\texttt{clan}[A]$ with brackets is used because the parameter is an index of the clan rather than its parent set, which we do not yet know how to describe.

Codes for near-litters

The set $\texttt{litter}_*^A(p)$ is defined as

$$\{(1, p, A, \alpha) : \alpha < \kappa\},$$

whenever this is a set of codes of type $\texttt{clan}_*(A)$. A code of type $\texttt{clan}_*^\circ(A)$ is a subset of type $\texttt{clan}_*(A)$ with small symmetric difference from some $\texttt{litter}_*^A(p)$ (this p is called the formal parent of the code of type $\texttt{clan}_*^\circ(A)$), and with no distinct but $\sim$-equivalent members. Two codes M, N of this type are $\sim$-equivalent iff each element of M is $\sim$-equivalent to an element of N and vice versa. The intended referent of a code M of type $\texttt{clan}_*^\circ(A)$ is the set of atoms δ“M (easily seen to be a near-litter if all codes involved have referents as expected, and it is also easy to see that each near-litter will have codes of this kind).

Overview of the development of set codes

•General form of codes of types $C^n(A)$: A code of any of these types will be of the form $(2, f, L)$, which we will write $f[L]$ to suggest function application, where f will be a "function code" (to be defined below) and L will be an "argument list" (to be defined below). A code of a type $C^{k+1}(A)$ is called a set code.

General features of argument lists and function codes

• An argument list will be a function with domain a small set of small ordinals and codomain the union of all types $\texttt{clan}_*(A)$ and $\texttt{clan}^\circ_*(A)$ which further belongs to an argument list type (these types to be defined below). If L is an argument list and $\alpha \neq \beta$ belong to the domain of L, then $L(\alpha) \not\sim L(\beta)$, and moreover if $L(\alpha)$ and $L(\beta)$ both belong to the same type $\texttt{clan}^\circ_*(B)$, then no element of $L(\alpha)$ is $\sim$-equivalent to any element of $L(\beta)$. (atomic referents of values at distinct ordinals are distinct; near-litter referents of values at distinct ordinals are disjoint).

• Each function code has an input type which is an argument list type and an output type which is a $C^n(A)$. A code $f[L]$ is well-formed as a code iff L belongs to the input type of f, and the code $f[L]$ will belong to the output type of f.

The definition of argument list types

• Each argument list in T has the same domain D_T (a small subset of κ). We define the relation $L \leq M$ on argument lists as holding when $L \subseteq M$ and all elements of $\mathtt{dom}(M) \setminus \mathtt{dom}(L)$ are greater than all elements of $\mathtt{dom}(L)$. We call this the extension order on argument lists, and say that M extends L to mean $L \leq M$.

• absolute type information: There is a function τ_T from D_T to types such that for each $\beta \in D_T$, and each $L \in T$, $L(\beta)$ belongs to a type determined by $\tau_T(\beta)$: if $\tau_T(\beta) = (0, B)$ then $L(\beta)$ is of type $\mathtt{clan}_*(B)$ and if $\tau_T(\beta) = (1, B)$ then $L(\beta)$ is of type $\mathtt{clan}^\circ_*(B)$; all values of τ_T are of one of these two forms.

• relative type information: There is a function ρ_T with domain D_T which returns additional type information.

If $\tau_T(\beta) = (0, A)$ then $\rho_T(\beta)$ is either an ordinal in $D_T \cap \beta$ such that for any $L \in T$, $L(\beta) \in L(\rho_T(\beta)) \in \mathtt{clan}^\circ_*(A)$ or $\rho_T(\beta) = \lambda$ and it is not the case for any $\gamma \in D_T$ with $L(\gamma)$ of type $\mathtt{clan}^\circ_*(A)$ that $L(\beta)$ is $\sim$-equivalent to any element of $L(\gamma)$.

For the next paragraph, define $g[M]\downarrow$ as the element of the range of M equivalent to $g[M]$ if $g[M]$ is of a type $C^0(A)$ (see the way that type $C^0(A)$ codes are defined below to see that this makes sense), and otherwise as $g[M]$.

If $\tau_T(\beta) = (1, A)$ then either $\rho_T(\beta) = (0, g, M)$ where M is a subset of $D_T \cap \beta$ and g is a function code, and for any $L \in T$, $L(\beta)$ has small symmetric difference from $\mathtt{litter}_*^A(g[L\lceil M]\downarrow)$ ($g[L\lceil M]$ being a well-formed code), or $A = \emptyset$, $\rho_T(\beta) = (1, B), B \in X$ and for any $L \in T$, $L(\beta)$ has small symmetric difference from $\mathtt{litter}_*^A(B)$

• Any L which meets the conditions stated under the three headings above is an element of T.

• $(3, D_T, \tau_T, \rho_T)$ meeting the conditions above is a name for an argument list type with

$$\delta(3, D_T, \tau_T, \rho_T) = T.$$

Other codes for atoms

A code of type $C^0(A)$ will be of the shape $(4, \beta, T^*)[L]$, where $\delta(T^*) = T$ is an argument list type and $\beta \in D_T$, satisfying the further conditions that $L \in T$ and $\tau_T(\beta) = (0, A)$. $(4, \beta, T^*)[L] \sim (4, \beta', T')[L']$ iff $L(\beta) \sim L'(\beta')$, and $\delta((4, \beta, T^*)[L]) = \delta(L(\beta))$. We further provide that $(4, \beta, T^*)[L] \sim c$ holds for c of type $\mathtt{clan}_*(A)$ iff $L(\beta) \sim c$. Codes of this type represent the same referents as codes of type $\mathtt{clan}_*(A)$, but they are not of type $\mathtt{clan}_*(A)$. Of course we have implicitly declared $(4, \beta, T^*)$ as a function code with input type $\delta(T^*) = T$ and output type $\mathtt{clan}_*(A)$; these are the only function codes with this kind of output type.

