NF consistency proof via construction of parent functions

M. Randall Holmes

5/22/2016 8 AM Boise time corrections and improvements after first Cambridge talk

Contents

1 Introduction 7
2 The simple theory of types 8
3 The definition of New Foundations 11
4 Well-known results about New Foundations 13
5 Consistency of NFU 15
6 Tangled type theories 18
   6.1 $\omega$- and $\alpha$-models from tangled type theory 20
7 Tangled webs of cardinals 22
8 Preliminaries and cardinal parameters 25
9 Definition of abstract parent functions 26
10 Results about abstract parent functions 31
11 Definition of and construction of typed parent functions 42
12 Definition of the tangled web and verification of its properties 47
13 Conclusions and questions 48
14 References and Index 50
Remarks about this version

5/22/2016: more cleanup of the arguments re combinatorics of the power sets of clans in section 10. It’s much simpler than I was making it. Other edits motivated by the experience of talking through this in Cambridge yesterday. I also hope that I clarified the way that the extension property must be used in the canonical construction of witnesses to the extension property that I give in section 10 (making clear that this is not a proof of the extension property without the massive detour through “coding” found in other versions; but it does support the induction step in the proof of the extension property for typed parent functions later). 8 AM added the silly fix for hereditarily finite elements of parent sets. There is of course also the option of restricting to nonempty sets in the iterated power sets included in parent sets. Second Cambridge talk repeated, and I probably have further revisions to make on reflection about how that went.

5/21/2016 after first Cambridge talk: First of all, I detected that I cannot prove the extension property conveniently, because this would require the full panoply of coding of elements of parent sets which I want to avoid. Nonetheless, the procedure for constructing canonical extensions using a global support order is useful and is preserved (it is needed for the proof that the extension property is preserved by extensions of typed parent functions in the construction). The proof that distinct litters have distinct cardinals was omitted: it is now present. Orbit signatures had a bad component. I used too many supports in various places in the argument for the structure of subsets of clans.

5/21/2016: 2 AM further surgery due to provability of extension property from existence of extended strong supports. This simplifies matters notably at the end of the argument (there is a substantial block of text which is marked as omittable in favor of a brief discussion of why extended strong supports exist). 4:30 AM removed blocks of text no longer needed because the extension property is a theorem. Shortened the definition of abstract parent function by moving extensive proofs related to the extension property into the next section. Added an argument that the stronger assertion about existence of extended strong supports is not depending on the choice of which near-litters are litters: this is purely esthetic, as if this were not the case we would simply
specify the particular choice set from the near-litters as a component of the abstract parent function. But it appears that this is not necessary.

5/20/2016: 5 AM minor fixes 10 AM this version supports (currently as an alternative) a stronger statement of what sorts of strong supports exist from which the extension property can be proved. It also contains a better (hopefully at least correct) statement of the proof that supports can be extended to strong supports. The proof of the extension property should be familiar in its general approach from older versions: it requires a stronger hypothesis concerning existence of strong supports.

5/19/2016: The extension property has been restated in a perhaps somewhat more familiar way (and notions of anomaly and exception have been reintroduced in connection with it). The form working on nice maps has been proved from a form working on locally small bijections on atoms. Harmony with later sections will be reestablished as I work today. 7:15 am more updates and edits. 8:30 AM the proof of the extension property in the inductive construction of typed parent functions is now verifying the new version of the extension property. 9 am further tweaks. 9:30 am: index updated. 3 PM: made important point about why the construction of extensions in the induction step of the construction of typed parent functions works. 4 PM technical point in proving that all supports can be extended to strong supports.

5/16/2016: More cleanup. Some of this is just proofreading. At 3 pm, fixed a substantial oversight: the extension property was not stated with the full generality needed for the argument (though it was used with this full generality later in the argument, which confused me when I read it in this pass).

5/13/2016: Eliminated the notion of indexed support set in favor of working directly with strong supports with the order component a well-ordering rather than a general irreflexive well-founded relation. This is just a reduction of notational burden. Incorporated the condition that clan indices are extended type definitions into the definition of abstract parent function, removing fogginess from the definition of complexity. Section 10 should be thoroughly revised along these lines; the next release will have a rewrite of Section 11. 1:30 PM more edits. 430 pm more edits
5/8/2016: Careful (I hope!) editing of notation for strong supports throughout. Using more precise notation and stating things in a (hopefully) helpfully finicky manner. The work of this time interval was a fairly thorough editing pass through sections 9 and 10. The main changes were as noted more careful choices of notation for strong supports and their components and more careful use of these notations in arguments.

5/6/2016: Debugging definition of abstract parent function. There were various places where more clarity was needed and some definite slips. Further debugging of definition of strong support set following.

4/25/2016: 12:30 am: This is a “different approach”. It is actually rather like the very first attempts I made to write down this proof. Readers of other versions might find it useful to see what the motivation is of things which are more mysterious in other versions.

Needless to say, much of this is new material and may have annoying bugs in it. New notation and bookkeeping always needs to be reviewed.

I also think that the verification of all properties of abstract parent functions at each step of the construction of typed parent functions needs attention (the abstract parent set definition being quite large).

The definition of parent set has reverted to a perhaps apparently simpler form: the parent set indexed by the empty set is a large collection of atoms not belonging to any clan (rather than the complex set it is in the current posted version). A clan does not as in recent versions have its own power set in its parent set. Both of these are technical issues: the more complex structure of the empty-set indexed parent set was motivated by the desire to make this set large, which is stipulated directly here. In the coding based versions, the inclusion of the power set of a clan in its parent set is irritating but harmless, and makes the definition more uniform. Here the original approach seems better.

It has the merit that one is working with atoms, clans and litters directly from the outset. One does not have to construct codes for atoms and their parent sets before constructing them.

The elementarity property in this version is trivial, and it would be fairly easy to turn this model directly into a model of tangled type theory, I believe, though I do not need to.
It appears that one can state the elementarity property in the natural way, without the unexpected appeal to the smallest $n + 1$ elements of a type index where $n$ is expected, but this might yet prove to be a fencepost error.

The very large definition of “abstract parent function” at the beginning (section 9) is in the nature of a very complex inductive hypothesis used in the construction of “typed parent functions” which follows. Many of its consequences can be deduced before we have to go into the nitty-gritty of the details of the parent functions we actually use (section 10). In section 11, we construct the actual parent function we use. In section 12 we describe the tangled web and verify its properties.

The index needs to be extended to the construction. In general, this is an early draft, and surely needs more work.
1 Introduction

In this paper we will present a proof of the consistency of Quine’s set theory “New Foundations” (hereinafter NF), so-called after the title of the 1937 paper [11] in which it was introduced. We will present two different constructions of models of theories which we will show to be equiconsistent with NF.
2 The simple theory of types

New Foundations is introduced as a modification of a simple typed theory of sets which we will call “the simple theory of types” and abbreviate TST (following the usage of Thomas Forster and others).

Definition (the theory TST): TST is a first-order many-sorted theory with equality and membership as primitive relations and with sorts (referred to traditionally as “types”) indexed by the natural numbers. A variable may be written in the form $x^i$ to indicate that it has type $i$ but this is not required; in any event each variable $x$ has a natural number type $\text{type}(x)$. In each atomic formula $x = y$, the types of $x$ and $y$ will be equal; in each atomic formula $x \in y$ the type of $y$ will be the successor of the type of $x$.

The axioms of TST are axioms of extensionality

$$(\forall xy.x = y \leftrightarrow (\forall z.z \in x \leftrightarrow z \in y))$$

for any variables $x, y, z$ of appropriate types and axioms of comprehension, the universal closures of all formulas of the form

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

for any variables $x, A$ of appropriate types and formula $\phi$ in which the variable $A$ does not occur.

Definition (set abstract notation): We define $\{x : \phi\}$ as the witness (unique by extensionality) to the truth of the comprehension axiom

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

For purposes of syntax, the type of $\{x : \phi\}$ is the successor of the type of $x$ (we allow $\{x : \phi\}$ to appear in contexts (other than binders) where variables of the same type may appear).

This completes the definition of TST. The resemblance to naive set theory is not an accident. This theory results by simplification of the type theory of the famous [20] of Russell and Whitehead in two steps. The predicativist scruples of [20] must first be abandoned, following Ramsey’s [12]. Then it needs to be observed that the ordered pair can be defined as a set, a fact
not known to Whitehead and Russell, first revealed by Wiener in 1914 ([21]).
Because Whitehead and Russell did not have a definition of the ordered pair as a set, the system of [20] has a far more complicated type system inhabited by arbitrarily heterogeneous types of $n$-ary relations. The explicit presentation of this simple theory only happens rather late (about 1930): Wang gives a nice discussion of the history in [19].

The semantics of TST are straightforward (at least, the natural semantics are). Type 0 may be thought of as a collection of individuals. Type 1 is inhabited by sets of individuals, type 2 by sets of sets of individuals, and in general type $n + 1$ is inhabited by sets of type $n$ objects. We do not call the type 0 individuals “atoms”: an atom is an object with no elements, and we do not discuss what elements individuals may or may not have.

**Definition (natural model of TST):** A natural model of TST is determined by a sequence of sets $X_i$ indexed by natural numbers $i$ and bijective maps $f_i: X_{i+1} \to \mathcal{P}(X_i)$. Notice that the $f_i$'s witness the fact that $|X_i| = |\mathcal{P}^i(X_0)|$ for each natural number $i$. The interpretation of a sentence in the language of TST in a natural model is obtained by replacing each variable of type $i$ with a variable restricted to $X_i$ (bounding quantifiers binding variables of type $i$ appropriately), leaving atomic formulas $x^i = y^i$ unmodified and changing $x^i \in y^{i+1}$ to $x^i \in f_i(y^{i+1})$.

**Observations about natural models:** It is straightforward to establish that

1. The axioms translate to true sentences in the natural model.
2. The first-order theory of the natural model is completely determined by the cardinality of $X_0$. It is straightforward to construct an isomorphism between natural models with base types of the same size.

It is usual to adjoin axioms of Infinity and Choice to TST. We do not do this here, and the precise form of such axioms does not concern us at the moment.

The theory $\text{TST}_n$ is defined in the same way as TST, except that the sorts are restricted to $\{0, \ldots, n - 1\}$. A natural model of $\text{TST}_n$ is defined in the obvious way as a substructure of a natural model of TST.
The interesting theory TNT (the “theory of negative types”) proposed by Hao Wang is defined as TST except that the sorts are indexed by the integers. TNT is readily shown to be consistent (any proof of a contradiction in TNT could be transformed to a proof of a contradiction in TST by raising types) and can be shown to have no natural models.

We define a variant TST\(\lambda\) with types indexed by more general ordinals.

**Parameter of the construction introduced:** We fix a limit ordinal \(\lambda\) for the rest of the paper.

**Definition (type index):** A type index is defined as an ordinal less than \(\lambda\). For purposes of the basic result \(\text{Con}(\text{NF})\) \(\lambda = \omega\) suffices, but for more general conclusions having more type indices available is useful.

**Definition (the theory TST\(\lambda\)):** TST\(\lambda\) is a first-order many-sorted theory with equality and membership as primitive relations and with sorts (referred to traditionally as “types”) indexed by the type indices. A variable may be written in the form \(x^i\) to indicate that it has type \(i\) but this is not required; in any event each variable \(x\) has an associated type\('x'\) \(< \lambda\). In each atomic formula \(x = y\), the types of \(x\) and \(y\) will be equal; in each atomic formula \(x \in y\) the type of \(y\) will be the successor of the type of \(x\).

The axioms of TST\(\lambda\) are axioms of extensionality

\[
(\forall xy. x = y \iff (\forall z. z \in x \leftrightarrow z \in y))
\]

for any variables \(x, y, z\) of appropriate types and axioms of comprehension, the universal closures of all formulas of the form

\[
(\exists A. (\forall x. x \in A \leftrightarrow \phi))
\]

for any variables \(x, A\) of appropriate types and formula \(\phi\) in which the variable \(A\) does not occur.

