New Foundations is consistent: a production version

M. Randall Holmes

May 18, 2022

Contents

1 Remarks on this version

This document is probably my best overall version so far. The immediate occasion for its preparation is to serve students attempting to verify the proof in Lean. As we have discussed, the formal verification should at least initially avoid metamathematics, so it is the fact that the structure defined in section 3 is a model of TTT which should be verified, and further, a finite axiomatization (mod type indexing) of TST and thus TTT should be verified in the model in lieu of the usual statement of the axiom of comprehension of TTT.

2 Development of relevant theories

2.1 The simple theory of types TST and TSTU

We introduce a theory which we call the simple typed theory of sets or TST, a name favored by the school of Belgian logicians who studied NF (*théorie simple de types*). This is not the same as the simple type theory of Ramsey and it is most certainly not Russell's type theory (see historical remarks below).

TST is a first order multi-sorted theory with sorts (types) indexed by the nonnegative integers. The primitive predicates of TST are equality and membership.

The type of a variable x is written $\texttt{type}(`x')$: this will be a nonnegative integer. A countably infinite supply of variables of each type is supposed. An atomic equality sentence '$x = y$' is well-formed iff $\texttt{type}(`x') = \texttt{type}(`y')$. An atomic membership sentence '$x \in y$' is well-formed iff $\texttt{type}`(x') + 1 = \texttt{type}(`y')$.

The axioms of TST are extensionality axioms and comprehension axioms.

The extensionality axioms are all the well-formed assertions of the shape $(\forall xy : x = y \leftrightarrow (\forall z : z \in x \leftrightarrow z \in y))$. For this to be well typed, the variables x and y must be of the same type, one type higher than the type of z.

The comprehension axioms are all the well-formed assertions of the shape $(\exists A : (\forall x : x \in A \leftrightarrow \phi))$, where ϕ is any formula in which A does not occur free.

The witness to $(\exists A : (\forall x : x \in A \leftrightarrow \phi))$ is unique by extensionality, and we introduce the notation $\{x : \phi\}$ for this object. Of course, $\{x : \phi\}$ is to be assigned type one higher than that of x; in general, term constructions will have types as variables do.

The modification which gives TSTU (the simple type theory of sets with urelements) replaces the extensionality axioms with the formulas of the shape

$$(\forall xyw : w \in x \rightarrow (x = y \leftrightarrow (\forall z : z \in x \leftrightarrow z \in y))),$$

allowing many objects with no elements (called atoms or urelements) in each positive type. A technically useful refinement adds a constant $\emptyset^i$ of each positive type i with no elements: we can then address the problem that $\{x^i : \phi\}$ is not uniquely defined when ϕ is uniformly false by defining $\{x^i : \phi\}$ as $\emptyset^{i+1}$ in this case.

2.1.1 Typical ambiguity

TST exhibits a symmetry which is important in the sequel.

Provide a bijection $(x \mapsto x^+)$ from variables to variables of positive type satisfying $\mathtt{type}(x^+) = \mathtt{type}(x) + 1$.

If ϕ is a formula, define ϕ^+ as the result of replacing every variable x (free and bound) in ϕ with x^+. It should be evident that if ϕ is well-formed, so is ϕ^+, and that if ϕ is a theorem, so is ϕ^+ (the converse is not the case). Further, if we define a mathematical object as a set abstract $\{x : \phi\}$ we have an analogous object $\{x^+ : \phi^+\}$ of the next higher type (this process can be iterated).

The axiom scheme asserting $\phi \leftrightarrow \phi^+$ for each closed formula ϕ is called the Ambiguity Scheme. Notice that this is a stronger assertion than is warranted by the symmetry of proofs described above.

2.1.2 Historical remarks

TST is not the type theory of the *Principia Mathematica* of Russell and Whitehead, though a description of TST is a common careless description of Russell's theory of types.

Russell described something like TST informally in his 1904 *Principles of Mathematics*. The obstruction to giving such an account in *Principia Mathematica* was that Russell and Whitehead did not know how to describe ordered pairs as sets. As a result, the system of *Principia Mathematica* has an elaborate system of complex types inhabited by n-ary relations with arguments of specified previously defined types, further complicated by predicativity restrictions (which are cancelled by an axiom of reducibility). The simple theory of types of Ramsey eliminates the predicativity restrictions and the axiom of reducibility, but is still a theory with complex types inhabited by n-ary relations.

Russell noticed a phenomenon like the typical ambiguity of TST in the more complex system of *Principia Mathematica*, which he refers to as "systematic ambiguity".

In 1914, Norbert Wiener gave a definition of the ordered pair as a set (not the one now in use) and seems to have recognized that the type theory of *Principia Mathematica* could be simplified to something like TST, but he did not give a formal description. The theory we call TST was apparently first described by Tarski in 1930.

It is worth observing that the axioms of TST look exactly like those of "naive set theory", the restriction preventing paradox being embodied in the restriction of the language by the type system. For example, the Russell paradox is averted because one cannot have $\{x : x \notin x\}$ because $x \in x$ (and so its negation $\neg x \in x$) cannot be a well-formed formula.

It was shown around 1950 that Zermelo set theory proves the consistency of TST with the axiom of infinity; TST + Infinity has the same consistency strength as Zermelo set theory with separation restricted to bounded formulas.

2.2 Some mathematics in TST; the theories TST_n and their natural models

We briefly discuss some mathematics in TST.

We indicate how to define the natural numbers. We use the definition of Frege (n is the set of all sets with n elements). 0 is $\{\emptyset\}$ (notice that we get a natural number 0 in each type $i+2$; we will be deliberately ambiguous in this discussion, but we are aware that anything we define is actually not unique, but reduplicated in each type above the lowest one in which it can be defined). For any set A at all we define $\sigma(A)$ as $\{a \cup \{x\} : a \in A \wedge x \notin a\}$. This is definable for any A of type $i+2$ (a being of type $i+1$ and x of type i). Define 1 as $\sigma(0)$, 2 as $\sigma(1)$, 3 as $\sigma(2)$, and so forth. Clearly we have successfully defined 3 as the set of all sets with three elements, without circularity. But further, we can define $\mathbb{N}$ as $\{n : (\forall I : 0 \in I \wedge (\forall x \in I : \sigma(x) \in I) \rightarrow n \in I\}$, that is, as the intersection of all inductive sets. $\mathbb{N}$ is again a typically ambiguous notation: there is an object defined in this way in each type $i+3$.