Codes of type $C^0(A)$ are never proper components of other codes, but their function code components do so occur.

Codes for sets

A code of type $C^{k+1}(A)$ will be of the shape $(5, U, T^*, k, A)[L]$ where $L \in \delta(T^*) = T$, an argument list type, and U is a set of function codes with input types inhabited by argument lists extending elements of T (in the technical sense defined above: for any L in T and M in the input type of an element of U, $L \leq M$) and output type $C^k(A)$, with the further restriction that the additional argument types appearing in the input types of elements of U (over and above those appearing in T) all be either types $\texttt{clan}_*(B)$ with B downward extending A_k (not necessarily properly) or types $\texttt{clan}^\circ_*(B)$ with B downward extending A_{k-1} (not necessarily properly) and $k > 0$.

We have implicitly declared $(5, U, T^*, k, A)$ to be a function code with input type $\delta(T^*)$ and output type $C^{k+1}(T)$. All function codes with such output types are of this form.

The referent $\delta((5, U, T^*, k, A)[L])$ is defined as the class

$$\{\delta(g[M]) : g \in U \wedge L \leq M\},$$

under the conditions that each such $\delta(g[M])$ is defined and that this class is a set.

A formal element of $(5, U, T^*, k, A)[L]$ is a code $g[M]$ with $g \in U$, $L \in \delta(T^*) = T$, an argument list type, and $L \leq M$. The equivalence $(5, U, T^*, k, A)[L] \sim (5, U', (T')^*, k', A')[L']$ holds iff each formal element of each of the two codes is $\sim$-equivalent to some formal element of the other. Note that we do here allow equivalence between codes of different types of the form $C^{k+1}(A)$, in the special case where the referents are hereditarily finite pure sets; this is an annoying technical point with no import for the proof, which can be handled in two or three different ways.

Equivalence is an equivalence relation; infinitary notations make up a set

• $\sim$ is an equivalence relation: This should be evident by induction on the structure of codes.

• Demonstration that function codes make up a set: Recall that a formal parent of an element of $\mathtt{clan}^{\circ}_{*}(B)$ (B nonempty) will be either an element of $\mathtt{clan}_{*}(B_1)$ or an element of a $C^{|D|-|B|+1}(D)$ where $D << B$. Assign to each type $C^n(D)$ a measure of complexity which is the minimum element of D_n (or λ if D_n is empty) and assign each $\mathtt{clan}_{*}(D)$ complexity $\mathtt{min}(D)$. Note that the complexity of the type of each parent of an element of $\mathtt{clan}^{\circ}_{*}(B)$ is the minimum element of B_1. Now observe that the output types of function codes occurring in the argument list types of elements of U but not in T in a function code $(5, U, T^*, k, A)$ will be of complexity at most the minimum element of A_k, and so strictly less than the complexity of the code of type $C_{k+1}(A)$ being constructed, which will be the minimum element of A_{k+1} or possibly λ.

Note that the set theoretical rank of an argument list type name T^* is displaced by no more than a finite constant above the maximum of λ and the maximum of the ranks of function codes embedded in it (and that only a small collection of function codes are embedded in any argument list type name). Notice that the set theoretical rank of a function code $(4, \beta, T^*)$ is displaced by no more than a finite constant above the maximum of λ and the rank of T^*. We claim further that the set theoretical rank of a function code $(5, U, T^*, k, A)$ is displaced upward from the maximum of λ and the rank of T^* by no more than a (non-finite) constant $\nu(\beta)$ depending on the complexity $\beta = \mathtt{min}(A_{k+1}$ (or λ) of its output type

$C^{k+1}(A)$. By inductive hypothesis, each function code in U, with output type of complexity $\beta' = \mathtt{min}(A_k)$, has rank exceeding the rank of its own input type by no more than $\nu(\beta')$. The rank of the input type of an element of U may exceed that of T^*, because it may include additional function codes of output type complexity bounded by β', whose ranks may exceed that of their own argument list types by no more than $\nu(\beta')$, but in any case the rank of this input type will not exceed that of T^* by more than $\nu(\beta') \cdot \kappa$ (there will be no more than a small collection of occasions for increments). So $\nu(\beta)$ may be taken to exceed the upper bound of ordinals $\nu(\beta') \cdot \kappa$ for $\beta' < \beta$ by a suitable finite constant. If every function code has a bound on its rank computable from the complexity of its output type and the ranks of a small collection of function codes of lower rank, it follows that there is a uniform bound on the rank of all function codes, whence function codes make up a set.

Note that the fact that the function codes make up a set ensures that the argument list types make up a set, so that each of the types of atoms and near-litter codes make up sets, so all clans are sets and we only require a set of atoms, correlated with the $\sim$-equivalence classes of elements of types $\texttt{clan}_*[A]$.

Notational conventions for the rest of the paper

• We will use the notation $f_{U,T}(L)$ to abbreviate $\delta((5, U, T^*, k, A)[L])$, and the notation $\pi_\beta(L)$ to abbreviate $\delta((4, \beta, T^*)[L])$ [these are implicit function definitions]. The objects $f_{U,T}$ and π_β we call "coding functions" (these notations are actually polymorphic as not all type information is included in the notation: more complete notation might be $\pi_{\beta,T}$, $f_{U,T,k,A}$, but the additional information is generally in the context).

• Our clans $\mathtt{clan}[A]$ are written with brackets in this section because A is not the parent set of $\mathtt{clan}[A]$. The parent set of $\mathtt{clan}[A]$, which we will denote by $P(A)$ is actually

$$\mathtt{clan}[A_1] \cup \bigcup_{B<<A} \delta\text{``}C^{|B|-|A|+1}(A).$$

We denote $\delta\text{``}C^n(A) \subseteq \mathcal{P}^n(\mathtt{clan}[A])$ by $\mathcal{P}^n_*(\mathtt{clan}[A])$: this is the set of all codable elements of the given iterated power set of the given clan. Note the equation $\texttt{clan}[A] = \texttt{clan}(P(A))$ relating the notation of the previous section for clans with the notation of this section.