In TST\(\lambda\), the objects of successor types may be thought of as sets, and the objects of limit types as individuals of various types. Of course, there is not really any interest in TST\(\lambda\) as such without some relationship postulated between types whose indices do not have finite difference.
3 The definition of New Foundations

The definition of New Foundations is motivated by a symmetry of TST.

**Definition (syntactical type-raising):** Define a bijection \( x \mapsto x^+ \) from variables in general to variables with positive type, with the type of \( x^+ \) being the successor of the type of \( x \) in all cases. Let \( \phi^+ \) be the result of replacing all variables in \( \phi \) with their images under this operation: \( \phi^+ \) is clearly well-formed if \( \phi \) is.

**Observations about syntactical type-raising:** If \( \phi \) is an axiom, so is \( \phi^+ \). If \( \phi \) is a theorem, so is \( \phi^+ \). If \( \{x : \phi\} \) is a set abstract, so is \( \{x^+ : \phi^+\} \).

This symmetry suggests that the world of TST resembles a hall of mirrors. Any theorem we can prove about any specific type we can also prove about all higher types; any object we construct in any type has precise analogues in all higher types.

Quine suggested that we should not multiply theorems and entities unnecessarily: he proposed that the types should be identified and so all the analogous theorems and objects at different types should be recognized as being the same. This results in the following definition.

**Definition (the theory NF):** NF is a first-order theory with equality and membership as primitive relations. We suppose for formal convenience that all the variables of the language of TST are also variables of the language of NF (though in the context of NF they do not have types). The axioms of TST are the axiom of extensionality

\[
(\forall xy.x = y \leftrightarrow (\forall z. z \in x \leftrightarrow z \in y))
\]

and axioms of comprehension, the universal closures of all formulas of the form

\[
(\exists A.(\forall x.x \in A \leftrightarrow \phi))
\]

for any formula \( \phi \) which is a well-formed formula of the language of TST and in which the variable \( A \) does not occur.

**Definition (set abstract notation):** We define \( \{x : \phi\} \) (for appropriate formulas \( \phi \)) as the witness (unique by extensionality) to the truth of the comprehension axiom

\[
(\exists A.(\forall x.x \in A \leftrightarrow \phi)).
\]
This is not the way that the comprehension axiom of NF is usually presented. It could make one uncomfortable to define an axiom scheme for one theory in terms of the language of another. So it is more usual to proceed as follows.

**Definition:** A formula $\phi$ of the language of NF is stratified iff there is a bijection $\sigma$ from variables to natural numbers (or integers), referred to as a stratification of $\phi$, with the property that for each atomic subformula $'x = y'$ of $\phi$ we have $\sigma('x') = \sigma('y')$ and for each atomic subformula $'x \in y'$ of $\phi$ we have $\sigma('x') + 1 = \sigma('y')$.

If we were to make more use of stratifications, we would not always be so careful about use and mention. Notice that a formula being stratified is exactly equivalent to the condition that it can be made a well-formed formula of the language of TST by an injective substitution of variables (recalling that all variables of the language of TST are conveniently supposed to also be variables of the language of NF). Of course, if we use the stratification criterion we do not need to assume that we inherit the variables of TST in NF.

**Axiom scheme of stratified comprehension:** We adopt as axioms all universal closures of formulas

\[(\exists A. (\forall x. x \in A \leftrightarrow \phi))\]

for any stratified formula $\phi$ in which the variable $A$ does not occur.

We discourage any philosophical weight being placed on the idea of stratification, and we in fact make no use of it whatsoever in this paper. We note that the axiom of stratified comprehension is equivalent to a finite conjunction of its instances, so in fact a finite axiomatization of NF can be given that makes no mention of the concept of type at all. However, the very first thing one would do in such a treatment of NF is prove stratified comprehension as a meta-theorem. The standard reference for such a treatment is [3].
4 Well-known results about New Foundations

We cite some known results about NF.

NF as a foundation of mathematics is as least as powerful as TST, since all reasoning in TST can be mirrored in NF.

NF acquires a certain philosophical cachet, because it appears to allow the formation of large objects excluded from the familiar set theories (by which we mean Zermelo set theory and ZFC) by the “limitation of size” doctrine which underlies them. The universal set exists in NF. Cardinal numbers can be defined as equivalence classes of sets under equinumerousness. Ordinal numbers can be defined as equivalence classes of well-orderings under similarity. We think that this philosophical cachet is largely illusory.

A consideration which one might take into account at this point is that we have not assumed Infinity. TST without Infinity is weaker than Peano arithmetic. TST with Infinity has the same strength as Zermelo set theory with separation restricted to $\Delta^0_0$ formulas (Mac Lane set theory), which is a quite respectable level of mathematical strength.

In [16], 1954, Specker proved that the Axiom of Choice is refutable in NF, which has the corollary that Infinity is a theorem of NF, so NF is as strong as Mac Lane set theory with the substantial practical inconvenience for mathematics as usually practiced of refuting Choice. It was this result which cast in sharp relief the problem that a relative consistency proof for NF had never been produced, though the proofs of the known paradoxes do not go through.

A positive result of Specker in [17], 1962, served to give some justification to Quine’s intuition in defining the theory, and indicated a path to take toward a relative consistency proof.

**Definition (ambiguity scheme):** We define the Ambiguity Scheme for TST (and some other similar theories) as the collection of sentences of the form $\phi \iff \phi^+$. 

**Theorem (Specker):** The following assertions are equivalent:

1. NF is consistent.
2. TST + Ambiguity is consistent
3. There is a model of TST with a “type shifting endomorphism”, that is, a map which sends each type $i$ bijectively to type $i + 1$
and commutes with the equality and membership relations of the model (it is also equivalent to assert that there is a model of TNT with a type shifting automorphism).

The equivalence also applies to any extension of TST which is closed as a set of formulas under syntactical type-raising and the corresponding extension of NF, and to other theories similar to NF (such as the theories TSTU and NFU described in the next section).
5 Consistency of NFU

In [10], 1969, Jensen produced a very substantial positive result which entirely justified Quine's proposal of NF as an approach to foundations of mathematics, with a slight adjustment of detail.

Define TSTU as a theory with almost the same language as TST (it is convenient though not strictly necessary to add a primitive constant $\emptyset^{i+1}$ in each positive type with the additional axioms $(\forall x. x \not\in \emptyset^{i+1})$ for $x$ of each type $i$) with the same comprehension scheme as TST and with extensionality weakened to allow atoms in each positive type:

**Axiom (weak extensionality, for TSTU):**

\[
(\forall xyz. z \in x \rightarrow (x = y \iff (\forall w. w \in x \leftrightarrow w \in y)))
\]

**Definition (sethood, set abstracts for TSTU):** Define set$(x)$ ($x$ is a set) as holding iff $x = \emptyset \lor (\exists y. y \in x)$ [we are using polymorphism here: the type index to be applied to $\emptyset$ is to be deduced from the type of $x$]. Define \{x : \phi\} as the witness to the appropriate comprehension axiom as above, with the qualification that if it has no elements it is to be taken to be $\emptyset$.

**Definition (natural models of TSTU):** It is convenient to reverse the direction of the functions $f_i$. A natural model of TSTU is determined by a sequence of sets $X_i$ indexed by natural numbers and a sequence of injections $f_i : P(X_i) \rightarrow X_{i+1}$. The interpretation of the language of TSTU in a natural model is as the interpretation of the language of TST in a natural model, except that $x^i \in y^{i+1}$ is interpreted as $(\exists z. x^i \in z \land f_i(z) = y^{i+1})$. We interpret $\emptyset^{i+1}$ as $f_i(\emptyset)$. It is straightforward to establish that the interpretations of the axioms of TSTU are true in a natural model of TSTU.

Define NFU as the untyped theory with equality, membership and the empty set as primitive notions and with the axioms of weak extensionality, the scheme of stratified comprehension, and the axiom $(\forall x. x \not\in \emptyset)$.

Jensen’s proof rests on the curious feature that it is possible to skip types in a natural model of TSTU in a way that we now describe. For generality it is advantageous to first present natural models with types indexed by general ordinals less than $\lambda$. 
Definition (natural models of $\text{TSTU}_\lambda$): A natural model of $\text{TSTU}_\lambda$ is determined by a sequence of sets $X_i$ indexed by ordinals $i < \lambda$ and a system of injections $f_{i,j} : \mathcal{P}(X_i) \to X_j$ for each $i < j$. Interpretations of the language of $\text{TSTU}$ in a natural model of $\text{TSTU}_\lambda$ are provided with a strictly increasing sequence $s_i$ of type indices as a parameter: they are as the interpretation of the language of TST in a natural model, except that each variable $x^i$ of type $i$ is to be interpreted as a variable $x^{s_i}$ restricted to the set $X_{s_i}$ and a membership formula $x^i \in y^{i+1}$ is interpreted as $(\exists z. x^{s_i} \in z \land f_{s_i,s_{i+1}}(z) = y^{s_j})$. It is straightforward to establish that the axioms of $\text{TSTU}$ have true interpretations in each such scheme. The special constant $\emptyset^{i+1}$ is interpreted as $f_{s_i,s_{i+1}}(\emptyset)$.

Theorem (Jensen): $\text{NFU}$ is consistent.

Proof of theorem: Clearly there are natural models of $\text{TSTU}_\lambda$ for each $\lambda$: such models are supported by any sequence $X_i$ indexed by $i < \lambda$ with each $X_i$ at least as large as $\mathcal{P}(X_j)$ for each $j < i$. Fix a natural model. Let $\Sigma$ be any finite set of sentences of the language of $\text{TSTU}$. Let $n$ be a strict upper bound on the type indices appearing in $\Sigma$. Define a partition of $[\lambda]^n$: the compartment into which an $n$-element subset $A$ of $\lambda$ is placed is determined by the truth values of the sentences in $\Sigma$ in the interpretation of $\text{TSTU}$ in the given natural model parameterized by any sequence $s$ such that $s|n = A$ (the truth values of interpretations of sentences in $\Sigma$ are determined entirely by this restriction of $s$). This partition of $[\lambda]^n$ into no more than $2^{[\Sigma]}$ compartments has an infinite homogeneous set $H$, which includes the range of some strictly increasing sequence $h$ of type indices. The interpretation of $\text{TSTU}$ determined by $h$ in the natural model satisfies $\phi \leftrightarrow \phi^+$ for each $\phi \in \Sigma$. Thus every finite subset of the Ambiguity Scheme is consistent with $\text{TSTU}$, whence the entire Ambiguity Scheme is consistent with $\text{TSTU}$, and by the results of Specker (the methods of whose proof apply as well to $\text{TSTU}$ and $\text{NFU}$ as they do to $\text{TST}$ and $\text{NF}$), $\text{NFU}$ is consistent.

Corollary: $\text{NFU}$ is consistent with Infinity and Choice. It is also consistent with the negation of Infinity.

Proof: If $X_0$ is infinite, all interpretations in the natural model will satisfy Infinity. If Choice holds in the metatheory, all interpretations in the natural model will satisfy Choice. If all $X_i$’s are finite (which is only
possible if \( \lambda = \omega \) the negation of Infinity will hold in the interpreted theory.

**Proof without appealing to Specker outlined:** Suppose that \( \leq \) is a well-ordering of the union of the \( X_i \)'s. Add the relation \( \leq \) to the language of TSTU with the same type rules as identity, and interpret \( x^i \leq y^{i+1} \) as \( x^i \leq y^{i+1} \). We obtain as above a consistency proof for TSTU + Ambiguity + existence of a primitive well-ordering \( \leq \) of each type (which can be mentioned in instances of ambiguity). The relation \( \leq \) can be used to define a Hilbert symbol: define \((\theta x : \phi)\) as the \( \leq \)-least \( x \) such that \( \phi \), or \( \emptyset \) if there is no such \( x \). Now construct a model of TSTU + Ambiguity + primitive well-ordering \( \leq \) with the same theory consisting entirely of referents of Hilbert symbols (a term model). The Ambiguity Scheme justifies abandoning the distinction between a Hilbert symbol \((\theta x : \phi)\) and its type-raised version \((\theta x^+ : \phi^+)\) in all cases and one obtains a model of NFU with a primitive well-ordering \( \leq \).