The collection of all finite sets can be defined as $\bigcup \mathbb{N}$. The axiom of infinity can be stated as $V \notin \bigcup \mathbb{N}$ (where $V = \{x : x = x\}$ is the typically ambiguous symbol for the type $i+1$ set of all type i objects). It is straightforward to show that the natural numbers in each type of a model of TST with Infinity are isomorphic in a way representable in the theory.

Ordered pairs can be defined following Kuratowski and a quite standard theory of functions and relations can be developed. Cardinal and ordinal numbers can be defined as Frege or Russell would have defined them, as isomorphism classes of sets under equinumerousness and isomorphism classes of well-orderings under similarity.

The Kuratowski pair $(x, y) = \{\{x\}, \{x, y\}\}$ is of course two types higher than its projections, which must be of the same type. There is an alternative definition (due to Quine) of an ordered pair $\langle x, y\rangle$ in TST + Infinity which is of the same type as its projections x, y. This is a considerable technical convenience but we will not need to define it here. Note for example that if we use the Kuratowski pair the cartesian product $A \times B$ is two types higher than A, B, so we cannot define $|A| \cdot |B|$ as $|A \times B|$ if we want multiplication of cardinals to be a sensible operation. Let ι be the singleton operation and define $T(|A|)$ as $|\iota``A|$ (this is a very useful operation sending cardinals of a given type to cardinals in the next higher type which seem intuitively to be the same). The definition of cardinal multiplication if we use the Kuratowski pair is then $|A| \cdot |B| = T^{-2}(|A \times B|)$. If we use the Quine pair this becomes the usual definition $|A| \cdot |B| = |A \times B|$. Use of the Quine pair simplifies matters in this case, but it should be noted that the T operation remains quite important (for example it provides the internally representable isomorphism between the systems of natural numbers in each sufficiently high type).

Note that the form of Cantor's Theorem in TST is not $|A| < |\mathcal{P}(A)|$, which would be ill-typed, but $|\iota``A| < |\mathcal{P}(A)|$: a set has fewer unit subsets than subsets. The exponential map $\exp(|A|) = 2^{|A|}$ is not defined as $|\mathcal{P}(A)||$, which would be one type too high, but as $T^{-1}(|\mathcal{P}(A))$, the cardinality of a set X such that

$|\iota``X| = |\mathcal{P}(A)|$; notice that this is partial. For example $2^{|V|}$ is not defined (where $V = \{x : x = x\}$, an entire type), because there is no $X$ with $|\iota``X| = |\mathcal{P}(V)|$, because $|\iota``V| < |\mathcal{P}(V)| \leq |V|$, and of course there is no set larger than V in its type.

For each natural number n, the theory TST_n is defined as the subtheory of TST with vocabulary restricted to use variables only of types less than n (TST with n types). In ordinary set theory TST and each theory TST_n have natural models, in which type 0 is implemented as a set X and each type i in use is implemented as $\mathcal{P}^i(X)$. It should be clear that each TST_n has natural models in bounded Zermelo set theory, and TST has natural models in a modestly stronger fragment of ZFC.

Further, each TST_n has natural models in TST itself, though some care must be exercised in defining them. Let X be a set. Implement type i for each $i < n$ as $\iota^{(n-1)-i}``\mathcal{P}^i(X)$. If X is in type j, each of the types of this interpretation of TST_n is a set in the same type $j + n - 1$. For any relation R, define R^ι as $\{(\{x\}, \{y\}) : xRy\}$. The membership relation of type $i - 1$ in type i in the interpretation described is the restriction of $\subseteq^{\iota^{(n-1)-i}}$ to the product of the sets implementing type $i - 1$ and type i.

Notice then that we can define truth for formulas in these natural models of TST_n for each n in TST, though not in a uniform way which would allow us to define truth for formulas in TST in TST.

Further, both in ordinary set theory and in TST, observe that truth of sentences in natural models of TST_n is completely determined by the cardinality of the set used as type 0. since two natural models of TST or TST_n with base types implemented by sets of the same cardinality are clearly isomorphic.

2.3 New Foundations and NFU

In 1937, Willard van Orman Quine proposed a set theory motivated by the typical ambiguity of TST described above. The paper in which he did this was titled "New foundations for mathematical logic", and the set theory it introduces is called "New Foundations" or NF, after the title of the paper.

Quine's observation is that since any theorem ϕ of TST is accompanied by theorems $\phi^+, \phi^{++}, \phi^{+++}, \ldots$ and every defined object $\{x : \phi\}$ is accompanied by $\{x^+ : \phi^+\}, \{x^{++} : \phi^{++}\}, \{x^{+++} : \phi^{+++}\}$, so the picture of what we can prove and construct in TST looks rather like a hall of mirrors, we might reasonably suppose that the types are all the same.

The concrete implementation follows. NF is the first order unsorted theory with equality and membership as primitive with an axiom of extensionality $(\forall xy : x = y \leftrightarrow (\forall z : z \in x \leftrightarrow z \in y))$ and an axiom of comprehension $(\exists A : (\forall x : x \in A \leftrightarrow \phi))$ for each formula ϕ in which A is not free which can be obtained from a formula of TST by dropping all distinctions of type. We give a precise formalization of this idea: provide a bijective map $(x \mapsto x^*)$ from the countable supply of variables (of all types) of TST onto the countable supply of variables of the language of NF. Where ϕ is a formula of the language of TST, let ϕ^* be the formula obtained by replacing every veriable x, free and bound, in ϕ with x^*. For each formula ϕ of the language of TST in which A is not free in ϕ^*, an axiom of comprehension of NF asserts $(\exists A : (\forall x : x \in A \leftrightarrow \phi^*))$.