• It should be noted how very odd the interlocking structure is of clans and parent sets introduced here. Notice that $P(A)$ includes $\texttt{clan}(P(A_1))$ and so includes a set as large as $P(A_1)$ and so on for each $P(A_n)$. But on the other hand $P(A_n)$ includes $\mathcal{P}_*^{n+1}(\texttt{clan}(P(A))$ (if $n > 0$ and A_n is nonempty) which certainly is at least as large as $P(A)$. We will see below, however, that $\mathcal{P}_*^{n+1}(\texttt{clan}(P(A))$ is the full iterated power set of its clan argument from the standpoint of the FM interpretation we will define.

We begin proving Lemmas about argument list manipulations

In this and the next series of slides we will be proving lemmas supporting basic manipulations of argument lists.

We start with a useful

Definition (application of a permutation to an argument list): If L is an argument list and π is a permutation of atoms which fixes clans and sends each near-litter in the range of L to a near-litter, define L_π as an argument list (we do not care which one) such that for each α in the domain of L, $\delta(L_\pi(\alpha)) = \pi(\delta(L(\alpha)))$.

The permutation lemma

Permutation Lemma: If L belongs to argument list type T and π is an allowable permutation (so L_π is certainly defined), $L_\pi \in T$ as well; further, for each coding function f, $\pi(f(L)) = f(L_\pi)$.

Proof of Permutation Lemma: The two parts are proved by mutual structural induction.

Certainly $L_\pi(\beta)$ will denote an element of the type indicated by $\tau_T(\beta)$ for each β, since an element of a clan or near-litter included in a clan which happens to be in the range of L will be sent to an element of the same clan or a near-litter included in the same clan, because L_π is defined. It remains to check the relative type conditions. For each α, β, it is evident that $L(\beta)$ denotes an atom belonging to an near-litter denoted by $L(\alpha)$ iff $L_\pi(\beta)$ denotes an atom belonging to an near-litter denoted by $L_\pi(\alpha)$, and this is enough for the relative type conditions for atoms to be preserved. Further, if g is a coding function and M is a subset of $D_T \cap \beta$, and $L(\beta)$ denotes a near-litter with parent $g(L\lceil M)$, we can suppose as an inductive hypothesis that $\pi(g(L\lceil M)) = g(((L\lceil M)_\pi) = g(L_\pi\lceil M)$ [the coding function g being simpler than the argument list type T, since it appears as a component of its specification], and this is the parent of $L_\pi(\beta)$, confirming that the relative type conditions for near-litters hold, and the type of L_π is the same as the type of L.

That $\pi(\pi_\beta(L)) = \pi_\beta(L_\pi)$ is obvious, as the coding functions with atomic output are in effect projection functions.

Now consider

$$\pi(f_{A,T}(L)) = \{\pi(g(M)) : g \in A \wedge L \leq M\}$$

(here we apply the inductive hypothesis; g has output type an iterated power set of a clan with smaller index)

$= \{g(M_\pi) : g \in A \wedge L \leq M\} = \{g(M) : g \in A \wedge L_\pi \leq M\}$

This completes the argument.

Corollary on supports: Any coded object $f(L)$ has a support consisting of the referents of range elements of L.

The list structure lemma

List Structure Lemma: If π is a permutation fixing clans and L, M are argument lists such that L_π and M_π are both defined, then for any suitable coding functions f, g, $f(L_\pi) = g(M_\pi)$ iff $f(L) = g(M)$. It is important here that we do not actually assume that π is an allowable permutation, but only that it is locally well-behaved.

Proof of List Structure Lemma: It is clear that this holds in the case where $f(L)$ and $g(M)$ are atoms: $\pi_\beta(L) = \pi_\gamma(M)$ iff $\delta(L(\beta)) = \delta(L(\gamma))$ iff $\pi(\delta(L(\beta))) = \pi(\delta(L(\gamma)))$ iff $\delta(L_\pi(\beta)) = \delta(M_\pi(\gamma))$ iff $\pi_\beta(L_\pi) = \pi_\gamma(M_\pi)$.

In the case where $f(L)$ and $g(M)$ are sets in a given iterated power set of a clan, assume that the result has already been shown for all iterated power sets of that clan with smaller index and all permutations. $f_{A,T}(L) = f_{B,U}(M)$ is equivalent to

$$\{g(L') : g \in A \wedge L \leq L'\} = \{h(M') : h \in B \wedge M \leq M'\}.$$

$f_{A,T}(L_\pi) = f_{B,U}(M_\pi)$ is equivalent to

$$\{g(L') : g \in A \wedge L_\pi \leq L'\} = \{h(M') : h \in B \wedge M_\pi \leq M'\}.$$

We show that if any $g(L')$, where $g \in A$ and $L \leq L'$, is equal to some $h(M')$, with $h \in B$ and $M \subseteq M'$, then $g(L'')$, for any L'' extending L_π, is equivalent to some $h(M'')$ with M'' extending M_π. Observe that L'' can be written $(L')_{\pi'}$, where π' is a permutation agreeing with π at atoms appearing in the range of L or M and at elements of near-litters appearing in the range of L or M, and sending near-litters in the range of $(L' \setminus L)$ or the range of $(M' \setminus M)$ to near-litters. Clearly such a permutation exists. Now define M'' as $(M')_{\pi'}$ and apply the inductive hypothesis. Now prove the same thing with the roles of f, g and L, M interchanged, in the same way, and the equation $\{g(L') : g \in A \wedge L_\pi \subseteq L'\} = \{h(M') : h \in B \wedge M_\pi \subseteq M'\}$ is proved, as required.