The consistency proof for NFU assures us that the usual paradoxes of set theory are indeed successfully avoided by NF, because NFU avoids them in exactly the same ways. This does not rule out NF falling prey to some other unsuspected paradox. Further, though this is not our business here, the consistency proof for NFU shows that NFU is a reasonable foundation for mathematics: NFU + Infinity + Choice is a reasonably fluent mathematical system with enough strength to handle almost all mathematics outside of technical set theory, and extensions of NFU with greater consistency strength are readily obtained from natural models of TST\( \lambda \) for larger ordinals \( \lambda \) (see [7]).
6 Tangled type theories

In [5], 1995, we pointed out that the method of proof of Jensen can be adapted to NF, establishing the equiconsistency of NF with a certain type theory. This does not immediately give a relative consistency proof for NF, because the type theory under consideration is very strange, and not obviously consistent.

**Definition (the theory TTT\(\lambda\)):** TTT\(\lambda\) (tangled type theory with \(\lambda\) types) is a first-order many-sorted theory with equality and membership as primitive relations and with sorts (referred to traditionally as “types”) indexed by the type indices. A variable may be written in the form \(x^i\) to indicate that it has type \(i\) but this is not required; in any event each variable \(x\) has an associated type \(\text{type}(x^i) < \lambda\). In each atomic formula \(x = y\), the types of \(x\) and \(y\) will be equal; in each atomic formula \(x \in y\) the type of \(y\) will be strictly greater than the type of \(x\).

Let \(s\) be a strictly increasing sequence of type indices. Provide a map \((x \mapsto x^s)\) whose domain is the set of variables of the language of TST and whose restriction to type \(i\) variables \(x\) is a bijection from the collection of type \(i\) variables in the language of TST to the collection of type \(s_i\) variables in the language of TTT\(\lambda\). For each formula \(\phi\) in the language of TST, define \(\phi^s\) as the result of replacing each variable \(x\) in \(\phi\) with \(x^s\). We observe that \(\phi^s\) will be a formula of the language of TTT\(\lambda\).

The axioms of TTT\(\lambda\) are exactly the formulas \(\phi^s\) where \(s\) is any strictly increasing sequence of type indices and \(\phi\) is an axiom of TST.

**Theorem:** TTT\(\lambda\) is consistent iff NF is consistent.

**Proof of theorem:** If NF is consistent, one gets a model of TTT\(\lambda\) by using the model of NF (or if one prefers, disjoint copies of the model of NF indexed by \(\lambda\)) to implement each type, and defining the membership relations of the model in the obvious way.

Suppose that TTT\(\lambda\) is consistent. Fix a model of TTT\(\lambda\). Let \(\Sigma\) be a finite set of sentences of the language of TST. Let \(n\) be a strict upper bound on the (natural number) type indices appearing in \(\Sigma\). We define a partition of \([\lambda]^n\): the compartment into which an \(n\)-element subset \(A\) of \(\lambda\) is placed is determined by the truth values of the sentences \(\phi^s\) for \(\phi \in \Sigma\) for strictly increasing sequences \(s\) of type indices such
that \( s[n] = A \). This partition of \([\lambda]^n\) into no more than \(2^{[\Sigma]}\) compartments has an infinite homogeneous set \( H \) which includes the range of a strictly increasing sequence \( h \) of type indices. The interpretation of TST obtained by assigning to each formula \( \phi \) of the language of TST the truth value of \( \phi^h \) in our model of tangled type theory will satisfy each instance \( \phi \leftrightarrow \phi^+ \) of Ambiguity for \( \phi \in \Sigma \). It follows by compactness that the Ambiguity Scheme is consistent with TST, and so by the results of Specker that NF is consistent.

**Proof without appealing to Specker outlined:** Suppose that \( \leq \) is a well-ordering of the union of the types of our model of tangled type theory (this is not an internal relation of the model in any sense). Add the relation symbol \( \leq \) to the language of TST, with the same type rules as identity, and interpret an atomic formula \( x \leq y \) in the extended language of TST as meaning \( x^s \leq y^s \) in the model of tangled type theory. We obtain as above, using this extended language to define our partition, a consistency proof for TST + Ambiguity + existence of a primitive relation \( \leq \) of each type (which can be mentioned in instances of ambiguity, but which cannot be used in instances of comprehension) which is a linear order and a well-ordering in a suitable external sense (any definable nonempty class has a \( \leq \)-least element) and which can be used to define a Hilbert symbol: we can define \( (\theta x : \phi) \) as the \( \leq \)-least \( x \) such that \( \phi \), or \( \emptyset \) if there is no such \( x \). Now construct a model of TST + Ambiguity + primitive “external well-ordering” \( \leq \) with the same theory consisting entirely of referents of Hilbert symbols (a term model). The Ambiguity Scheme justifies abandoning the distinction between a Hilbert symbol \( (\theta x : \phi) \) and its type-raised version \( (\theta x^+ : \phi^+) \) in all cases and one obtains a model of NF with a primitive well-ordering \( \leq \).

Examination of our presentation of Jensen’s consistency proof for NFU should reveal that this is an adaptation of the same method to the case of NF. In fact, our “natural model of TSTU\( \lambda \)” above can readily be understood as a model of TTTU\( \lambda \).

It should also be clear that TTT\( \lambda \) is an extremely strange theory. We cannot possibly construct a “natural model” of this theory, as each type is apparently intended to implement a “power set” of each lower type, and Cantor’s theorem precludes these being honest power sets.
6.1 ω- and α-models from tangled type theory

Jensen continued in his original paper [10] by showing that for any ordinal \(\alpha\) there is an \(\alpha\)-model of NFU. We show that under suitable conditions on the size of \(\lambda\) and the existence of sets in a model of TTT\(_\lambda\), this argument can be reproduced for NF.

We quote the form of the Erdős-Rado theorem that Jensen uses: Let \(\delta\) be an uncountable cardinal number such that \(2^\beta < \delta\) for \(\beta < \delta\) (i.e., a strong limit cardinal). Then for each pair of cardinals \(\beta, \mu < \delta\) and for each \(n > 1\) there exists a \(\gamma < \delta\) such that for any partition \(f: [\gamma]^n \rightarrow \mu\) there is a set \(X\) of size \(\beta\) such that \(f\) is constant on \([X]^n\) (\(X\) is a homogeneous set for the partition of size \(\beta\)).

Let \(\lambda\) be a strong limit cardinal with cofinality greater than \(2^{\alpha}\). Our types in TTT will be indexed by ordinals \(< \lambda\) as usual. We make this stipulation about \(\lambda\) only for this subsection.

We assume the existence of a model of TTT\(_\lambda\) in which each type contains a well-ordering of type \(\alpha\) (from the standpoint of the metatheory as well as internally), elements of whose domain are explicitly named with a component representing the type and an index \(< \alpha\). Note that this does not imply directly that the model of NF we eventually obtain, which comes from an application of compactness, will actually contain such an order: many non-standard elements might be added to it, and it might cease to be externally a well-ordering at all. We need to do extra work.

Let \(\Sigma_n\) be the collection of all sentences of the language of TST\(_n\) which begin with an existential quantifier restricted to the domain of the special well-ordering of order type \(\alpha\) in some type, in a language which includes special constants in each type, indexed by ordinals \(< \alpha\), representing the elements of the special well-ordering of type \(\alpha\) in that type, indexed of course in order. Let the partition determined by \(\Sigma_n\) make use not of the truth values of the formulas in \(\Sigma_n\), but of the indices \(< \alpha\) of the minimally indexed witnesses to the truth of each formula, or \(\alpha\) if they are false. The Erdős-Rado Theorem in the form cited tells us that we can find homogeneous sets of any desired size less than \(\lambda\) for this partition, and moreover (because of the cofinality of \(\lambda\)) we can find homogeneous sets of any desired size with the same witnesses taken from \(\alpha\) for each sentence in \(\Sigma_n\). This allows us to see that ambiguity of \(\Sigma_n\) is consistent, and moreover consistent with standard values for witnesses to each of the formulas in \(\Sigma\). We can then extend the determination of truth values and witnesses as many times as desired, because
if we expand the set of formulas considered to $\Sigma_{n+1}$ and partition $(n + 1)$-element sets instead of $n$-element sets, we can restrict our attention to a large enough set homogeneous for the previously given partition to ensure homogeneity for the partition determined by the larger set of formulas. After we carry out this process for each $n$, we obtain a full description of a model of TST + Ambiguity with standard witnesses for each existentially quantified statement over the domain of a special well-ordering of type $\alpha$. We can reproduce our Hilbert symbol trick (add a predicate representing a well-ordering of our model of TTT to the language as above) to pass to a model of NF with the same characteristics.
7 Tangled webs of cardinals

In this section, we convert the weird type theory TTTλ into a (still weird) extension of ordinary set theory (Mac Lane set theory, Zermelo set theory or ZFC) whose consistency would imply the consistency of NF. We do this by examining what TTTλ says internally about the relationships between the cardinalities of its types, which from an external standpoint look quite impossible.

The theory TST itself contains natural models of the theories TSTn. The cardinal |A| of a set A in TST is defined as the collection of all sets b equinumerous with A (this will be one type higher than A). We define i(x) as \{x\} and i\{x\} as \{y : y \in x\}. We define a natural model of TSTn as determined by a finite sequence Xi (the index i ranging from 0 to n − 1) and maps fi (i ranging from 0 to n − 2) where fi : i\cdot Xi+1 → P(Xi) is a bijection and a membership sentence x\in y\cdot i+1 is interpreted as x\in fi({yi}). The use of i here serves to handle the fact that all the Xi’s need to live at the same type in TST. A natural model of TST in TST can be defined in the same way: but of course TST cannot prove the existence of a natural model of TST internally (though it is consistent with TST that there be such a model, this hypothesis essentially strengthens the theory). Now a model of TSTn internal to TST is defined by taking Xi = {i\cdot n-i(x) : x = x}, the set of (n − i)-fold singletons [clearly the common type of the Xi’s is at least type n + 1], and letting fi map each \{i\cdot n-(i+1)(x)\} (singleton of an element of Xi+1) to \{i\cdot n-i(y) : y \in x\} (the set of elements of Xi which are suitably iterated singletons of elements of x).

The same internalization of natural models can be carried out in TTTλ but gives a much more complex system of cardinals. For each pair of type indices i < j the map ui,j sends each type i object x to its singleton in type j. Iterated singleton maps can conveniently be indexed by nonempty finite subsets of λ. Where A is a finite subset of λ which is nonempty, define A∗ as A \ {\max(A)}. Define iA(x) as i_{\max(A∗),\max(A)}(iA∗(x)) where this is defined, with i\{i\}(x) = x for x of type i.

We can now define a sort of ramified natural model consisting of all XA = \{iA(x) : x = x\} for finite subsets A having a common maximum element α (so that all of the elements of the sets we are discussing are of the same type). Further reasoning will be conducted in any fixed sequence of types above type α. There is no reason to believe that the system of sets XA makes up a set unless α is finite, but we can nonetheless say some interesting things
about these sets. Define \( A_1 \) as \( A \setminus \{ \min(A) \} \). We can show that \( \iota^\ast X_{A_1} \) is the same size as \( \mathcal{P}(X_A) \) for each \( A \) just as we showed that \( \iota^\ast X_{i+1} \) is the same size as \( \mathcal{P}(X_i) \) in TST.

Notice that the first-order theory of the natural model of \( \text{TST}_n \) with base type of cardinal \( |X_A| \) will be determined by the first \( n \) elements of \( A \), at least with regard to concrete sentences (the internalized theory of this natural model might contain nonstandard sentences which behave oddly) since from an external standpoint we can see it as a copy of the theory of \( \text{TST}_n \) as interpreted in the types of our model of \( \text{TTT}_\lambda \) indexed by the smallest \( n \) elements of \( A \).