In the original paper, this is expressed in a way which avoids explicit dependence on the language of another theory. Let ϕ be a formula of the language of NF. A function σ is a stratification of ϕ if it is a (possibly partial) map from variables to non-negative integers such that for each atomic subformula '$x = y$' of ϕ we have $\sigma(\text{'}x\text{'}) = \sigma(\text{'}y\text{'})$ and for each atomic subformula '$x \in y$' of ϕ we have $\sigma(\text{'}x\text{'})+1 = \sigma(\text{'}y\text{'})$. A formula ϕ is said to be stratified iff there is a stratification of ϕ. Then for each stratified formula ϕ of the language of NF we have an axiom $(\exists A : (\forall x : x \in A \leftrightarrow \phi))$. The stratified formulas are exactly the formulas ϕ^* up to renaming of variables.

NF has been dismissed as a "syntactical trick" because of the way it is defined. It might go some way toward dispelling this impression to note that the stratified comprehension scheme is equivalent to a finite collection of its instances, so the theory can be presented in a way which makes no reference to types at all. This is a result of Hailperin, refined by others. One obtains a finite axiomatization of NF by analogy with the method of finitely axiomatizing von Neumann-Gödel-Bernays predicate class theory. It should further be noted that the first thing one does with the finite axiomatization is prove stratified comprehension as a meta-theorem, in practice, but it remains significant that the theory can be axiomatized with no reference to types at all.

For each stratified formula ϕ, there is a unique witness to

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

(uniqueness follows by extensionality) whch we denote by $\{x : \phi\}$.

Jensen in 1969 proposed the theory NFU which replaces the extensionality axiom of NF with

$$(\forall xyw : w \in x \rightarrow (x = y \leftrightarrow (\forall z : z \in x \leftrightarrow z \in y))),$$

allowing many atoms or urelements. One can reasonably add an elementless constant $\emptyset$, and define $\{x : \phi\}$ as $\emptyset$ when ϕ is false for all x.

Jensen showed that NFU is consistent and moreoever NFU + Infinity + Choice is consistent. We will give an argument similar in spirit though not the same in detail for the consistency of NFU in the next section.

An important theorem of Specker (1962) is that NF is consistent if and only if TST + the Ambiguity Scheme is consistent. His method of proof adapts to show that NFU is consistent if and only if TSTU + the Ambiguity Scheme is consistent. Jensen used this fact in his proof of the consistency of NFU. We indicate a proof of Specker's result using concepts from this paper below.

In 1954, Specker had shown that NF disproves Choice, and so proves Infinity. At this point if not before it was clear that there is a serious issue of showing that NF is consistent relative to some set theory in which we have confidence. There is no evidence that NF is any stronger than TST + Infinity, the lower bound established by Specker's result.

Note that NF or NFU supports the implementation of mathematics in the same style as TST, but with the representations of mathematical concepts losing their ambiguous character. The number 3 really is realized as the unique set of all sets with three elements, for example. The universe is a set and sets make up a Boolean algebra. Cardinal and ordinal numbers can be defined in the manner of Russell and Whitehead.

The apparent vulnerability to the paradox of Cantor is an illusion. Applying Cantor's theorem to the cardinality of the universe in NFU gives $|\iota``V| < |(V)| \leq |V|$ (the last inequality would be an equation in NF), from which we conclude that there are fewer singletons of objects than objects in the universe. The operation $(x \mapsto \{x\})$ is not a set function, and there is every reason to expect it not to be, as its definition is unstratified. The resolution of the Burali-Forti paradox is also weird and wonderful in NF(U), but would take us too far afield.

2.4 Tangled type theory TTT and TTTU

In 1995, this author described a reduction of the NF consistency problem to consistency of a typed theory, motivated by reverse engineering from Jensen's method of proving the consistency of NFU.

Let λ be a limit ordinal. It can be ω but it does not have to be.

In the theory TTT (tangled type theory) which we develop, each variable x is supplied with a type $\mathtt{type}(`x') < \lambda$; we are provided with countably many distinct variables of each type.

For any formula ϕ of the language of TST and any strictly increasing sequence s in λ, let ϕ^s be the formula obtained by replacing each variable of type i with a variable of type $s(i)$. To make this work rigorously, we suppose that we have a bijection from type i variables of the language of TST to type α variables of the language of TTT for each natural number i and ordinal $\alpha < \lambda$.

TTT is then the first order theory with types indexed by the ordinals below λ whose well formed atomic sentences '$x = y$' have $\mathtt{type}(`x') = \mathtt{type}(`y')$ and whose atomic sentences '$x \in y$' satisfy $\mathtt{type}(`x') < \mathtt{type}(`y')$, and whose axioms are the sentences ϕ^s for each axiom ϕ of TST and each strictly increasing sequence s in λ. TTTU has the same relation to TSTU (with the addition of constants $\emptyset^{\alpha,\beta}$ for each $\alpha < \beta < \lambda$ such that $(\forall \mathbf{x}_0^\alpha : \mathbf{x}_0^\alpha \not\in \emptyset^{\alpha,\beta})$ is an axiom).

It is important to notice how weird a theory TTT is. This is not cumulative type theory. Each type β is being interpreted as a power set of *each* lower type α. Cantor's theorem in the metatheory makes it clear that most of these power set interpretations cannot be honest.

There is now a striking

Theorem (Holmes): TTT(U) is consistent iff NF(U) is consistent.

Proof: Suppose NF(U) is consistent. Let (M, E) be a model of NF(U) (a set M with a membership relation E). Implement type α as $M \times \{\alpha\}$ for each $\alpha < \lambda$. Define $E_{\alpha,\beta}$ for $\alpha < \beta$ as $\{((x, \alpha), (y, \beta)) : xEy\}$. This gives a model of TTT(U). Empty sets in TTTU present no essential additional difficulties.