The redundancy lemma

Redundancy Lemma: If $f[L]$ is a code of type $C^n(A)$, $f[L] \sim f'[L']$, where $L' \subseteq L$ is determined by choosing a small ordinal α and removing all elements from L whose first projection is $\geq \alpha$ and whose second projection is of a type $\mathtt{clan}_*(B)$ with B not downward extending A_n or (if $n > 0$) of a type $\mathtt{clan}^\circ_*(B)$ with B not downward extending A_{n-1}. The function code f' is obtained from f by removing appropriate elements from all component argument list types (including carrying out the same operation on function codes appearing as components of its argument list type).

Proof of Redundancy Lemma: Observe first that in any case L' defined as above is an argument list. An atom code $L'(\beta)$ with $\beta \geq \alpha$ belonging to an $L(\gamma)$ in $\texttt{clan}^{\circ}_{*}(A_{n-1})$ with $\gamma \geq \alpha$ would have to have its relative type reset in the argument list type T' of L' to indicate that it belonged to no near-litter code in L', since the near-litter to which it actually belongs is omitted.

A near-litter code $L'(\beta)$ with $\beta > \alpha$ which belongs to a $\mathtt{clan}^{\circ}_{*}(B)$ with B downward extending A_{n-1} either has an atom code formal parent which will remain in the domain of L' because it is in $\mathtt{clan}^{\circ}_{*}(B_1)$ and B_1 downward extends A_n, or has formal parent of a type $C^{|C|-|B|+1}$ for C downward extending B expressed as a function $g[L\lceil M]$ of earlier arguments in L which can be converted to $g'[L'\lceil M']$ by a suitable inductive hypothesis, omitting arguments from $L\lceil M$ which are of a type $\mathtt{clan}_{*}(D)$ with D not downward extending $C_{|C|-|B|+1} = B_1$, which includes the case of D not downward extending A_n, or (if $n > 0$) of a type $\mathtt{clan}^{\circ}_{*}(D)$ with D not downward extending $C_{|C|-|B|} = B$, which includes the case of D not downward extending A_{n-1}, without affecting the referent of $g[M]$: in either case the relative type information in T is readily transformed to appropriate information for T'.

If $n = 0$ the main claim is evident: $(4, \beta, T^*)(L) \sim (4, \beta, (T')^*)(L')$ where L' contains all and only the elements of L with second component in the type $\texttt{clan}_*(B)$ to which $L(\beta)$ belongs and T' is the argument list type of L'. The function codes differ precisely in having different component argument list types in the expected way.

Assume that the result holds for $n \leq k$. $(5, U, T^*, k, A)$ denotes $\{\delta(g[M]) : g \in U \wedge L \leq M\}$. Note that the restriction on argument types for function codes in U ensures that $M \setminus L = M' \setminus L$: the new arguments in argument lists M here satisfy the range restriction we impose on M' already. By inductive hypothesis, we know that $\delta(g[M]) = \delta(g'(M'))$ for each specific g and M. Any $\delta(g[M])$ is equivalent to $\delta(g'[M'])$ by inductive hypothesis, and certainly $L' \leq M'$. What does still need to be shown is that any $\delta(g'[M])$ with $L' \leq M$ is in fact a $\delta(g'[M'])$. But this follows from the restriction on types of arguments in $M \setminus L$: no elements of $M \setminus L$ will be removed. So $\{\delta(g[M]) : g \in U \wedge L \leq M\} = \{\delta(g'[M']) : g \in U \wedge L \leq M\}$ (by ind hyp elementwise) $= \{\delta(g'[M]) : g \in U \wedge L' \leq M\} = (5, U', (T')^*, k, A)[L']$, where U' is the set of all g' for $g \in U$.

The Substitution Property

Theorem (Substitution Property): For any permutation π^* of a small set of atoms, sending each atom in its domain to an atom in the same clan, there is an allowable permutation extending π^*, all of whose exceptions are in the domain of π^*.

Proof: For each pair of elements a, b in the same parent set P, choose a map $f_{a,b,P}$ intended to serve as the restriction of π to $\texttt{litter}(a)$ if we find subsequently that $\pi(a) = b$, with the property that it is a bijection from the set of elements of $\texttt{litter}(a)$ not in the field of π^* to the set of elements of $\texttt{litter}(b)$ not in the field of π^*. The value of π at a set A is computable as soon as its value at each element of a support of A obtained as the range of an argument list is computable (this follows from the Permutation Lemma). It is then possible to compute the value of π at every atom and parent, and so at all sets, by a recursion on the well-founded structure of the infinitary notation. Given a notation for an atom, we may assume that we already know how to compute the extended map π at its parent by well-foundedness of the notation: in the case of pure sets, we know that π fixes them; in the case of elements of iterated power sets of clans, we can determine the image of $f(L)$ as $f(L_\pi)$, which we can already compute because all values of π that are needed to compute L_π have already been determined (again by structural induction on the notation) and we can be sure that this does not depend on the particular code $f[L]$ for the set which we use by the List Structure Lemma; in the case of an atomic parent we can appeal directly to induction on the structure of the notation. Now we can compute the value of π at the given atom, either by applying π^* if it is in the domain of π^*, or by applying the appropriate $f_{a,b,P}$, which we have determined because we know the image under π of the parent of the atom.

That all exceptions of the map constructed are in the range of π^* is evident from the construction.