Avoiding further internal reasoning in \( \text{TTT}_\lambda \), we transition to ordinary set theory. Since we are working in set theory without Choice, we note without going into details that we will use Scott’s definition of cardinal number, which works for cardinalities of non-well-orderable sets. It is useful to note that the definition stated below and the theorem proved are motivated by the reasoning in tangled type theory described above but do not depend on them.

**Definition (extended type index, operations on extended type indices):**

We define an extended type index as a nonempty finite subset of \( \lambda \). For any extended type index \( A \), we define \( A_0 \) as \( A \), \( A_1 \) as \( A \setminus \{ \min(A) \} \) and \( A_{n+1} \) as \( (A_n)_1 \) when this is defined.

**Definition (tangled web of cardinals):** A tangled web of cardinals is a function \( \tau \) from extended type indices to cardinals with the following properties:

- **naturality:** For each \( A \) with \( |A| > 2 \), \( 2^{\tau(A)} = \tau(A_1) \).
- **elementarity:** For each \( A \) with at least \( n \) elements, the first-order theory of the natural model of \( \text{TST}_n \) with base type \( \tau(A) \) is completely determined by the set \( A \setminus A_n \) of the smallest \( n \) elements of \( A \).

**Theorem:** If there is a tangled web of cardinals, \( \text{NF} \) is consistent.

**Proof of theorem:** Suppose that we are given a tangled web of cardinals \( \tau \). Let \( \Sigma \) be a finite set of sentences of the language of TST. Let \( n \) be a strict upper bound on the natural number type indices appearing in \( \Sigma \). Define a partition of \( [\lambda]^n \): the compartment in which an \( n \)-element set
A is placed is determined by the truth values of the sentences in $\Sigma$ in the natural model of TST$_n$ with base type of size $\tau(A)$. This partition of $[\lambda]^n$ into no more than $2^{2^n}$ compartments has a homogeneous set $H$ of size $n + 1$. The natural model of TST with base type $\tau(H)$ satisfies all instances $\phi \leftrightarrow \phi^+$ of Ambiguity for $\phi \in \Sigma$: type 1 of this model is of size $2^{\tau(H)} = \tau(H_1)$ and the theory of the natural models of TST with base type $\tau(H_1)$ decides the sentences in $\Sigma$ in the same way that the theory of the natural models of $\tau(H)$ decides them by homogeneity of $H$ for the indicated partition and the fact that the first order theory of a model of TST$_n$ with base type of size $\tau(A)$ is determined by the smallest $n$ elements of $A$. Thus any finite subset of the Ambiguity Scheme is consistent with TST, so TST + Ambiguity is consistent by compactness, so NF is consistent by the results of Specker.

There is no converse result: NF itself proves the existence of concrete finite fragments of tangled webs as TTT$_{\lambda}$ does.

We should expect by considering the two theorems of Specker which we have cited that existence of a tangled web of cardinals is inconsistent with Choice. We have elsewhere given an explicit proof of this with some instructive analogies with Specker’s disproof of Choice in NF, but it would take us too far afield to reproduce this here.
8 Preliminaries and cardinal parameters

Our context: We work in ZFA, initially with choice.

Cardinal parameters of the construction and related definitions:

the parameter $\kappa$: small and large sets: We choose a regular uncountable cardinal $\kappa$. Sets of cardinality $< \kappa$ are called small sets; all other sets are called large sets.

the parameter $\lambda$: extended type indices with operations and relations:
We choose a limit cardinal $\lambda$. Elements of $\lambda$ are called type indices. Finite subsets of $\lambda$ are called “extended type indices”. If $A$ is an extended type index, we define $A_1$ as $A \setminus \{\min(A)\}$ if $A$ is nonempty; otherwise it is undefined. We define $A_0$ as $A$ and $A_{n+1}$ as $(A_n)_1$ when this is defined. We define $A << B$ as holding when $(\exists n > 0 : A_n = B)$. We say if $A << B$ that $A$ properly downward extends $B$. The notation $A \ll B$ (A downward extends B) means $A << B \lor A = B$.

the parameter $\mu$: We choose a cardinal $\mu$ which is strong limit with cofinality greater than $|\lambda|$ and greater than $\kappa$. 
9 Definition of abstract parent functions

Definition (abstract parent function): We state the conditions required for a function $\Pi$ to be an “abstract parent function”. The reader needs to notice that this definition is rather long. Some clauses contain explicit assertions which are part of the definition of the notion “abstract parent function”; some clauses contain auxiliary definitions used in following discussion and further parts of the main definition; some clauses may report facts about abstract parent functions provable on the basis of the definition thus far.

regular atoms: The set $\bigcup \bigcup \text{dom}(\Pi)$ is a set of atoms $A_{\Pi}$ (the subscript $\Pi$ will generally be omitted) which we call the collection of $\Pi$-regular atoms. In general, though I do not specifically note this in most cases, all concepts defined in the internals of the definition of “abstract parent function” are relative to a specific abstract parent function understood from context and can be decorated with the name of a specific abstract parent function if necessary.

clans: The set $A$ is partitioned into sets called “clans”. The set $A$ is of size $\mu$. Each clan is of size $\mu$. There are $\leq |\lambda|$ clans.

local cardinals: For any subset $N$ of a clan $C$ such that $|N| = \kappa$, we define the “local cardinal” $[N]$ as the collection of all subsets of $C$ with small symmetric difference from $N$. Notice that all elements of local cardinals are of cardinality exactly $\kappa$.

litters: All elements of the domain of $\Pi$ are local cardinals. There is a choice set from $\text{dom}(\Pi)$ which is a partition of $A$ (and so subsets of this choice set partition each clan). We fix one such choice set and call its elements “litters”. It should be clear that there are $\mu$ litters included in each clan and that each litter is of cardinality exactly $\kappa$. Define $\mathbb{K}$ as the set of all local cardinals of litters, the domain of $\Pi$.

It should be noted that while it is convenient to choose a specific choice set from $\text{dom}(\Pi)$ to identify as the collection of litters, there is nothing special about any particular such choice set, and no concept we make essential use of depends on which choice set is used. (note added 5/21/2016: it is not clear that this is true of
the property asserting that every object has an extended strong support: this may depend on the selection of near-litters which are litters; I believe I now have a proof of this, but nothing hinges on this – if necessary we can add the specific choice of which near-litters are litters as a component of the abstract parent function).

near-litters: A set which belongs to an element of $\text{dom}(\Pi)$, or equivalently, a set which is a subset of a clan and has small symmetric difference from a litter, is called a near-litter.

further conditions on abstract parent functions: For each clan $C$ there is an extended type index $c$ such that

$$\Pi^\pi(P^2(C) \cap \mathbb{K}) = \{(x, c) : (x, c) \in \text{rng}(\Pi)\}.$$ 

We may denote $C$ as clan$[c]$, and we may denote $c$ as index$[C]$. The extended type index $c$ may be called a “clan index” or the clan index of $C$.

This condition makes it clear that the partition of $A$ into clans is definable from $\Pi$.

We make no assumption in this definition that all extended type indices are clan indices.

notions of parent: For any litter $L \subseteq \text{clan}[c]$, we may refer to $\pi_1(\Pi([L]))$ as the parent of $L$, or of an element of $L$, or of a near-litter $N$ with $[N] = [L]$. We may refer to $\pi_1^\pi(\Pi^\pi(P^2(\text{clan}[c]) \cap \mathbb{K}))$ as the “parent set” of clan$[c]$. We may use the notation parents$[c]$ for this set.

stipulation concerning irregular atoms: Atoms in the transitive closure of $\Pi$ which do not belong to $A$ are called irregular atoms. We stipulate as part of the definition of abstract parent function that all irregular atoms are parents of litters.

permutations of atoms: Any permutation $\pi$ of the set of atoms is by convention extended to the entire universe of sets by the rule $\pi(A) = \pi^\pi A$.

allowable permutations: A (II)-allowable permutation is defined as a permutation of the atoms whose action fixes $\Pi$. Note that conditions already imposed ensure that an allowable permutation will fix each clan and parent set (the fact that indices of clans are
pure sets ensures this). For any near-litter \( N \) and allowable permutation \( \pi \), note that it is a consequence of this definition that \( \pi(N) \) must be a near-litter included in the same clan as \( N \), and \( \Pi([\pi(N)]) = \Pi(\pi([N])) = \pi(\Pi([N])) \).

**support set, support of an object, symmetry:** A support set is a small set of regular atoms and near-litters with the property that distinct near-litters in the set are disjoint. An object \( x \) has support \( S \) iff each allowable permutation \( \pi \) such that \( \forall s \in S. \pi(s) = s \) also satisfies \( \pi(x) = x \). An object is symmetric iff it has a support. An object is hereditarily symmetric iff it is an atom or it is symmetric and all elements of its transitive closure are symmetric.

Note that if \( x \) has support \( S \) and \( \pi \) is an allowable permutation, the value of \( \pi(x) \) is determined by \( \pi[S] \) and \( x \), because if \( \pi \) and \( \pi' \) are allowable permutations and \( \pi[S] = \pi'[S] \), the function \( \pi^{-1} \circ \pi' \) will fix \( x \), as it fixes all elements of \( S \), so \( \pi(x) = \pi'(x) \).

**objects known to be symmetric:** It is evident that regular atoms and near-litters are symmetric with support their own singletons. It is evident that \( \Pi \) is symmetric, and that each clan and each parent set is symmetric. An irregular atom is symmetric with support the singleton of any litter with the irregular atom as parent.

**further conditions on abstract parent functions:** As part of the definition of abstract parent function, we stipulate that the function \( \Pi \) is required not just to be \( \Pi \)-symmetric (obvious from the definition) but also \( \Pi \)-hereditarily symmetric. This is equivalent to an additional requirement that elements of parent sets are hereditarily symmetric.

**strong support set:** A strong support is a pair \( S = (D_S, <_S) \) where \( D_S \) is a support set and \( <_S \) is a strict well-ordering of \( D_S \) such that for each atom \( x \in D_S \) and near-litter \( N \in D_S \) with \( x \in N \) (there can be at most one such \( N \) for each such \( x \)) we have \( N <_S x \), and for each near-litter \( N \in D_S \), \( \{ y : y <_S x \} \) is a support for \( \pi_1(\Pi([N])) \), unless \( \pi_1(\Pi([N])) \) is an atom not belonging to \( D_S \). If \( (D_S, <_S) \) is a strong support and \( D_S \) is a support of the object \( x \), we say that \( S = (D_S, <_S) \) is a strong support for \( x \).

If \( D_S \) is nonempty, we define \( S(0) \) as the \( <_S \)-minimal (or only)
element of $D_S$ and if $D_S$ has at least two elements, we define $S(1)$ as the $<_S$-minimal (or only) element of $D_S \setminus \{S(0)\}$ (to avoid silly issues about order types of strict well-orderings on sets with 0 and 1 element). For each other ordinal $\alpha$ less than the order type of $<_S$, we define $S(\alpha)$ as the unique $s \in D_S$ such that the restriction of $<_S$ to $\{t : t <_S s\}$ has order type $\alpha$.

**an additional stipulation (existence of strong supports):** Clearly every regular atom has a strong support (its own singleton paired with the empty relation). We stipulate as part of the definition of abstract parent function that every set belonging to a parent set has a strong support, from which it follows immediately that every near-litter has a strong support.

We stipulate further that every set belonging to a parent set has a strong support in which each near-litter element is a litter and in which each litter with atomic parent not in the support has an irregular atomic parent and each atom in the strong support belongs to a litter in the strong support: call this an extended strong support.

**complexity condition on elements of parent sets:** We state a further stipulation in the definition of abstract parent function. For each clan index $c$, define $\rho(c)$ as $\min(c)$ if $c$ is nonempty and as $\lambda$ otherwise. Define for objects $x$ which are atoms, subsets of clans (including but not restricted to near-litters), or sets belonging to parent sets the (Π-)complexity of $x$ to be the smallest ordinal $\alpha$ such that for some $c$, $\rho(c) = \alpha$ and $x$ belongs to $\text{clan}[c]$ or is a subset of $\text{clan}[c]$ or is a set (not an atom) belonging to $\text{parents}[c]$. For other objects, the complexity is not defined.