Suppose TTT(U) is consistent, and so we can assume we are working with a fixed model of TTT(U). Let Σ be a finite set of sentences in the language of TST(U). Let n be the smallest type such that no type n variable occurs in any sentence in Σ. We define a partition of the n-element subsets of λ. Each $A \in [\lambda]^n$ is put in a compartment determined by the truth values of the sentences ϕ^s in our model of TTT(U), where $\phi \in \Sigma$ and $\mathtt{rng}(s\lceil\{0, \ldots, n-1\}) = A$. By Ramsey's theorem, there is a homogeneous set $H \subseteq \lambda$ for this partition, which includes the range of a strictly increasing sequence h. There is a complete extension of TST(U) which includes ϕ iff the theory of our model of TTT(U) includes ϕ^h. This extension satisfies $\phi \leftrightarrow \phi^+$ for each $\phi \in \Sigma$. But this implies by compactness that the full Ambiguity Scheme $\phi \leftrightarrow \phi^+$ is consistent with TST(U), and so that NF(U) is consistent by the 1962 result of Specker.

We note that we can give a treatment of the result of Specker (rather different from Specker's own) using TTT(U). Note that it is easy to see that if we have a model of TST(U) augmented with a Hilbert symbol (a primitive term construction $(\epsilon x : \phi)$ (same type as x) with axiom scheme $\phi[(\epsilon x : \phi)/x] \leftrightarrow (\exists x : \phi))$ which cannot appear in instances of comprehension (the quantifiers are not defined in terms of the Hilbert symbol, because they do need to appear in instances of comprehension) and Ambiguity (for all formulas, including those which mention the Hilbert symbol) then we can readily get a model of NF, by constructing a term model using the Hilbert symbol in the natural way, then identifying all terms with their type-raised versions. All statements in the resulting type-free theory can be decided by raising types far enough (the truth value of an atomic sentence $(\epsilon x : \phi)\, R\, (\epsilon y : \psi)$ in the model of NF is determined by raising the type of both sides (possibly by different amounts) until the formula is well-typed in TST and reading the truth value of the type raised version; R is either $=$ or $\in$). Now observe that a model of TTT(U) can readily be equipped with a Hilbert symbol if this creates no obligation to add instances of comprehension containing the Hilbert symbol (use a well-ordering of the set implementing each type to interpret a Hilbert symbol $(\epsilon x : \phi)$ in that type as the first x such that ϕ), and the argument above for consistency of TST(U) plus Ambiguity with the Hilbert symbol goes through.

Theorem (essentially due to Jensen): NFU is consistent.

Proof: It is enough to exhibit a model of TTTU. Suppose $\lambda > \omega$. Represent type α as $V_{\omega+\alpha} \times \{\alpha\}$ for each $\alpha < \lambda$ ($V_{\omega+\alpha}$ being a rank of the usual cumulative hierarchy). Define $\in_{\alpha,\beta}$ for $\alpha < \beta < \lambda$ as

$$\{((x, \alpha), (y, \beta)) : x \in V_{\omega+\alpha} \wedge y \in V_{\omega+\alpha+1} \wedge x \in y\}.$$

This gives a model of TTTU in which the membership of type α in type β interprets each (y, β) with $y \in V_{\omega+\beta} \setminus V_{\omega+\alpha+1}$ as an urelement.

Our use of $V_{\omega+\alpha}$ enforces Infinity in the resulting models of NFU (note that we did not have to do this: if we set $\lambda = \omega$ and interpret type α using V_α we prove the consistency of NFU with the negation of Infinity). It should be clear that Choice holds in the models of NFU eventually obtained if it holds in the ambient set theory.

This shows in fact that mathematics in NFU is quite ordinary (with respect to stratified sentences), because mathematics in the models of TSTU embedded in the indicated model of TTTU is quite ordinary. The notorious ways in which NF evades the paradoxes of Russell, Cantor and Burali-Forti can be examined in actual models and we can see how they work (since they work in NFU in the same way they work in NF).

Of course Jensen did not phrase his argument in terms of tangled type theory. Our contribution here was to reverse engineer from Jensen's original argument for the consistency of NFU an argument for the consistency of NF itself, which requires additional input which we did not know how to supply (a proof of the consistency of TTT itself). An intuitive way to say what is happening here is that Jensen noticed that it is possible to skip types in a certain sense in TSTU in a way which is not obviously possible in TST itself; to suppose that TTT might be consistent is to suppose that such type skipping is also possible in TST.

2.4.1 How internal type representations unfold in TTT

We have seen above that TST can internally represent TST_n. An attempt to represent types of TTT internally to TTT has stranger results.

In TST the strategy for representing type i in type $n \geq i$ is to use the $n-i$-iterated singleton of any type i object x to represent x; then membership of representations of type $i-1$ objects in type i objects is represented by the relation on $n-i$-iterated singletons induced by the subset relation and with domain restricted to $n-(i+1)$-fold singletons. This is described more formally above.

In TTT the complication is that there are numerous ways to embed type α into type β for $\alpha < \beta$ along the lines just suggested. We define a generalized iterated singleton operation: where A is a finite subset of λ, ι_A is an operation defined on objects of type $\mathtt{min}(A)$. $\iota_{\{\alpha\}}(x) = x$. If A has $\alpha < \beta$ as its two smallest elements, $\iota_A(x)$ is $\iota_{A_1}(\iota_{\alpha,\beta}(x))$, where A_1 is defined as $A \setminus \{\mathtt{min}(A)\}$ (a notation we will continue to use) and $\iota_{\alpha,\beta}(x)$ is the unique type β object whose only type α element is x.