All symmetric sets (in appropriate iterated power sets of clans) are codable

Theorem: All elements of iterated power sets $\mathcal{P}^n(\texttt{clan}[A])$ of clans which have support are actually referents of infinitary notations, and can in fact be expressed as referents of infinitary notations $g[M]$ with argument list extending any fixed argument list L, with each element of the range of $M \setminus L$ belonging to a type $\texttt{clan}_*(B)$ with B downward extending A_n (not necessarily properly) or (if $n > 0$) to a type $\texttt{clan}_*^\circ(B)$ with B downward extending A_{n-1}.

Proof: Fix an argument list L.

This is clearly true for elements of clans, for which all coding functions are in effect projection functions. In this case $n = 0$. One can create an argument list extending L with a single additional argument denoting an atom in $\texttt{clan}[A]$ unless a notation for the atom which is the value is actually in the range of L, and the single additional argument if present is in $\texttt{clan}_*(A)$ and A downward extends $A_n = A_0 = A$.

Suppose the result to be true for all kth power sets of clans. We choose any element E of the $k+1$-th power set of a clan which has a support S. Choose a code $g[M]$ for each element of E, with the argument list M extending an argument list L' which extends the fixed argument list L and whose range includes notations for each element of the given support S of E (subject to a restriction discussed in the last paragraph) and having the property that each element of the range of $M \setminus L'$ belonging to a type $\texttt{clan}_*[B]$ with B downward extending A_k (not necessarily properly) or (if $n > 0$) to a type $\texttt{clan}^\circ_*[B]$ with B downward extending A_{k-1}. We can do this by inductive hypothesis.

The set $f_{A,T}(L')$, where A is the set of codes g used in codes for elements of E and T is the type of L', certainly contains every element of the original set E. Now any element of $f_{A,T}(L')$ is of the form $g(M')$ where there is $g(M) \in E$, and we can define a small map π_0 implementing a substitution, fixing the referent of each element of the range of L', which "sends M to M'" [map atoms with referents in M to atoms with referents in corresponding positions in M'; further extensions to the small map are needed to handle anomalous elements for near-litters referenced in corresponding positions in M and M'.] This substitution map can be extended to a small bijection respecting clans (with additional elements of the domain of the bijection belonging to near-litters referenced in the range of M mapped into near-litters referenced in corresponding positions in the range of M' and additional elements belonging to near-litters referenced in the range of M' having preimages in the near-litters referenced in corresponding positions in the range of M), and then to an allowable permutation fixing each referent of an element of the range of L', mapping $g(M) \in E$ to $g(M')$ (having no exceptions other than elements of the domain of the small bijection causes it to treat near-litters referenced in the range of M correctly). But then $g(M') \in E$ as well, since the allowable permutation fixes all referents of elements of the range of L', a support of E, by construction, and it follows that $f_{A,T}(L') = E$.

To enforce the restriction on types of arguments in the range of L' we apply the Redundancy Lemma with the ordinal α in the proof of the Lemma being set to the first ordinal dominating the domain of L.

Definition of our FM interpretation

The group G defining our FM permutation is the group of allowable permutations. For any support set S, G_S is defined as the set of allowable permutations which fix each element of S. The filter Γ is defined as the set of subgroups of G which include a G_S.

That Γ is a normal filter is straightforward to establish. The normality condition is the only one which requires any work: if H contains G_S, it is straightforward to show that for any $\pi \in G$, $\pi H \pi^{-1}$ includes $G_{\pi(S)}$.

•Observation and convention on iterated power set notation: We have shown that the subset of $\mathcal{P}^{n+1}(\mathtt{clan}(P))$ which is included in a parent set, if it has non-pure members, is the full iterated power set of the clan in the sense of the FM interpretation. Subsequently in this paper, we will use the notation $\mathcal{P}^{n+1}(\texttt{clan}(P))$ to denote the iterated power set in the sense of the FM interpretation, unless we specifically say otherwise. We do have the notation $\mathcal{P}_*^{n+1}(\texttt{clan}(P))$ available for this set by the previous theorem if we need to draw a distinction.

•Observation about internal vs. external cardinalities: We argued above that if A and A_n are both nonempty, then $P(A)$ and $P(A_n)$ (and the associated clans) are of the same cardinality in the ground interpretation. But in the FM interpretation $P(A_n)$ includes

$$\mathcal{P}^{n+1}(\texttt{clan}(P(A))),$$

and so is a much larger set in terms of the FM interpretation. The moral here is that the power sets of the FM interpretation are quite impoverished.

Combinatorics of clans

We now discuss the combinatorics of a fixed $\texttt{clan}(P)$ in the FM model.

• **The domain of the FM interpretation contains its small subsets:** We first observe that every small set of elements of the FM model in an iterated power set of the clan is an element of the FM model. Take the union of supports for each element of the small set to get a support for the small set.

•**Supports may be taken to contain only litters proper:** We note that any support S can be refined to one in which all near-litter elements of S are litters (replace near-litter elements with the litters with small symmetric difference from them; add the elements of the symmetric differences (the anomalous elements for the near-litters) to the support). We do this everywhere below; allowing near-litters in supports is important because it simplifies the proof of normality, but here we prefer to eliminate them.

•**Strong support: definition and lemma** A strong support for a set A is a support obtained from a code $f[L]$ for A by adding to the set of referents of elements of the range of L all anomalous elements for near-litter referents of elements of the range of L (which allows us to replace near-litters with litters with the same parent). An allowable permutation π which fixes all atomic elements of a strong support of A and has no exceptions in litter elements of S other than possibly fixed points of π will fix A: if it moves A, it moves some element of the strong support, which cannot be an atom, so it must move a litter, and it must move a first one in the order on the original argument list, whose parent it must fix as all elements of a support thereof (references to which appear earlier in the list) are fixed, and it can only move a litter whose parent it fixes (and all of whose anomalous elements it fixes) by having an exception in the litter, which will not be a fixed point of π, again because the parent is fixed.