We require further that

1. For any set element $x$ of $\text{parents}[c]$ and strong support $(D_S,<_S)$ for $x$, the restriction $(D^c_S,<_S \cap (D^c_S \times D^c_S))$, where $D^c_S$ is the collection of elements of $D_S$ of complexity $\leq \rho(c)$, is also a strong support for $x$.

2. Further, any set belonging to $\text{parents}[c]$ has all of its elements of complexity $< \rho(c)$ (which includes the condition that they must be the sorts of objects that have complexity). [A corollary of this is that nonempty subsets of clans cannot be
elements of parent sets; this is not especially important and things could be arranged differently and have been in other versions of this argument].

Note that this allows us to observe that if \((D_S, <_S)\) is a strong support, so is \((D_S, <'_S)\), where \(x <'_S y\) is defined as “\(x <_S y\) and \(y\) is an atom or \(x\) is an atom or the complexity of \(x\) is less than or equal to the complexity of \(y\)”.

**extension property:** We say that an atom \(x\) is an exception of an allowable permutation \(\pi\) iff \(x\) is an element of a litter \(L\) and either \(\pi(x)\) does not belong to the litter belonging to \(\pi([L])\) or \(\pi^{-1}(x)\) does not belong to the litter belonging to \(\pi^{-1}([L])\).

We say that \(\pi_0\) is a locally small bijection iff it is a bijection on a set of atoms with the property that if \(x\) is an atom and \(C\) is a clan, \(x \in C \leftrightarrow \pi_0(x) \in C\) [note that this implies that \(x\) is irregular iff \(\pi_0(x)\) is irregular], and further the intersection of the domain of \(\pi_0\) with each clan is small.

We stipulate as part of the definition of abstract parent function that every locally small bijection can be extended to an allowable permutation. We call this the “extension property”.

**definition completed:** This completes the definition of abstract parent function.
10 Results about abstract parent functions

Parent function usually understood from context: In the discussion that follows, we usually fix a particular $\Pi$ and define all concepts appearing in the previous definition relative to the fixed $\Pi$ understood from the context.

A fact about strong support sets: Let $S = (D_S, <_S)$ be a strong support set. Note that for a near-litter $N \in D_S$,

$$
\{ y : y <_S N \} , <_S \cap (\{ y : y <_S N \} \times \{ y : y <_S N \})
$$

is a strong support for $[N]$ unless the parent of $N$ is an atom not in $D_S$. We denote this strong support as $S_N = (D_{S_N}, <_{S_N})$. We further define $S_{cN}$ as the restriction of $S_N$ to elements with complexity $\leq \rho(c)$ in the obvious sense, which will also be a strong support for $[N]$ if $N \subseteq \clan[c]$ and further define $(S_{cN})^*$ as the restriction of $S_N$ to atoms in $\clan[c]$.

Extension of support sets to strong supports: We argue that every support set $S$ can be extended to a set $D_T$ such that for some $<_T$, $T = (D_T, <_T)$ is a strong support. Let $<_S$ be a strict well-ordering of $S$. For each $s \in S$, choose a strong support $\sigma(s) = (D_{\sigma(s)}, <_{\sigma(s)})$ for $s$, subject to the constraint that if $s$ belongs to $\sigma(t)$ for any $t <_S s$, we let $\sigma(s)$ be the appropriate restriction of $\sigma(t)$. We then obtain a strong support $T = (D_T, <_T)$ where $D_T$ is the union of the sets $D_{\sigma(s)}$ for $s \in S$ and $s <_T t$ holds iff $(\exists u. (u = t \lor u <^S t) \land s \in \sigma(u) \land s \neq t)$.

Similarly, support sets in which all near-litter elements are litters can be extended to extended strong supports. Any near-litter has an extended strong support constructible by extending the support set for it consisting of the litter from which it has small symmetric difference and the atoms in that small symmetric difference.

(What follows is a technical device: the only effect if it doesn’t work is that the selection of which near-litters are litters is actually operative): Further, a change of which near-litters are selected as the litters does not break the existence of extended strong supports. An extended strong support with respect to the old scheme of litters can be transformed to an extended strong support with respect to the new scheme.
For any old litter, there is a small collection of old litters with the property that the new litter with small symmetric difference from any element of the collection is included in the union of the collection: call this the overlap closure of the litter. Extend the original extended strong support so that it contains all elements of the overlap closure of each of its litter elements, then replace each old litter in it with the new litter with small symmetric difference from it and the atoms in this small symmetric difference.

**global support orders and the extension property:** We define a global support order by choosing a well-ordering $<^g$ of all atoms and litters, then for each atom or litter $s$, choosing an extended strong support $\sigma(s) = (D_{\sigma(s)}, <_{\sigma(s)})$ for $s$, [which will contain the litter containing $s$ if $s$ is an atom, and which will contain no near-litter which is not a litter] subject to the constraint that if $s$ belongs to $\sigma(t)$ for any $t <^g s$, we let $\sigma(s)$ be the appropriate restriction of $\sigma(t)$, then obtaining the global support order $<_G: s <_G t$ holds iff $(\exists u. (u = t \lor u <^g t) \land s \in \sigma(u) \land s \neq t)$. This is constructed exactly in the way we extended support sets to strong supports above, but what we obtain is too large to be a strong support.

We use recursion on a global support order to construct witnesses to the extension property in a canonical way which will be useful later.

Choose a locally small bijection $\pi_0$. For each pair of litters $L, M$ included in the same clan, choose a bijection $\pi_{L,M}$ from $L \setminus \text{dom}(\pi_0)$ to $M \setminus \text{dom}(\pi_0)$.

Now define values for the allowable permutation $\pi$ extending $\pi_0$ by recursion on the global support order used. At an irregular atom, either apply $\pi_0$ or apply the identity. At any regular atom, one has already computed $\pi$ at its parent: either apply $\pi_0$ or apply $\pi_{L,\pi(L)^*}$ where the atom belongs to the litter $L$ and $\pi(L)^*$ is the litter belonging to $\pi([L])$.

At any litter, we can already compute the image of the parent because we can compute the image under $\pi$ of elements of a support for it appearing earlier in the global order: we can then compute the image of the litter by computing the image of each of its elements using either $\pi_0$ or the appropriate $\pi_{L,\pi(L)^*}$. We actually have to appeal to the extension property at this point in the construction (so without immense further elaboration this argument is not a proof of the extension property):
since there is a support for the parent of the litter at each element of which we have computed the value of \( \pi \), we have uniquely determined the value of \( \pi \) at the parent of the litter – if there is any allowable permutation at all with these values, and we have to appeal to the extension property to argue that there is such a permutation. The locally small bijection that we extend will involve all atoms in the support of the parent of the litter in question and the elements of all orbits in \( \pi_0 \) meeting litters in the support of the parent of the litter in question (these will cover all exceptions, since our recursive process never calls for any exceptions outside the domain of \( \pi_0 \)).

**nice function on a strong support:** If \( (D_S, <_S) \) is a strong support, a map \( \pi_0 \) with domain \( D_S \) is said to be a nice function (interchangeably, nice map) on \( (D_S, <_S) \) if

1. \( \pi_0 \) is injective and sends disjoint near-litters to disjoint near-litters,
2. \( \pi_0(x) \) belongs to the same clan as \( x \) if either is an atom,
3. if \( x < y \) then \( \pi_0(x) \in \pi_0(y) \), for any atom \( x \) and near-litter \( y \) in \( D_S \).
4. and if \( x \) is a near-litter whose parent is a set, then the parent of \( \pi_0(x) \) is the image of the parent of \( x \) under an allowable permutation extending \( \pi_0[\{y : y <_S x\}] \).

It is worth noting here that in this last case the values of the nice function at the elements of \( \{y : y <_S x\} \) completely determine the parent of \( \pi_0(x) \), because
\( \{y : y <_S x\} \) is a support for the parent of \( x \). It is also worth noting that if \( x \) is a near-litter whose parent is an atom not in \( D_S \), \( \pi_0(x) \) will be an atom in the same clan as \( x \) and not belonging to \( \pi_0\bar{D}_S \) by conditions already stated.

**the extension property for nice maps:** We show that the union of a nice map \( \pi_0 \) on a strong support with a bijection \( \pi_1 \) on irregular atoms with which it coheres in a certain sense will extend to an allowable permutation.

The coherence condition is that if the domain of \( \pi_0 \) includes a near-litter \( N \) with parent an irregular atom \( x \), \( x \) will be in the domain of \( \pi_1 \) and the parent of \( \pi_0(N) \) will be \( \pi_1(x) \).
An anomaly of a near-litter \( N \) is defined as an element of the symmetric difference of \( N \) and the litter belonging to \([N]\).

We choose a locally small bijection \( \pi_2 \) extending \( \pi_1 \) and the restriction of \( \pi_0 \) to atoms, and defined on further atoms in such a way that \( \pi_2 \) is defined at every anomaly of an element of the domain or range of \( \pi_0 \) and the restriction of \( \pi_2 \) to any near-litter \( N \) in the domain of \( \pi_0 \) has range in \( \pi_0(N) \), and the restriction of \( \pi^{-1}_2 \) to any near-litter \( \pi_0(N) \) has range included in \( N \).

There is an allowable permutation \( \pi \) extending \( \pi_2 \). Obviously \( \pi \) extends \( \pi_1 \). If \( \pi \) does not extend \( \pi_0 \), there will be a \( <s \)-first element of \( D_S \) at which the two functions differ. This element would have to be a near-litter \( N \). The near-litter \( \pi(N) \) would have the same parent as the near-litter \( \pi_0(N) \), because either the parent is an irregular atom (handled correctly) or there is a support of the parent in the \( <s \)-earlier part of \( D_S \) which is mapped by \( \pi \) to values agreeing with those of \( \pi_0 \). Thus for \( \pi(N) \) to be unequal to \( \pi_0(N) \) there would have to be an atom mapped into \( \pi_0(N) \) from outside of \( N \) by \( \pi \) or mapped from \( N \) to the complement of \( \pi_0(N) \) by \( \pi \), but \( \pi \) has exceptions only in the domain of \( \pi_2 \), so elements of the litter in \([N]\) which are not in the domain of \( \pi_2 \) are mapped to elements of the litter in \([\pi_0(N)]\) which are not in the domain of \( \pi_2 \), and all anomalies of \( N \) and \( \pi_0(N) \) are in the domain of \( \pi_2 \) and have been coerced to map in a way which coheres with \( \pi_0 \).

It is also possible to use this extension property for nice functions cohering with maps on irregular atoms as the basic extension property (as I did in earlier drafts of this document) but the extension property as now stated seems conceptually simpler. The extension property for nice functions and maps on irregular atoms might also be referred to in the sequel simply as “the extension property”.

**Coding function:** If \( x \) is an object with strong support \( S \) and \( \pi \) is an allowable permutation, it is straightforward to verify that \( \pi(S) \) is a strong support for \( \pi(x) \). It is also straightforward to verify that \( \pi(S) = \pi'(S) \rightarrow \pi(x) = \pi'(x) \): \( \pi^{-1} \circ \pi' \) will fix each element of the first component of \( S \). Define \( K_{x,S}(\pi(S)) = \pi(x) \). The function \( K_{x,S} \) is called a coding function: its domain is an orbit in the strong supports under the allowable permutations.
orbits of strong supports: We claim that strong supports $S_1$ and $S_2$ belong to the same orbit under the allowable permutations iff $<_{S_1}$ and $<_{S_2}$ have the same order type $\ot(S_1)$ and

1. for each $\alpha < \ot(S_1)$ and each extended type index $A$, $S_1(\alpha) \in \clan[A]$ $\leftrightarrow$ $S_2(\alpha) \in \clan[A]$.
2. and for each $\alpha < \ot(S_1)$ and each extended type index $A$, $S_1(\alpha) \subseteq \clan[A]$ $\leftrightarrow$ $S_2(\alpha) \subseteq \clan[A]$.
3. for each $\alpha, \beta < \ot(S_1)$, $S_1(\alpha) \in S_1(\beta)$ $\leftrightarrow$ $S_2(\alpha) \in S_2(\beta)$.
4. and for each $\alpha$ with $S_1(\alpha) = N_1$ and $S_2(\alpha) = N_2$ near-litters included in $\clan[c]$ and $g$ a coding function, $g(((S_1)^{N_1}_{c}) = \pi_1(\Pi([N_1]))$ $\leftrightarrow$ $g(((S_2)^{N_2}_{c}) = \pi_1(\Pi([N_2]))$ in case the parent happens to be an atom. Note that there will be a unique $g$ for which this is true if there is any, unless $\pi_1(\Pi([N_i]))$ is an atom not in $S_i$ for $i = 1, 2$ as appropriate (in which case this will also be true for $i = 2, 1$ if $S_1$ and $S_2$ are in the same orbit).