Now for any nonempty finite $A \subseteq \lambda$ with minimum α and maximum β. the range of ι_A is a set, and a representation of type α in type β. For simplicity we carry out further analysis in types $\beta, \beta+1, \beta+2\ldots$ though it could be done in more general increasing sequences. Use the notation τ_A for the range of ι_A, for each set A with β as its maximum. Each such set has a cardinal $|\tau_A|$ in type $\beta+2$. It is a straightforward argument in the version of TST with types taken from A and a small finite number of types $\beta+i$ that $2^{|\tau_A|} = |\tau_{A_1}|$ for each A with at least two elements. The relevant theorem in TST is that $2^{|\iota^{n+1}\text{“}X|} = \iota^n\text{“}X$, relabelled with suitable types from λ. We use the notation $\exp(\kappa)$ for 2^κ to support iteration. Notice that for any τ_A we have $\exp^{|A|-1}(|\tau_A|) = |\tau_{\{\beta\}}|$, the cardinality of type β. Now if A and A' have the same minimum α and maximum β but are of different sizes, we see that $|\tau_A| \neq |\tau_{A'}|$, since one has its $|A|-1$-iterated exponential equal to $|\tau_{\{\beta\}}|$ and the other has its $|A'|-1$-iterated exponential equal to $|\tau_{\{\beta\}}|$. This is odd because there is an obvious external bijection between the sets τ_A and $\tau_{A'}$: we see that this external bijection cannot be realized as a set. τ_A and $\tau_{A'}$ are representations of the same type, but this is not obvious from inside TTT. We recall that we denote $A \setminus \{\mathtt{min}(A)\}$ by A_1; we further denote $(A_i)_1$ as A_{i+1}. Now suppose that A and B both have maximum β and $A \setminus A_i = B \setminus B_i$, where $i < |A| \leq |B|$. We observe that for any concrete sentence ϕ in the language of TST_i, the truth value of ϕ in natural models with base type of sizes $|\tau_A|$ and $|\tau_B|$ will be the same, because the truth values we read off are the truth values in the model of TTT of versions of ϕ in exactly the same types of the model (truth values of ϕ^s for any s having $A \setminus A_i = B \setminus B_i$ as the range of an initial segment). This much information telling us that τ_{A_j} and τ_{B_j} for $j < i$ are representations of the same type is visible to us internally, though the external isomorphism is not. We can conclude that the full first-order theories of natural models of TST_i with base types $|\tau_A|$ and $|\tau_B|$ are the same as seen inside the model of TTT, if we assume that the natural numbers of our model of TTT are standard.

3 The model description

We give a complete description of what we claim is a model of tangled type theory. The construction may be supposed carried out in ZFC (or some weak subsystem thereof: we will see how much ZFC is needed).

3.1 Cardinal parameters

Let λ be a limit ordinal. Elements of λ will be indices of types in the model of tangled type theory. -1 is an index of an additional type in the structure we are using to build the model.

Let $\kappa > \lambda$ be an uncountable regular cardinal. We refer to sets of size smaller than κ as small and all other sets as large.

Let μ be a strong limit cardinal $> \kappa$ of cofinality at least κ.

3.2 Type -1: "Atoms", litters, and local cardinals

We choose a set τ_{-1} (type -1 of the structure we are building) of cardinality μ. Elements of τ_{-1} are conveniently supposed to be ordered triples with first component -1; we do not care about any other details of these objects. We may refer to objects of type -1 as "atoms". They are not understood here to be atoms in a conventional sense, but the analogous objects in earlier versions of the construction were atoms and I have this mental habit.

We choose a partition of τ_{-1} into sets of size κ which we call litters.

For each litter L, we define $[L]$, which we call the local cardinal of L, as the set of all subsets of τ_{-1} with small symmetric difference from L:

$$[L] = \{N \subseteq \tau_{-1} : |N \Delta L| < \kappa\}.$$

Elements of local cardinals we call *near-litters*. For any near-litter N belonging to $[L]$, we define $[N]$ as the same as $[L]$: a local cardinal is referred to as the local cardinal of each of its elements.

We introduce the notation $N^\circ = [N]^\circ$ for the litter with small symmetric difference from the near-litter N.

3.3 Set codes and alternative extensions: membership defined and extensionality enforced

For each $\alpha < \lambda$ we will define τ_α, the implementation of type α of the model. This definition is recursive: when we are defining type α and associated concepts, we are supposing that related concepts have already been defined for all $\beta < \alpha$.

A type α code is a triple (α, β, B) where $\beta < \alpha$ and $B \subseteq \tau_\beta$. There is an equivalence relation $\equiv_\alpha$ on the type α codes: the elements of τ_α are representatives of the equivalence classes under this relation on the type α codes which are symmetric in a sense to be defined later.

The equivalence relation $\equiv_{-1}$ is simply identity restricted to τ_{-1}.

We designate a pairwise disjoint family of subsets X_γ of the set of local cardinals, each of cardinality μ, indexed by ordinals $\gamma < \lambda$.

All codes $(\beta, -1, N)$ where N is a near-litter are stipulated to be symmetric and in fact elements of τ_β (of course we need to verify this when the relevant notion of symmetry is defined). We refer to such objects as typed near-litters. On each type α, $-1 \leq \alpha < \lambda$, we will choose a well-ordering $\leq_\alpha$ of order type μ (with corresponding strict well-ordering $<_\alpha$). We define $\iota(x)$ for $x \in \tau_\beta$ as the order type of $\leq_\beta$ restricted to $\{y \in \tau_\beta : y <_\beta x\}$. We define for distinct $\beta, \gamma < \alpha$, $\gamma \neq -1$ (β can be -1) $f_{\beta,\gamma}(x)$ as the local cardinal $[N]$ of the third component of the first $(\gamma, -1, N)$ in $<_\gamma$ such that N is a near-litter and $[N] \in X_\gamma$ and $\iota(\gamma, -1, M) > \iota(x)$ for each $M \in [N]$ and $[N] \neq f_{\beta,\gamma}(y)$ for any $y <_\beta x$. Notice that when we are constructing type α, we assume that we have constructed all earlier types and we have the information already to define $f_{\beta,\gamma}$ when β and γ are less than α.

We now define the equivalence relation on codes (which we will specialize to symmetric codes) and specify what the representatives are of the equivalence classes in the process.

A code $(\alpha, \beta, \emptyset)$ is equivalent to all codes $(\alpha, \gamma, \emptyset)$ and the representative code in this class is $(\alpha, -1, \emptyset)$.

For any code (α, β, B) with B nonempty and γ less than α and distinct from β and from -1 (β can be -1), we define $A_\gamma(\alpha, \beta, B)$ as

$$(\alpha, \gamma, \{(\gamma, -1, N) : N \in f_{\beta,\gamma}``B\}).$$

Notice that since the ranges of $f_{\beta,\gamma}$ and $f_{\beta,\delta}$ are disjoint if $\beta \neq \delta$, it follows that the ranges of maps A_γ with distinct indices are distinct. Clearly each A_γ is injective, and so the union of the A_γ's is a (partial) function with an inverse which we will call A^{-1}. Further, it should be clear from the definition of the f maps that no code can have infinitely many iterated images under A^{-1}.