•**Litters are sets of the FM interpretation:** Each litter is a set of the FM interpretation with support consisting of its own singleton.

•**Definition (κ-amorphous set):** We call a set κ-amorphous iff all its subsets are small or co-small.

•**Litters are κ-amorphous in the FM interpretation:** We show that the litters in $\texttt{clan}(P)$ are κ-amorphous sets in the FM interpretation. Suppose to the contrary that L is a litter, $A \subseteq L$ is large and $L \setminus A$ is large, and A has strong support S. Let $a \in A$ and $b \in L \setminus A$ with neither a nor b belonging to S nor to a fixed strong support of L. There will be an allowable permutation π extending the small map which interchanges a and b and fixes each atomic element of a strong support of L and each atomic element of S and has no exceptions belonging to any litter in the strong support of L or in S (because in fact it has no exceptions which it moves), and so fixes all elements of S. But this is impossible, because such a map would move A while fixing every element of its given support.

•**Theorem:** The subsets of each $\texttt{clan}(P)$ in the FM interpretation are exactly the sets with small symmetric difference from the unions of small or co-small collections of the litters in the clan.

•**Proof:**

We say that a set A cuts a set B iff $B \cap A$ and $B \setminus A$ are both nonempty.

We claim that for any subset A of $\texttt{clan}(P)$ in the FM interpretation, A can cut only a small collection of litters. Suppose otherwise, that A cuts a large collection of litters and has a strong support S. Choose a and b in the same litter L, one belonging to A and the other not belonging to A, with neither belonging to S. There will be a permutation π extending the small map exchanging a and b and fixing each atomic element of a strong support of L and each atomic element of S, and having no exceptions other than elements of S and of the strong support of L; but this is impossible, as this map must also fix A, as it fixes all atomic elements of its given support and has no exceptions which it moves belonging to litters in the given support, since the map has no exceptions which it moves.

We claim that for any subset A of $\mathtt{clan}(P)$ in the FM interpretation, it cannot be the case that a large number of litters meet A and a large number of litters do not meet A. Suppose that A has strong support S and a large number of litters meet A and a large number of litters do not meet A. Choose atoms a and b, one belonging to a litter L meeting A and one belonging to a litter M not meeting A, chosen so that none of a, b, L, M belong to S. There will be a permutation extending the small map extending a and b and fixing all atomic elements of S and of strong supports of L, M and having no exceptions which it moves belonging to any litter element of S (since a, b are the only exceptions which it moves) and so fixing all litter elements of S. This permutation moves A but it cannot do so because it fixes all elements of its given support.

Now there are two kinds of subset of $\texttt{clan}(P)$ in the FM interpretation:

1. sets which meet a small collection of litters in P and so have small symmetric difference from the union of the litters in this small collection whose intersection with A is co-small,

2. and sets which meet a large collection of litters in P (and so fail to meet only the litters in a small collection), but cut only a

small collection of them, which thus have small symmetric difference from the union of the co-small collection of litters which have co-small intersection with A.

So every such set A has small symmetric difference from a small or co-small union of litters. Further, it is evident that any small union of litters is a set of the FM interpretation, so any small or co-small union of litters is a set of the FM interpretation, so any set with symmetric difference from a small or co-small union of litters is a set of the FM interpretation.

•**Observation:** Note that this tells us that $\texttt{clan}(P)$ has the same power set in the FM interpretation quite independently of what power set P has in the FM interpretation; sets $\texttt{clan}(P)$ with parent sets of the same cardinality in terms of the ground interpretation will have power sets in the FM interpretation which are isomorphic in terms of the ground interpretation. This will not be true for further iterated power sets in the FM interpretation.

•**Theorem (double power set lemma):** The set $\mathcal{P}^2(\texttt{cal}(P))$ contains a set the same size as $\mathcal{P}(P)$ acccording to the FM interpretation.

•**Proof:** We argue that a subset of $\texttt{clan}(P)$ is the same size as a litter $L \subseteq \texttt{clan}(P)$ in the FM interpretation iff it has small symmetric difference from L. First, it is clear that a set which has small symmetric difference from L is the same size as L in the FM interpretation, as a bijection witnessing this fact can be obtained which has small symmetric difference from the identity map and so certainly is a set of the FM interpretation. Now suppose that there is a bijection f from L to a set A where $L \Delta A$ is large, with strong support S. Let U be one of $L \setminus A$ and $A \setminus L$ which happens to be large. Let g be the one of f and f^{-1} which is defined on U. We choose two elements a, b from U in such a way that none of $a, b, g(a), g(b)$ belong to S; we choose these so that the elements of each of the pairs a, b and $g(a), g(b)$ each belong to the same litter (one of the pairs both belong to L; some litter must have a large intersection with the large set $A \setminus L$). There is a permutation π which swaps a, b and fixes $g(a)$ and $g(b)$, and further fixes each atomic element of S and has no exceptions which it moves in near-litter elements of S (since it has no exceptions which it moves) so fixes all elements of S. The resulting map will move f, but this is impossible because it fixes all elements of the given support of f.

Thus a reasonable nonce definition for $|\mathtt{litter}(a)|$ is as the collection of near-litters with small symmetric difference from the litter, as this is exactly the collection of subsets of the same clan with this cardinality. Now the map $(a \in P \mapsto |\mathtt{litter}(a)|)$ has empty support in the allowable permutations, so is a set of the FM interpretation. Moreover

$$(B \subseteq P \mapsto \bigcup_{a \in B} |\mathtt{litter}(a)|)$$

is a set of the FM interpretation for the same reason, and is a bijection from $\mathcal{P}(P)$ into $\mathcal{P}^2(\texttt{clan}(P))$ in the sense of the FM interpretation. So the abundance of subsets of P in the FM interpretation has no effect on the extent of $\mathcal{P}(\texttt{clan}(P))$ in the FM interpretation, but has a strong effect on the extent of $\mathcal{P}^2(\texttt{clan}(P))$.