We prove the claim.

The conditions indicated will hold if $S_2 = \pi(S_1)$ where $\pi$ is an allowable permutation. This is readily checked directly.

Now suppose that the conditions listed above hold for $S_1$ and $S_2$. We establish the existence of an allowable permutation $\pi$ such that $\pi(S_1) = S_2$ using the extension property for nice functions: the details follow.

For any strong support $S$, define $S[\alpha]$ as the restriction of $S$ to $\{t : t <_S S(\alpha)\}$ if $\alpha < \ot(S)$, and as $S$ if $\alpha \geq \ot(S)$.

Suppose that two permutations $S_1$ and $S_2$ satisfy the conditions given above. We prove that there is an allowable permutation sending $S_1$ to $S_2$. Suppose otherwise. Then there must be a least $\alpha$ such that there is no allowable permutation sending $S_1[\alpha]$ to $S_2[\alpha]$.

This $\alpha$ is not zero.

If $\alpha$ is a successor $\beta + 1$, we observe that there is an allowable permutation sending $S_1[\beta]$ to $S_2[\beta]$, which restricts to a nice map on $S_1[\beta]$ with domain $D_{S_1[\beta]}$ and range $D_{S_2[\beta]}$: all that remains is to show that
we can extend this nice map to $S_1(\beta)$. If $S_1(\beta)$ is an atom there is no problem with extending the nice map to send the atom $S_1(\beta)$ to $S_2(\beta)$, and it can then be extended to an allowable permutation. If $S_1(\beta)$ is a near-litter with parent an atom not in $S_1$, we extend our nice map to send $S_1(\beta)$ to $S_2(\beta)$, and the extended map is nice. Extending this nice map using the extension property gives an allowable permutation sending $S_1[(\beta + 1)]$ to $S_2[(\beta + 1)]$ contrary to assumption. If $S_1(\beta)$ is a near-litter $N$ with parent a set, this set is the image under some coding function $g$ of $(S_1)^c_N$ (where $N \subseteq \text{clan}[c]$) and there is an allowable permutation sending $S_1[\beta]$ to $S_2[\beta]$, which will send the parent of $S_1(\beta)$ to $g((S_2)^c_N)$ which is the parent of $S_2(\beta)$ because the allowable permutation sends $(S_1)^c_N$ to $(S_2)^c_N$. In each case, we thus construct a nice map which can be extended to an allowable permutation sending $S_1[(\beta + 1)]$ to $S_2[(\beta + 1)]$ contrary to assumption.

In the limit case we observe that each restriction to a proper initial segment of $D_{S_1}[\alpha]$ under its given order of the map sending each $S_1(\beta)$ with $\beta < \alpha$ to $S_2(\beta)$ is nice because it is the restriction of an allowable permutation by inductive hypothesis. But this implies that the entire map is nice, and so extendible to an allowable permutation sending $S_1[\alpha]$ to $S_2[\alpha]$ by the extension property.

The proof of the claim is complete.

**orbit signature:** By the considerations given in the previous paragraph, we can associate with each orbit of a strong support $S$ an object we will call an orbit signature, which is a pair $(\tau_S, \rho_S)$, where

1. $\tau_S$ is a function with domain $\text{ot}(\prec_S)$; $\rho_S$ is a function with domain a subset of $\text{ot}(S)$.
2. $\tau_S(\alpha) = (1, c)$ iff $S(\alpha) \in \text{clan}[c]$.
3. $\tau_S(\alpha) = (2, c)$ iff $S(\alpha) \subseteq \text{clan}[c]$.
4. $\rho_S(\alpha) = (1, \beta)$ if $S(\alpha) \subseteq S(\beta)$ and $\beta < \alpha$ will hold here; $\rho_S(\alpha) = (2, g)$, $g$ a coding function, iff $S(\alpha) = N$, a near-litter, and $g(S_N^c) = \pi_1(\Pi([N]))$ [if the parent is a set] or $g((S_N^c)^*) = \pi_1(\Pi([N]))$ [if the parent is an atom], where $N \subseteq \text{clan}[c]$; $\rho_S$ is undefined if neither of these situations holds.
The discussion of the previous point (along with the complexity property of abstract parent functions) indicates why an orbit signature exactly determines an orbit and vice versa.

**restricted coding functions:** We define a restricted coding function as a coding function $K_{x,S}$ where

1. if $x$ is an atom in $\text{clan}[c]$, $D_S$ has only atoms in $\text{clan}[c]$ as elements. Notice that the behaviour of $K_{x,S}$ will be defined by an equation $K_{x,S}(T) = T(\beta)$ for a fixed small $\beta$; these functions are in effect projection operators. There are no more than $\kappa \times |\lambda|$ such functions. Notice that we defined orbit signatures above to support use of such functions.

2. if $x$ is a set and $x$ belongs to $\text{parents}[c]$ for some clan index $c$, all elements of $S$ either have complexity $\leq \rho(c)$ or are hereditarily finite pure sets, and $S$ has an orbit signature in which all coding functions appearing are restricted. Any $x$ in a parent set has such a support by the complexity property of abstract parent functions.

We argue that there are $< \mu$ restricted coding functions. Clearly there are $< \mu$ restricted coding functions with atoms as values.

Choose any clan index $c$ and any set $x \in \text{parents}[c]$, and choose a support $S$ such that $K_{x,S}$ is a restricted coding function. For each element $y$ of $x$, choose a restricted coding function $K_{y,T}$ where the strong support $T = (D_T, \leq T)$ coheres with the strong support $S = (D_S, < S)$ in the sense that some $(D_S \cup D_T, <_{S \cup T})$ is an strong support where $<_{S \cup T}$ is a well-ordering extending $<_{S \cup T}$. We can do this because if we choose a support $T_0$ for $y$ we can extend $D_S \cup T_0$ to a strong support (choosing an order on it in the process), then restrict this support to elements of bounded complexity suitably for a restricted coding function, producing the desired strong support $T$. Let $Y$ be the set of restricted coding functions chosen in this way. $K_{x,y}(u)$ for any $u$ is the set of all values $g(T)$ such that $g \in Y$ and $T$ coheres with $S$: each element of $x$ is such a $g(T)$ by construction of $Y$ and each $g(T)$ is an image of a $g(T') \in x$ under an allowable permutation fixing all elements of $S$ (using the extension property for nice functions on the strong support with domain $D_S \cup D_T$) and so also an element of $x$. 
The function $K_{x,S}$ is completely determined by the small collection of coding functions appearing as components of the orbit signature of $S$, each either atom-valued or with output no more complex (in terms of $\rho$) than $x$ and with domain elements $T = (D_T, <_T)$ with the order type of $<_T$ strictly less than the order type of $<_S$, and the set of restricted coding functions used to generate elements of $x$, all of which have output of strictly smaller complexity than $x$: if there are $< \mu$ restricted coding functions with output complexity lower than that of $x$ (or hereditarily finite pure set output: such functions are constant functions), there are $< \mu$ sets of such functions because $\mu$ is strong limit. If we assume that we have already established that we have $< \mu$ restricted coding functions with set output of smaller complexity than that of $x$ or of the same complexity with shorter order components of its domain elements, we see that there are $< \mu$ possibilities for $K_{x,S}$. Since there are $\leq |\lambda|$ clan indices, there are $< \mu$ restricted coding functions.

**FM interpretation; definition of FM power set operation:** It follows from well-known results (FM techniques for constructing models of ZFA without choice) that the hereditarily symmetric sets (relative to any given II) make up a class model of ZFA which does not as a rule satisfy Choice. We define $P^*_X$ as the collection of hereditarily symmetric subsets of $X$ for each hereditarily symmetric set $X$. Our discussion will be in terms of the ambient ZFA with Choice except where we say so specifically.

We briefly describe the Frankel-Mostowski technique for constructing class models of ZFA, originally developed to prove the independence of Choice from ZFA. Our treatment is adapted from [9].

Any permutation $\pi$ 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 $\Gamma$ be a nonempty subset of the collection of subgroups of $G$ with the following properties:

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

2. The subset $\Gamma$ includes all subgroups $\bigcap C$ of $G$ where $C \subseteq \Gamma$ and $C$ is small [smallness being defined in terms of the parameter $\kappa$ already introduced above].
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 $a$, $\text{fix}_G(a) \in \Gamma$, where $\text{fix}_G(a)$ is the set of elements of $G$ which fix $a$.

A nonempty $\Gamma$ satisfying the first three conditions is what is called a $\kappa$-complete normal filter on $G$. We call a set $A$ $\Gamma$-symmetric iff the group of permutations in $G$ fixing $A$ belongs to $\Gamma$. The major theorem which we use but do not prove here is the assertion that the class of hereditarily $\Gamma$-symmetric objects (including all the atoms) is a class model of ZFA (usually not satisfying Choice). The assumption that the filter is $\kappa$-complete is not needed for the theorem ("finite" usually appears instead of "small"), but it does hold in our construction.

For us, $G$ is the group of allowable permutations and $\Gamma$ is the set of all subgroups $H$ of $G$ which contain, for some support set $S$, the group $G_S$ of all permutations which fix each element of $S$. The only condition on $\Gamma$ which takes any work to verify is the normality condition: if $H$ includes $G_S$, $\pi H \pi^{-1}$ includes $G_{\pi(S)}$.

**Theorem:** Any small subset of a hereditarily symmetric $X$ belongs to $P^\ast(X)$.

**Proof:** A support for this set is the union of supports of each of its elements.

**Theorem:** The symmetric power set of a clan, $P^\ast(\text{clan}[c])$, consists exactly of those subsets of $\text{clan}[c]$ which have small symmetric difference from a small or co-small union of litters [this picks out the same set no matter what choice set is used to define which near-litters are the litters].

**Proof of theorem:** A near-litter included in $\text{clan}[c]$ is symmetric with support its own singleton. If $\Lambda$ is a small collection of litters included in $\text{clan}[c]$, $\bigcup \Lambda$ and $\text{clan}[c] \setminus \bigcup \Lambda$ are both symmetric (and hereditarily symmetric) with support $\Lambda$. If $D$ is a small set of atoms in $\text{clan}[c]$, $D \Delta \bigcup \Lambda$ and $D \Delta (\text{clan}[c] \setminus \bigcup \Lambda)$ are both symmetric (and hereditarily symmetric) with support $\Lambda \cup D$.

Now suppose that $X \subseteq \text{clan}[c]$ has strong support $S = (D_S, <_S)$. We show first that if $L$ is a litter, it cannot be the case that both $L \cap X$ and $L \setminus X$ are large. Suppose otherwise. Choose atoms $a \in L \cap X$ and
$b \in L \setminus X$, neither belonging to the small domain of $S$. $(D_S \cup \{a, b\}, <'_S)$ will be a strong support set, for $<'_S$ obtained by adding $a, b$ at the end. The map on this set which exchanges $a$ and $b$ and fixes everything else is a nice map on the extended strong support, and so can be extended to an allowable permutation, which both fixes every element of the support $D_S$ of $X$ and moves the set $X$, which is absurd.

We show next that it cannot be the case that there is a large collection of litters which are neither included in nor disjoint from $X$. Suppose otherwise. Choose a litter $L$ which is neither included in nor disjoint from $X$, and no element of which belongs to $D_S$. Let $a \in L \setminus X$ and $b \in L \cap X$. Again, $(D_S \cup \{a, b\}, <'_S)$ is a strong support set (order extended as above), and the map interchanging $a, b$ and fixing everything in $D_S$ is a nice map on $S$, which will fix every element of the support $D_S$ of $X$ and move $X$, which is absurd.