We provide that $(\alpha, \beta, B) \equiv_\alpha (\alpha, \beta, B')$ iff $B = B'$.

We then provide that if (α, β, B) has an even number of iterated preimages under A^{-1} (including none, as an important possibility) and $\gamma \neq \beta$ then $(\alpha, \gamma, G) \equiv_\alpha (\alpha, \beta, B)$ iff $(\alpha, \gamma, G) = A_\gamma(\alpha, \beta, B)$, and in this case (α, β, B) is the representative of its equivalence class. This implies that each (α, β, B) which has an odd number of iterated preimages under A^{-1} is equivalent to its inverse image $A^{-1}(\alpha, \beta, B)$ (which we write (α, γ, G)) and to all $A_\delta(\alpha, \gamma, G)$ for $\delta \neq \gamma$.

The membership relations of the intended model of tangled type theory are then defined as follows: $(\gamma, \delta, D) \in_{TTT} (\alpha, \beta, B)$ (where both codes given are representatives of their equivalence classes) iff $\beta = \gamma$ and $(\gamma, \delta, D) \in B$ or $\beta \neq \gamma$ and $(\gamma, \delta, D) \in \pi_3(A_\gamma(\alpha, \beta, B))$. Note that in this latter case we will certainly have $\delta = -1$.

This all serves to enforce extensionality, but something much more radical needs to be done to make all this work, as we are assuming the existence of the maps $f_{\beta,\gamma}$ which witness that all the types are the same size. There must be a very strong restriction on what sets can appear as third components of codes in the model.

3.4 Permutations, symmetry and the model definition

We now define the notion of an α-allowable permutation (recalling that we have already defined what a β-allowable permutation is for each $\beta < \alpha$).

A -1-allowable permutation is a permutation π of τ_{-1} such that for any litter L, $\pi``L$ is a near-litter.

We stipulate that $(\alpha, \beta, \{b\})$ is an element of τ_α (a symmetric code, and clearly the representative of its equivalence class) for each $b \in \tau_\beta$, $\beta < \alpha$.

An α-allowable permutation is a permutation π of the type α codes with the property that each map τ_β defined implicitly by $\tau(\alpha, \beta, \{b\}) = (\alpha, \beta, \{\tau_\beta(b)\})$ is a β-allowable permutation, and in general $\pi(\alpha, \beta, B) = (\alpha, \beta, \pi_\beta``B)$, and which further satisfies the coherence condition that $(\alpha, \beta, B) \equiv_\alpha (\alpha, \gamma, G)$ implies $\pi(\alpha, \beta, B) \equiv_\alpha \pi(\alpha, \gamma, G)$.

This coherence condition can be unpacked.

$$(\alpha, \beta, \{b\}) \equiv_\alpha (\alpha, \gamma, \{(\gamma, -1, N) : N \in f_{\beta,\gamma}(b)\})$$

(where $\gamma \neq \beta$). Thus we expect

$$\pi(\alpha, \beta, \{b\}) \equiv_\alpha \pi(\alpha, \gamma, \{(\gamma, -1, N) : N \in f_{\beta,\gamma}(b)\}),$$

that is,

$$(\alpha, \beta, \{\pi_\beta(b)\}) \equiv_\alpha (\alpha, \gamma, \{(\gamma, -1, (\pi_\gamma)_{-1}``N) : N \in f_{\beta,\gamma}(b)\}),$$

so $f_{\beta,\gamma}(\pi_\beta(b)) = [(\pi_\gamma)_{-1}``L]$, where $f_{\beta,\gamma}(b) = [L]$.

Recalling the notations $N^\circ = [N]^\circ$ for the litter with small symmetric difference from the near-litter N, we can write this

$$f_{\beta,\gamma}(\pi_\beta(b)) = [(\pi_\gamma)_{-1}``f_{\beta,\gamma}(b)^\circ].$$

It is straightforward to show that this condition is equivalent to the coherence condition. Notice that π_β imposes some restrictions on π_γ, but only on the way it acts on certain typed near-litters (and of course there are reciprocal relations between π_γ and π_β).

It is straightforward to show that the image under an allowable permutation of a representative code is a representative code.

We provide extended notation for the lower type permutations on which an α-allowable permutation depends. For any nonempty set A of type indices with α as its largest element, define π_A as π if $A = \{\alpha\}$ and otherwise as $(\pi_{A \setminus \{\mathtt{min}(A)\}})_{\mathtt{min}(A)}$. We refer to permutations π_A as derivatives of π.

Define an α-support set as a small (cardinality $< \kappa$) set of pairs $((\beta, -1, x), A)$ where A is a set of type indices whose largest element is α and whose smallest element is β (β may be different for different elements of S), and x is either a singleton or a near-litter. It is convenient to stipulate in addition that when $((\beta, -1, x), A)$ and $((\beta, -1, x'), A)$ both belong to a support and x and x' are near-litters, that they are disjoint.

We say that a code (α, β, B) has support S iff S is a support set and any α-allowable permutation π with the property that for each $(x, A) \in S$, $\pi_A(x) = x$ also has the property $\pi(\alpha, \beta, B) = (\alpha, \beta, B)$.

We then stipulate that the elements of τ_α are precisely the representatives of equivalence classes of type α codes that have supports: such codes are said to be symmetric. It should be evident that typed near-litters are symmetric, and typed singletons of symmetric objects are symmetric, as we assumed above. It should also be evident that (α, β, B) will always be symmetric if $|B| < \kappa$: all small subsets of a type are realized in each higher type.

Once τ_α is constructed (and we verify that it is actually of cardinality μ) we choose a well-ordering $\leq_\alpha$ of τ_α with order type μ for use in the definition of more f maps.

There is lots to be proven, but that is the entire description.

Note for the formal verification project: I believe that the description of the model is complete and ready to be formalized. Do notice that a notion of support with more structure is introduced in the next section: you might want to work with that definition from the outset (supports are equipped with well-orderings and some extra decoration).

4 Showing that it is all true: proving that the structure described in the previous section is a model of tangled type theory

4.1 Strong supports defined

Treating supports as sets suffices for the model description, but we will need to analyze supports and orbits with more care, so it is better for purposes of the proof to regard a support as a well-ordering.