•This is a key idea of the proof: the ability to construct sets which are externally isomorphic (in the sense of the ground interpretation) and have quite different power sets (in the sense of the FM interpretation) is essential for getting an argument for Con(NF) analogous to Jensen's argument for Con(NFU) to work. Further applications of this machinery allow us to do the same thing with models of initial segments of simple type theory with arbitrarily many types, getting externally isomorphic natural models of TST_n in the FM interpretation whose top types have non-isomorphic power sets in the FM interpretation, so the natural models of TST_{n+1} extending them are not isomorphic.

Parent clans; sizes of iterated power sets of clans

Note that $\mathcal{P}(D)$ in this section denotes the power set of D in the FM interpretation.

Where P is a parent set $P(A)$, we use the notation P_1 to represent $P(A_1)$.

We know from above that $\mathcal{P}^2(\texttt{clan}(P))$ contains a subset the same size as $\mathcal{P}(P)$ in the sense of the FM interpretation. This means that it further contains a set the same size as $\mathcal{P}(\texttt{clan}(P_1))$. Thus $\mathcal{P}^3(\texttt{clan}(P))$ contains a set the same size as $\mathcal{P}^2(\texttt{clan}(P_1)$ which contains a set the same size as $\mathcal{P}(P_1)$. Now we have an argument by induction. Suppose that we have shown that $\mathcal{P}^{n+2}(\texttt{clan}(P))$ contains a set the same size as $\mathcal{P}(P_n)$. It follows that $\mathcal{P}^{n+3}(\texttt{clan}(P))$ contains a set the same size as $\mathcal{P}^2(P_n)$ which contains a set the same size as

$\mathcal{P}^2(\texttt{clan}(P_{n+1}))$ which contains a set the same size as $\mathcal{P}(P_{n+1})$. This completes a proof by induction of the following

Theorem: $\mathcal{P}^{n+2}(\texttt{clan}(P))$ contains a set the same size as $\mathcal{P}(P_n)$, in the sense of the FM interpretation, for every n for which P_n is defined.

Convergent cardinalities of iterated power sets

We show that if $A_i = B_j$,

$$|\mathcal{P}^{i+2}(\texttt{clan}[A])| = |\mathcal{P}^{j+2}(\texttt{clan}[B])|$$

in the FM interpretation. We know from results above that $\mathcal{P}^{i+2}(\texttt{clan}[A])$ contains a set the same size as $\mathcal{P}(P(A_i))$ and $\mathcal{P}^{j+2}(\texttt{clan}[B])$ contains a set the same size as $\mathcal{P}(P(B_j))$ (cardinalities here being understood in the sense of the FM interpretation). But $A << A_i$ so $P(A_i) = P(B_j)$ contains $\mathcal{P}^{|A|-|A_i|+1}(\texttt{clan}[A]) = \mathcal{P}^{i+1}(\texttt{clan}[A])$ and similarly contains $\mathcal{P}^{j+1}(\texttt{clan}[B])$, from which it follows that $\mathcal{P}^{i+2}(\texttt{clan}[A])$ contains a set the same size as $\mathcal{P}(P(A_i))$ which contains a set the same size as $\mathcal{P}(\mathcal{P}^{j+1}(\texttt{clan}[B]))$ which is $\mathcal{P}^{j+2}(\texttt{clan}[B])$, and vice versa, so

$$|\mathcal{P}^{i+2}(\texttt{clan}[A])| = |\mathcal{P}^{j+2}(\texttt{clan}[B])|$$

in the FM interpretation by Schröder-Bernstein.

Isomorphism of iterated power sets

Every item in the iterated power set $\mathcal{P}^n(\texttt{clan}[A])$ has a code with argument list containing no code for an atom in a $\texttt{clan}[B]$ with B not downward extending A_n nor any code for a near-litter in a $\texttt{clan}^\circ[B]$ with B not downward extending A_{n-1}, by the Redundancy Lemma. Call such codes "formally restricted". Now if $B \setminus B_n = A \setminus A_n$ then choose any external bijection from $\texttt{clan}[A_n]$ to $\texttt{clan}[B_n]$: this bijection can be naturally extended (its action on parents dictating its action on (formal representations of) atoms, its action on (formal representations of) elements dictating its action on (formal representations of) sets) to a map converting any formally restricted code for an element of $\mathcal{P}^n(\texttt{clan}[A])$ to a formally restricted code for an element of $\mathcal{P}^n(\texttt{clan}[B])$ bijectively (with adjustments of type indices by replacing each type index C appearing with $(C \setminus A_n) \cup B_n$: this exactly preserves structure). In the case where A_n or B_n is empty, we need the assumption that $P(\emptyset) = X$ is as large as the other $P(A)$'s [which can indeed be arranged, but we choose not to discuss this], but this is not strictly needed as the case where both are nonempty is sufficient for the proof (as we will see below). This bijection preserves structure, because facts of membership and equality in the power sets are computable from the infinitary notation considered abstractly, and the features of the infinitary notation which support this computation are preserved by the transformation in question. Note that this transformation acts on all lower indexed iterated power sets of $\texttt{clan}[A]$ and $\texttt{clan}[B]$ as well, but there is no reason to expect it even to send sets to sets on $\mathcal{P}^{n+1}(\texttt{clan}[A])$: this transformation is not a function of the FM interpretation.

This shows us that the first order theory of the natural model of the first $n+1$ types in the FM interpretation whose base type is $\texttt{clan}[A]$ and whose top type $\mathcal{P}^n(\texttt{clan}[A])$ depends only on $A \setminus A_n$, at least as long as A_n is not empty (and in fact not in this case, either, but we are not concerned to show this).