This implies that any symmetric set $X$ has small symmetric difference from a union of litters: it cuts only a small collection of litters, and in each case one of the partitions of the litter is itself a small set.

Now suppose that $Y$ and $Z$ are large collections of litters, and $\bigcup Y$ is included in $X$ and $\bigcup Z$ is disjoint from $X$. Choose a litter $L \in Y$ and a litter $M \in Z$ such that neither $L, M$, nor any element of $L$ or $M$ belongs to $D_S$. $(D_S \cup \{a, b\}, <'_S)$ is a strong support set (with the order extended to $a, b$ as above), and the map which exchanges $a, b$ and fixes all other elements of $D_S \cup \{a, b\}$ is a nice map on this set, which fixes each element of $D_S$ and moves $X$, which is impossible. This shows that a union of litters which is symmetric must be the union of a small or co-small collection of litters.

This implies that any symmetric set $X$ has small symmetric difference from a union of litters: it cuts only a small collection of litters, and in each case one of the partitions of the litter is itself a small set.

Now suppose that $Y$ and $Z$ are large collections of litters, and $\bigcup Y$ is included in $X$ and $\bigcup Z$ is disjoint from $X$. Choose a litter $L \in Y$ and a litter $M \in Z$ such that neither $L, M$, nor any element of $L$ or $M$ belongs to $D_S$. $(D_S \cup \{a, b\}, <'_S)$ is a strong support set (with the order extended to $a, b$ as above), and the map which exchanges $a, b$ and fixes all other elements of $D_S \cup \{a, b\}$ is a nice map on this set, which fixes each element of $D_S$ and moves $X$, which is impossible. This shows that a union of litters which is symmetric must be the union of a small or co-small collection of litters.

This implies that any symmetric set $X$ has small symmetric difference from a union of litters: it cuts only a small collection of litters, and in each case one of the partitions of the litter is itself a small set.

Now suppose that $Y$ and $Z$ are large collections of litters, and $\bigcup Y$ is included in $X$ and $\bigcup Z$ is disjoint from $X$. Choose a litter $L \in Y$ and a litter $M \in Z$ such that neither $L, M$, nor any element of $L$ or $M$ belongs to $D_S$. $(D_S \cup \{a, b\}, <'_S)$ is a strong support set (with the order extended to $a, b$ as above), and the map which exchanges $a, b$ and fixes all other elements of $D_S \cup \{a, b\}$ is a nice map on this set, which fixes each element of $D_S$ and moves $X$, which is impossible. This shows that a union of litters which is symmetric must be the union of a small or co-small collection of litters.

This completes the argument that the symmetric subsets of $\text{clan}[c]$ are exactly the sets with small symmetric difference from small or co-small unions of litters.

**Theorem:** if $f$ is a symmetric bijection from $L \subseteq \text{clan}[A]$ to a subset $X$ of $\text{clan}[A]$ then $X$ has small symmetric difference from $L$.

**Proof:** We prove the stronger claim that $f$ is the identity map off a small set. Suppose otherwise (that $f$ has a large collection of non-fixed points). Then we can fix a strong support $S$ for $f$ and choose $a$ and $b$ such that $a, b, f(a), f(b)$ are distinct (we get these by showing that there is
more than one orbit among the non-fixed points which do not meet $S$, orbits being small). We can then extend the domain of $S$ to include $a, b, f(a), f(b)$ and extend the nice map which swaps $a, b$ and fixes all elements of $S$ and fixes $f(a)$ and $f(b)$.

Certainly if $L$ and $X$ have small symmetric difference we can construct a symmetric bijection from one to the other.

Thus the cardinalities of the litters in a clan are each $\kappa$-amorphous and all are distinct.

**Theorem (double power set lemma):** For any clan $[c]$, there is a hereditarily symmetric bijection from $\mathcal{P}_s(\text{parents}[c])$ into $\mathcal{P}_s^2(\text{clan}[c])$.

**Proof of theorem:** The bijection $\pi_c$ sending each element $\pi_{\Pi([N])}$ of $\text{parents}[c]$ to $[N] \in \mathcal{P}_s^2(\text{clan}[c])$ is hereditarily symmetric. The bijection sending each hereditarily symmetric $X \subseteq \text{parents}[c]$ to $\pi_c^{-1}X$ is hereditarily symmetric.
11 Definition of and construction of typed parent functions

Definition (typed parent function of order $\alpha$): Let $\alpha \leq \lambda$ be an ordinal. A typed parent function is an abstract parent function $\Pi$ satisfying certain additional conditions.

**extended type indices index the clans:** For each clan $C$, $\text{index}[C]$ is an extended type index $A$ with all elements of $A$ less than $\alpha$. The clan $\text{clan}[A]$ exists for each such extended type index.

**parent set of the empty set indexed clan:** The parent set of $\text{clan}[\emptyset]$ is a collection of irregular atoms of cardinality $\mu$.

**parent sets of other clans:** The parent set of $\text{clan}[A]$ for $A$ nonempty is

$$\text{clan}[A_1] \cup \bigcup_{B \ll A} P_*^{[|B| - |A| + 1]}(\text{clan}[B]).$$

Note that in this scheme the complexity of an element of $P_*^{[|B| - |A| + 1]}(\text{clan}[B])$ is $\min(A)$, that is, the complexity of an element of $P^n(\text{clan}[B])$, for $n$ positive, is the minimum element of $B_{n-1}$ (this is also true for $n = 1$ by the basic stipulation about complexity of sets of atoms). This enforces the rule that the elements of a set of a given complexity belonging to a set $\text{parents}[c]$ are of lower complexity (a set of atoms is of the same complexity as its elements, but sets of atoms do not belong to parent sets).

**Construction (build typed parent functions of each order $\leq \lambda$):** Our inductive hypothesis for construction of a typed parent function of order $\alpha$ is that typed parent functions $\Pi_\beta$ of each order $\beta < \alpha$ have been constructed, with $\Pi_\beta \subseteq \Pi_\gamma$ holding whenever $\beta < \gamma < \alpha$.

Further inductive hypotheses (which we will see are enforced quite directly at each step by the most superficial description of the construction) are that for any extended type index $B$, and integer $n$ with $0 < n < |B| + 1$, the structure of $P^n_*(\text{clan}[B])$ is determined up to the action of a bijection between atoms by $B \setminus B_{n-1}$, and any element of $P^n_*(\text{clan}[B])$ has a strong support containing only elements of clans and subsets of clans with index equal to or downward extending $B_{n-1}$. 
The structure of $\mathcal{P}_*(\text{clan}[B])$ is the same for any $B$, the symmetric power set of a clan being simply the collection of all subsets of the clan with small symmetric difference from a small or co-small union of litters included in the clan [proved above].

A typed parent function of order 0 is easily constructed. The only extended type index used is $\emptyset$ and a single clan of size $\mu$ is provided with parents taken from a set of irregular atoms of size $\mu$. The extension property holds because any permutation of the irregular atoms, extended to an allowable permutation without exceptions, composed with a permutation of the regular atoms which moves only a small number of regular atoms in each litter, gives an allowable permutation, and all the maps required by the extension property can be constructed in this way. The complexity property holds trivially because no sets belong to the only parent set.

If $\alpha$ is limit, the inductive hypothesis stated above implies immediately that we have a typed parent function of order $\alpha$, defined as the union of the typed parent functions of each order $< \alpha$ already constructed.

We argue that a typed parent function of order $\alpha$ can be extended to a typed parent function of order $\alpha + 1$, establishing the successor case of our induction.

The construction proceeds as follows. Let $\Pi_\alpha$ be the typed parent function of order $\alpha$ already constructed. Let $\sigma$ be a bijection from the collection of atoms in the transitive closure of $\Pi_\alpha$ to a collection of atoms disjoint from this set. Extend $\sigma$ to any other atoms by letting the extended map fix all such atoms, then extend $\sigma$ to sets by the rule $\sigma(A) = \sigma^*A$. Define $\text{fix}(a, A)$ as $(a, \{\alpha\} \cup A)$ for any ordered pair $(a, A)$ in the range of $\sigma(\Pi_\alpha)$. Define $\Pi_{\alpha+1}^1$ as $\Pi_\alpha \cup (\text{fix} \circ (\sigma(\Pi_\alpha)))$.

$\Pi_{\alpha+1}^1$ is an abstract parent function. It is not a typed parent function until we make a further modification, though the indices of clans are exactly the desired extended type indices. All the conditions to be an abstract parent function follow directly from the fact that the set $\Pi_{\alpha+1}^1$ is basically made up of two isomorphic copies of $\Pi_\alpha$.

It fails to be a typed parent function most obviously because $\text{parents}[^{\{\alpha\}}]$ is the set of irregular atoms $\sigma^*\text{parents}[^{\emptyset}]$ instead of being as expected $\text{clan}[\emptyset] \cup \bigcup_{B <^{\{\alpha\}} B} \mathcal{P}_*[^{B-\{\alpha\}+1}](\text{clan}[B])$. Notice that $\bigcup_{B <^{\{\alpha\}} B} \mathcal{P}_*[^{B-\{\alpha\}+1}](\text{clan}[B])$
is the image under $\sigma$ of the set $\bigcup_{\alpha} \mathcal{P}^{[\alpha]+1}_* \clan[A]$, a set entirely definable in terms of $\Pi_\alpha$.

We first need to show that $\clan[\emptyset] \cup \bigcup_{B < \{\alpha\}} \mathcal{P}^{[\alpha]+1}_* \clan[B]$ is a set of size $\mu$. The included clan is of course of size $\mu$. It suffices to show that each $\bigcup_{\alpha} \mathcal{P}^{[\alpha]+1}_* \clan[A]$ for $A$ strictly bounded by $\alpha$ is of size $\mu$. We observe that applications of $< \mu$ restricted coding functions to $\mu$ possible indexed strong supports generate no more than $\mu$ objects (there are obviously at least $\mu$ objects in the set under consideration).

We then construct a bijection $h$ from $\sigma^{-1}\text{parents}[\emptyset] = \text{parents}[\{\alpha\}]$ to

$$\clan[\emptyset] \cup \bigcup_{B < \{\alpha\}} \mathcal{P}^{[\alpha]+1}_* \clan[B].$$

We start with a minimal length well-ordering of each of these two sets. At each stage, we match the first so-far-unused element of $\text{parents}[\{\alpha\}]$ with the first so-far-unused element of the set $\clan[\emptyset] \cup \bigcup_{B < \{\alpha\}} \mathcal{P}^{[\alpha]+1}_* \clan[B]$ which has a strong support containing no near-litter with a parent in $\text{parents}[\{\alpha\}]$ which has not yet been matched: we designate one such support for each element of $\clan[\emptyset] \cup \bigcup_{B < \{\alpha\}} \mathcal{P}^{[\alpha]+1}_* \clan[B]$ in the course of the construction of $h$, so that each $h(x)$ will have a strong support with the property that any element of $\text{parents}[\alpha]$ which is the parent of an element of its designated strong support occurs earlier than $x$ in the given order. Because the cofinality of $\mu$ is greater than $\kappa$, we can be certain that every element of the intended range of $h$ will actually be a value of $h$.

We then modify $\Pi_{\alpha+1}$ to obtain $\Pi_{\alpha+1}$. $\Pi_{\alpha+1}([N])$ is equal to $\Pi_{\alpha+1}([N])$ if $N$ is not included in $\clan[\{\alpha\}]$, and is equal to $(h(\Pi_{\alpha+1}([N])), \{\alpha\})$ if $N$ is included in $\clan[\{\alpha\}]$. What we have done is exactly to replace the undesired $\text{parents}[\{\alpha\}]$ with something which looks like (and which we hope will remain) the desired $\text{parents}[\{\alpha\}]$.