An α-support is a well-ordering of a small set of pairs $((\beta, -1, x), A, \gamma)$, where $\beta < \alpha$ (it may be different for different elements of the domain of the support), A is a finite subset of λ with maximum α and minimum β, x is either a near-litter or a singleton, and $0 \leq \gamma < \alpha$. Further, we impose the condition that if $((\beta, -1, x), A, \gamma)$ and $((\beta, -1, x'), A, \delta)$ belong to the domain of the support and x, x' are near-litters, then x and x' are disjoint.

We may write $x \leq_S y$ for $(x, y) \in S$, and $x <_S y$ when we also want to indicate that x, y are distinct.

If π is an α-allowable permutation and S is an α-support, we define $\pi[S]$ as $\{((\pi_A(x), A, \gamma), (\pi_B(y), B, \delta)) : ((x, A, \gamma), (y, B, \delta)) \in S\}$.

If S is an α-support, we define S^+ as $\{(x, A, \alpha) : (x, A, \alpha) \in S\}$.

We can then say that S is a support of X if X is an α-code, S is an α-support, and for any α-allowable permutation π, if $\pi[S^+] = S^+$ then $\pi(x) = x$. In some sense the items in the support with third components less than α are fluff, but they *are* important as we will see.

It is a useful observation that because μ has cofinality at least κ, there are no more than μ (and so exactly μ) near-litters, and similarly there are exactly μ supports.

It should be clear that the supports we have defined here do exactly the same work as the set supports in the model description (since the additional order structure and the third components of support domain elements actually do no work at this point).

For any α-support S and finite subset C of λ with minimum element greater than α, we define S^C as $\{((x, A \cup C, \gamma), (y, B \cup C, \delta)) : ((x, A, \gamma), (y, B, \delta)) \in S\}$.

A strong support is a support S with certain additional properties.

1. If $((\beta, -1, x), A, \gamma) \in S$ then x is a singleton or a litter.
2. If $((\beta, -1, \{x\}), A, \gamma) \in S$, then $((\beta, -1, L), A, \gamma) <_S ((\beta, -1, \{x\}), A, \gamma)$, where L is the litter containing x.
3. If $((\beta, -1, L), A, \delta) \in S$ and $[L] = f_{\gamma,\beta}(y)$, where $\gamma < \delta$, then there is a γ-support T of y such that $T^{A\setminus\{\beta\}} \subseteq S$ and each element of the domain of $T^{A\setminus\{\beta\}}$ is $\leq_S ((\beta, -1, L), A)$. Note that these conditions imply that there is an index-raised version of a strong γ support of y included in T.

It should be straightforward to see that any X with support S has a support S° which satisfies the first condition. Replace each element $((\beta, -1, x), A)$ of the domain of S with $((\beta, -1, x^\circ), A)$ and $((\beta, -1, \{y\}), A)$ for each y in the symmetric difference of x and x°.

We describe the process of extending a support to a strong support. Before each typed singleton element of the domain, insert the appropriate typed litter (removing an extra copy of it if it occurred later in the order). This will only need to be done once for each typed singleton element.

The condition $((\beta, -1, L), A, \delta) \in S$ and $[L] = f_{\gamma,\beta}(y)$ for $\gamma < \delta$ forces insertion of a γ-support for y. Of course this might cause further insertions. Notice that the third components of all inserted items will be $\leq \gamma < \delta$. So it is not possible for an infinite regress of insertions to occur which would cause the extended support to fail to be a well-ordering. (Of course, if an item is inserted which occurs later in the order, remove later occurrences).

The third components in the elements of support domains seem actually to be necessary to ensure that strong supports can always be produced. What the third component is doing is providing a comment on the index of the sub-support the element is required for.

Note for the formal verification project: This should be ready to go.

4.2 Freedom of action of allowable permutations

The practical application of strong supports is to the proof that allowable permutations act freely in a suitable sense, and in guiding applications of this theorem.

We claim that any locally small specification of values of derivatives of an allowable permutation at elements of type -1 can be realized.

We give an exact statement of what is meant, then we prove it.

An α-local approximation is a map π^0 whose domain is a set of pairs (A, x) where A is a nonempty finite subset of λ with maximum α and x is in type -1, and whose range is a subset of τ_{-1}. To state further conditions, we introduce the notation $\pi^0_A(x) = \pi^0(A, x)$ and state the further condition that each map π^0_A is injective and has domain the same as its range, and that the intersection of the domain of π^0_A with any litter is small (empty being an important case of small).

The freedom of action theorem asserts that for any α-local approximation π^0 there is an α-allowable permutation π such that $\pi_{A\cup\{-1\}}$ extends π^0_A for each A, and satisfying a further technical condition: we say that x is an exception of a -1-allowable permutation π iff (L being the litter containing x) either $\pi(x) \notin (\pi``L)^\circ$ or $\pi^{-1}(x) \notin (\pi^{-1}``L)^\circ$. The technical condition is that the permutation π obtained from π^0 has no exceptions of its derivatives other than elements of the domain of maps π^0_A.

We prove this by exhibiting a recursive procedure for computing π and its derivatives along a strong support; this succeeds because all objects have strong supports, and because computing all values of derivatives of π on type -1 allows computation of all derivatives of π at all types.

We consider an item $((\beta, -1, \{x\}), A, \gamma)$ and our aim is to compute $\pi_{A\cup\{-1\}}(x)$. By hypothesis of the recursion, we have already computed π_A at $((\beta, -1, L), A)$, where L is the litter which contains x.

For every litter L, we provide a well-ordering $<_L$ of type κ (fixed information preceding the start of the construction).

There are two cases. If (A, x) is in the domain of π^0, we compute $\pi_{A\cup\{-1\}}(x) = \pi^0_A(x)$ and we are done.

Otherwise we use the hypothesis of the recursion: we compute $\pi_{A\cup\{-1\}}(x)$ for any x in L with (A, x) not in the domain of π_0 using the fact that we have already computed $\pi_A(\beta, -1, L) = (\beta, -1, N)$: we define $\pi_{A\cup\{-1\}}$ to agree with the unique bijective map from the elements of L not in the domain of π^0_A to the elements of N° not in the domain of π^0_A which is strictly increasing in the sense that it sends larger objects in the sense of $<_L$ to larger objects in the sense of $<_{N^\circ}$.