Consistency of NF

Suppose that $\lambda > \omega$.

The final step is to observe that for any fixed limit ordinal $\alpha < \lambda$, $\tau(A) = |\mathcal{P}^2(\texttt{clan}(A \cup \{\alpha\}))|$ for nonempty A dominated by α defines a tangled web of cardinals in the FM interpretation.

The cardinality of the power set of

$$\mathcal{P}^2(\texttt{clan}[A \cup \{\alpha\}])$$

is the same as that of $\mathcal{P}^2(\texttt{clan}(A_1 \cup \{\alpha\}))$ by results above (if A has at least two elements). Of course $A_1 \cup \{\alpha\} = (A \cup \{\alpha\})_1$. This establishes $2^{\tau(A)} = \tau(A_1)$ for A with at least two elements, which verifies the naturality property of tangled webs for this τ.

Consider natural models of initial segments of simple type theory with base type $\mathcal{P}^2(\texttt{clan}[A \cup$

$\{\alpha\}])$. The theory of such a model is determined by the cardinality of its base type. The theory of the first n types of this model, that is the theory of the model with top type $\mathcal{P}^{n+1}(\texttt{clan}[A \cup \{\alpha\}])$, is completely determined by the first $n+1$ elements of A by results above. And this establishes the elementarity property of a tangled web for this τ.

We have already shown that the existence of a tangled web implies the consistency of NF.

The assumption that $\lambda > \omega$ and the use of α in the definition of tangled webs is purely technical; we have avoided proving that the theory of the natural model of TST_n with bottom type $\texttt{clan}(A)$ and top type $\mathcal{P}^n(\texttt{clan}[A])$ depends only on $A \backslash A_n$ in the case where A_n is empty, though this does in fact hold, so the tangled web could be defined as $\tau(A) = |\mathcal{P}^2(\texttt{clan}(A))|$.

Conclusions to be drawn about NF

The conclusions to be drawn about NF are rather unexciting ones.

By choosing the parameter λ to be larger (and so to have stronger partition properties) one can show the consistency of a hierarchy of extensions of NF similar to extensions of NFU known to be consistent: one can replicate Jensen's construction of ω- and α-models of NFU to get ω- and α-models of NF. One can show the consistency of NF + Rosser's Axiom of Counting (see [**?**]), Henson's Axiom of Cantorian Sets (see [**?**]), or the author's axioms of Small and Large Ordinals (see [**?**], [**?**], [**?**]) in basically the same way as in NFU.

It seems clear that this argument, suitably refined, shows that the consistency strength of

NF is exactly the minimum possible on previous information, that of TST + Infinity, or Mac Lane set theory (Zermelo set theory with comprehension restricted to bounded formulas). Actually showing that the consistency strength is the very lowest possible might be technically tricky, of course. I have not been concerned to do this here. It is clear from what is done here that NF is much weaker than ZFC.

By choosing the parameter κ to be large enough, one can get local versions of Choice for sets as large as desired, using the fact that any small subset of a type of the structure is symmetric. The minimum value ω_1 for κ already enforces Denumerable Choice (Rosser's assumption in his book) or Dependent Choices. It is unclear whether one can get a linear order on the universe or the Prime Ideal Theorem: that would require major changes in this construction. But certainly the question of whether NF has interesting consequences for familiar mathematical structures such as the continuum is answered in the negative: set κ large enough and what our model of NF will say about such a structure will be entirely in accordance with what our original model of ZFC said. It is worth noting that the models of NF that we obtain are not κ-complete in the sense of containing every subset of their domains of size κ; it is well-known that a model of NF cannot contain all *countable* subsets of its domain. But the models of TST from which its theory is constructed will be κ-complete, so combinatorial consequences of κ-completeness will hold in the model of NF (which could further be made a κ-model by making λ large enough).

The consistency of NF with the existence of a linear order on the universe or the Prime Ideal theorem is not established: questions about many weak versions of Choice remain.

The question of Maurice Boffa as to whether there is an ω-model of TNT (the theory of negative types, that is TST with all integers as types, proposed by Hao Wang ([**?**])) is settled: an ω-model of NF yields an ω-model of TNT instantly. This work does not answer the question, very interesting to the author, of whether there is a model of TNT in which every set is symmetric under permutations of some lower type.

The question of the possibility of cardinals of infinite Specker rank (in ZFA at least) is answered, and we see that the existence of such cardinals doesn't require much consistency strength. For those not familiar with this question, the Specker tree of a cardinal is the tree with that cardinal at the top and the children of each node (a cardinal) being its preimages under $\alpha \mapsto 2^\alpha$. It is a theorem of Forster (a corollary of a well known theorem of Sierpinski) that the Specker tree of a cardinal is well-founded (see [**?**], p. 48), so has an ordinal rank, which we call the Specker rank of the cardinal. NF + Rosser's Axiom of Counting proves that the Specker rank of the cardinality of the universe is infinite; it was unknown until this point whether the existence of a cardinal of infinite Specker rank was consistent with any set theory in which we had confidence. The possibility of a cardinal of infinite Specker rank in ZFA is established by the construction here; we are confident that standard methods of transfer of results obtained from FM constructions in ZFA to ZF will apply to show that cardinals of infinite Specker rank are possible in ZF.

This work does not answer the question as to whether NF proves the existence of infinitely many infinite cardinals (discussed in [**?**], p. 52). A model with only finitely many infinite cardinals would have to be constructed in a totally different way. We conjecture on the basis of our work here that NF probably does prove the existence of infinitely many infinite cardinals, though without knowing what a proof will look like.

A natural general question which arises is, to what extent are *all* models of NF like the ones indirectly shown to exist here? Do any of the features of this construction reflect facts about the universe of NF which we have not yet proved as theorems, or are there quite different models of NF as well?