Each element of a parent set has a $\Pi_{\alpha+1}$-extended strong support: take a $\Pi_{\alpha+1}$-extended support and for each element $x$ of $\text{parents}[\{\alpha\}]$ which is parent of a near-litter in this support add an extended support of $h(x)$, and iterate as necessary: because of the way $h$ is constructed this process is well-founded. From this the extension property for $\Pi_{\alpha+1}$ follows.
We need to argue for the $\Pi^{\alpha+1}$-extension property. Suppose that $\pi_0$ is a $\Pi^{\alpha+1}$-locally small bijection. It is then also a $\Pi^{\alpha+1}_1$-locally small bijection so it admits an extension to an allowable permutation, but this will not necessarily be $\Pi^{\alpha+1}_1$-allowable because its action on $\text{parents}[\alpha]$ is wrong (in fact arbitrary). This can be repaired: we can compute the intended value of $\pi$ at each $x$ in $\text{parents}[\alpha]$ in the order used in the construction of $h$, by computing the value of $\pi$ at $h(x)$, then extending $\pi_0$ to have the inverse image under $h$ of $\pi(h(x))$ as its value. We ensure that we get a determinate value by using the procedure described above for computing extensions using a global support order, ensuring that the order on near-litters with irregular parents agrees with the order used in the construction of $h$. Each value assigned at an element of $\text{parents}([\alpha])$ depends only on values assigned to earlier elements of this set in the order, so this procedure will after enough iterations give an assignment of values to all elements of $\text{parents}([\alpha])$ which can be seen to induce a $\Pi^{\alpha+1}_1$-allowable permutation.

It remains to be verified that what $\Pi^{\alpha+1}_1$ identified as

$$\text{clan}([\emptyset]) \cup \bigcup_{B << [\alpha]} \mathcal{P}^{[\alpha]+1}(\text{clan}[B])$$

is the same set which $\Pi^{\alpha+1}$ identifies as such. This is not unbelievable but also not obvious. More generally, to confirm that $\Pi^{\alpha+1}$ is a typed parent function we need to verify that $\Pi^{\alpha+1}_1$ and $\Pi^{\alpha+1}$ see the iterated power sets of clans included in parent sets as having the same elements (their notions of symmetry need to agree on these sets).

One direction of symmetry arguments is easy. Any $\Pi^{\alpha+1}_1$-allowable permutation is $\Pi^{\alpha+1}_1$-allowable, because the action of permutations on $\text{parents}([\alpha])$ by the latter class of permutations is completely free. So any $\Pi^{\alpha+1}_1$-symmetric set is actually $\Pi^{\alpha+1}_1$-symmetric. What we need to show is that any $\Pi^{\alpha+1}$-symmetric subset of a $\mathcal{P}^{[\alpha]+1}(\text{clan}[B])$ with $\max[B] = \alpha$ and $n \leq |B|$ is also $\Pi^{\alpha+1}_1$-symmetric: from this it follows that the new parent function is a typed parent function because it agrees with the old one on the identities of relevant iterated symmetric power sets of clans.

The required result follows by considering coding functions. Since all $\Pi^{\alpha+1}_1$-allowable permutations are also $\Pi^{\alpha+1}_1$-allowable, it follows that if
\[ \pi \text{ is a } \Pi_{\alpha+1} \text{-allowable permutation and } K_{a,S} \text{ is a } \Pi^1_{\alpha+1} \text{-coding function,} \]

the identity \( \pi(K_{a,S}(T)) = K_{a,S}(\pi(T)) \) continues to hold.

Suppose that \( x \) is a subset of \( \mathcal{P}^n(\text{clan}[B]) \) [as defined in terms of \( \Pi^1_{\alpha+1} \)]

with a support \( S \) relative to all \( \Pi_{\alpha+1} \)-allowable permutations. Suppose

that for all \( m < n \) the desired result has already been established (for \( n = 1 \) it is evident, by the description already given of the symmetric subsets of clans). We claim that if we close up \( S \) to a strong support and also so that it contains a near-litter with parent \( x \) in the domain of \( h \) [thus making it a strong support set relative to \( \Pi_{\alpha+1} \)], the resulting \( S^* \) is a support for \( x \) relative to \( \Pi^1_{\alpha+1} \)-allowable permutations. The reason is that every element of \( x \) can be expressed as a coding function applied to a support cohering with \( S^* \) and for complexity bound reasons not including any litters included in \( \text{clan}[\{\alpha\}] \), and will be sent by any \( \Pi^1_{\alpha+1} \)-allowable permutation fixing each element of \( S^* \) to something which by the extension property for \( \Pi_{\alpha+1} \) already established can be seen to be the image of the same element of \( x \) under a \( \Pi_{\alpha+1} \)-allowable permutation fixing each element of \( S \) and so by hypothesis fixing \( x \), and so an element of \( x \) (and similarly for the inverse of the permutation), so \( x \) is fixed as desired.

We expand on the last point. The claim is that if \( \pi \) is a \( \Pi^1_{\alpha+1} \)-allowable permutation fixing all elements of the domain of a \( \Pi_{\alpha+1} \)-strong support \( S \), and \( x \) is any object with an strong support \( T \) cohering with \( S \) and including no near-litter included in \( \text{clan}[\{\alpha\}] \), there is a \( \Pi_{\alpha+1} \)-allowable permutation \( \pi' \) such that \( \pi(x) = \pi(x') \). Let \( U \) be the componentwise union of \( S \) and \( T \) (with the union of the order components extended to a well-ordering arbitrarily). Express \( x \) as \( K_{x,S}(U) \) (this being a \( \Pi^1_{\alpha+1} \)-coding function). \( \pi(x) = K_{x,S}(\pi(U)) \). Now observe that the restriction of \( K_{a,S} \) to the domain of \( U \) is easily seen to be \( \Pi_{\alpha+1} \)-nice (here it is important that all near-litters included in \( \text{clan}[\{\alpha\}] \) which belong to \( U \) are fixed by this map). Thus it can be extended to a \( \Pi_{\alpha+1} \) allowable permutation \( \pi' \) which will fix all elements of \( S \) and \( \pi'(x) = \pi'(K_{a,S}(S)) = K_{a,S}(\pi'(S)) = K_{a,S}(\pi(S)) = \pi(x) . \)
12 Definition of the tangled web and verification of its properties

We now let $\Pi$ be a typed parent function of order $\lambda$, and argue that there is a tangled web in the FM interpretation associated with $\Pi$.

The definition of the tangled web is $\tau(A) = |P^2(\text{clan}[A])|$ for each nonempty extended type index $A$, the power sets and cardinalities being those of the FM interpretation.

Define $\exp(|X|)$ as $|P(X)|$.

We need to verify that $\exp(\tau(A)) = \tau(A_1)$ if $|A| \geq 2$ This is equivalent to showing that $|P^3(\text{clan}[A])| = |P^2(\text{clan}[A_1])|$. We have from the formula for parent sets that $|P^2(\text{clan}[A_1])| \geq |P(\text{parents}[A_1])| \geq |P(P(\text{parents}[A]) \cup \text{clan}[A])| = |P^3(\text{clan}[A])|$. On the other hand $|P^3(\text{clan}[A])| \geq |P^2(\text{parents}[A])| \geq |P^2(\text{clan}[A_1])|$. This verifies the naturality property of tangled webs.

The elementarity property of tangled webs falls directly out of the construction. We need to show that the natural model of an initial segment of type theory with $n$ types whose base type has cardinality $\tau(A)$ depends only on $A \setminus A_n$, set of the smallest $n$ elements of $A$. This reduces to consideration of models of type theory with $n$ types whose base type is $P^2(\text{clan}[A])$, that is, natural models of type theory with $n + 2$ types whose base type is $\text{clan}[A]$ and whose top type is $P^{n+1}(\text{clan}[A])$, where $A$ has at least $n$ elements [the power set operation here being that of the FM interpretation]. And all such models are images under the action of bijections on atoms [in the ground model of ZFA] of the natural model of type theory whose base type is $\text{clan}[A \setminus A_n]$ and whose top type is $P^{n+1}(\text{clan}[A \setminus A_n])$, by the construction, and so of course have the same first-order theory because they are isomorphic models of type theory from the standpoint of the ground model of ZFA.
13 Conclusions and questions

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

By choosing the parameter $\lambda$ 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 $\omega$- and $\alpha$-models of NFU to get $\omega$- and $\alpha$-models of NF (details given above). One can show the consistency of NF + Rosser’s Axiom of Counting (see [13]), Henson’s Axiom of Cantorian Sets (see [4]), or the author’s axioms of Small and Large Ordinals (see [6], [7], [15]) 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 $\kappa$ 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 $\omega_1$ for $\kappa$ 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 $\kappa$ 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 $\kappa$-complete in the sense of containing every subset of their domains of size $\kappa$; 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 $\kappa$-complete, so combinatorial consequences of $\kappa$-completeness will hold in the model of NF (which could further be made a $\kappa$-model by making $\lambda$ 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 \( \omega \)-model of TNT (the theory of negative types, that is TST with all integers as types, proposed by Hao Wang ([18])) is settled: an \( \omega \)-model of NF yields an \( \omega \)-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 (at least in ZFA) 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 [2], 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 [2], 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?
14 References and Index

References


Index

α-models, 20
κ, parameter introduced in the main construction, 25
λ, parameter introduced, 10
λ, parameter introduced in the main construction, 25
μ, parameter introduced in the main construction, 25
ω-models, 20

abstract parent function, definition begins, 26
abstract parent function, definition completed, 30
allowable permutation, 28
ambiguity, axiom scheme of, 13
anomaly of a near litter, 34
atom, irregular, 27
atoms, regular, 26
axiom of weak extensionality, 15
axiom scheme of ambiguity, 13
axioms of NF, 11
axioms of TST, 8, 10

Choice false in NF, 13
Choice, axiom of, 9
clan, 26
clan index, 27
coding function, defined, 34
coding function, restricted, 37
complexity condition on abstract parent sets, 29
complexity of an object, 29
comprehension in NF, 11
comprehension in TST, 8
comprehension in TSTλ, 10
comprehension, stratified, axiom of, 12
consistency of NFU, 16
double power set lemma, 41
downward extension relation on extended type indices, defined, 25
elementarity property of tangled webs, 23
Erdős-Rado theorem, 20
exception (of an allowable permutation), 30
extended type index, 23
extended type index, defined, 25
extension property for nice functions/maps, 33
extensionality in NF, 11
extensionality in TST, 8
extensionality in TSTλ, 10
Frankel-Mostowski construction, 38
hereditarily symmetric, 28

index, clan, 27
index, extended type, 25
index, type, 25
Infinity a theorem of NF, 13
Infinity, axiom of, 9

large set, defined, 25
litter, 26
local cardinal, 26
locally small bijection, 30
naive set theory, 8
natural models of TST, 9
natural models of TST in TST, 22
natural models of TSTU, 15
naturality property of tangled webs, 23
near-litter, 27
New Foundations, 7
NF, 7
NF, definition of, 11
NFU, 15
NFU is consistent, 16
nice function, 33
nice map, 33
normal filter, 39
orbit signature, 36
orbits of strong supports, 35
parameter $\lambda$ introduced, 10
parent (of an atom, litter, near-litter), 27
parent function, abstract, definition begins, 26
parent function, abstract, definition completed, 30
parent functions, typed, 42
parent set (of a clan), 27
permutation, allowable, 28

regular atoms, 26
restricted coding function, 37

set abstract notation, 8, 11
simple theory of types, 8
small set, defined, 25
sorts in TST, 8
sorts in TST, 10, 18
stratification of a formula, 12

stratified comprehension, axiom of, 12
stratified formula, 12
strong support, 28
support of an object, 28
support set, 28
symmetric, 28
symmetric, hereditarily, 28
tangled type theory defined, 18
tangled web of cardinals, 23
tangled web, specified in the main construction, 47

TNT, 10
TST, 8
TST, 10
TST, 9
TSTU, 15
TTT, 18
type index, 10
type index, defined, 25
type index, extended, 23
type-raising, syntactical, 11
typed parent functions, 42
types in TST, 8
types in TST, 10, 18

weak extensionality, axiom of, 15