Now we consider items of the form $((\beta, -1, L), A, \delta)$ in the strong support where L is a litter.

If $[L]$ is not of the form $f_{\gamma,\beta}(y)$ for any $\gamma < \mathtt{min}(A)$ and $y \in \tau_\gamma$, we assume that we have fixed in advance a bijective action χ_A on such local cardinals (all these maps are fixed information provided before the start of the construction).

We then compute $\pi_A((\beta, -1, L))$ as

$$(\beta, -1, \pi^0_A``L \cup (\chi_A([L])^\circ \setminus \pi^0_A``(\tau_{-1} \setminus L)))$$

we map L to the near-litter in $\chi_A([L])$ with the exact modifications required by the local bijection.

If $[L]$ is of the form $f_{\gamma,\beta}(y)$ for a $\gamma < \mathtt{min}(A)$ and $y \in \tau_\gamma$ then we proceed just as above but we take the action on $[L]$ from a different course: the coherence condition tells us that $[L]$ should be mapped to $f_{\gamma,\beta}(\pi_{A\setminus\{\beta\}\cup\{\gamma\}}(y))$, so we compute $\pi_A((\beta, -1, L))$ as

$$(\beta, -1, \pi^0_A``L \cup (f_{\gamma,\beta}(\pi_{A\setminus\{\beta\}\cup\{\gamma\}}(y))^\circ \setminus \pi^0_A``(\tau_{-1} \setminus L))),$$

which is essentially the same idea but a bit more complex.

The nasty recursive idea here is that we already know how to compute $\pi_{A\setminus\{\beta\}\cup\{\gamma\}}(y)$ from the embedded strong γ support of y because we assume as a hypothesis of the recursion that we already know how to carry out this computation for γ-supports, $\gamma < \alpha$. The local information we need about the γ allowable permutation (the local bijection to be used and the relevant χ maps) is included in the information we are given initially about π (though this data will have indices modified when used on the γ-support, of course). Notice that we certainly do know how to do it for 0-supports, because the recursive clause will never be invoked if $\alpha = 0$: the rest of the procedure tells us what to do.

Once we know how to carry out this calculation along any α-strong support, we can compute the derivatives of π on elements of type -1 along all type paths, and so compute the value of π and all of its derivatives on all types. The method of calculation clearly gives an allowable permutation without exceptions other than those dictated by the local bijection.

Note for the formal verification project: This section is vitally important and should be ready to work on (once the model and the definition of strong support are handled). Setting up the recursive definition of the computation may be nasty.

4.3 Types are of size μ (so the construction actually succeeds)

Now we argue that (given that everything worked out correctly already at lower types) each type α is of size μ, which ensures that the construction actually succeeds at every type.

Quite standard techniques show that if π is an α-allowable permutation and $X \in \tau_\alpha$ has α-support S, then $\pi(X)$ has α-support $\pi[S]$. This probably represents a chunk of formal verification work, as what is obvious to people is not always obvious to theorem provers.

For any support S and object x, we can define a function $\chi_{x,S}$ which sends $T = \pi(S)$ to $\pi(x)$ for every T in the orbit of S under the action of allowable permutations. We call such functions *coding functions*. Note that if $\pi[S] = \pi'[S]$ then $(\pi^{-1} \circ \pi')[S] = S$, so $(\pi^{-1} \circ \pi')(x) = x$, so $\pi(x) = \pi'(x)$, ensuring that the map $\chi_{x,S}$ for which we gave an implicit definition is well defined.

The strategy of our argument for the size of the types is to show that that there are $< \mu$ coding functions for each type whose domain includes a strong support, which implies that there are no more than μ (and so exactly μ) elements of each type, since every element of a type is obtainable by applying a coding function (of which there are $< \mu$) to a support (of which there are μ), and every element of a type has a strong support.

We describe all coding functions for type 0 (without concerning ourselves about whether supports are strong). The orbit of a 0-support in the allowable permutations is determined by the positions in the support order occupied by near-litters, and for each position in the support order occupied by a singleton, the position, if any, of the near-litter in the support order which includes it. There are no more than 2^κ ways to specify an orbit. Now for each such equivalence class, there is a natural partition of type -1 into near-litters, singletons, and a large complement set. Notice that near-litters in the partition will be obtained by removing any singletons in the domain of the support which are included in them. The partition has $\nu < \kappa$ elements, and there will be $2^\nu \leq 2^\kappa$ coding functions for that orbit in the supports, determined by specifying for each compartment in the partition whether it is to be included or excluded from the set computed from a support in that orbit. So there are no more than $2^\kappa < \mu$ coding functions over type 0.

We specify an object X and a strong support S for X, and develop a recipe for the coding function $\chi_{X,S}$ which can be used to see that there are $< \mu$ coding functions.

$X = (\alpha, \beta, B)$, where B is a subset of τ_β. By inductive hypothesis, each element b of B can be expressed as $\chi_{b,T_b}(T_b)$, where T_b is a strong support for b end extending S_β (which is defined as the largest β-support U such that $U^{\{\alpha\}} \subseteq S$).

We claim that $\chi_{X,S}$ can be defined in terms of the orbit of S in the allowable permutations and the set of coding functions χ_{b,T_b}. There are $< \mu$ sets of type β coding functions by inductive hypothesis, and we will argue that there are $< \mu$ orbits in the α-strong supports under allowable permutations, so this will imply that there are $\leq \mu$ elements of type α (it is obvious that there are $\geq \mu$ elements of each type). Of course we get $\leq \mu$ codes for each $\beta < \alpha$, but we know that $\lambda < \kappa < \mu$.

The definition that we claim works is that $\chi_{X,S}(U) = (\alpha, \beta, B')$, where B' is the set of all $\chi_{b,T_b}(U')$ for $b \in B$ and U' end extending U_β. Clearly this definition depends only on the orbit of S and the set of coding functions derived from B.

The function we have defined is certainly a coding function, in the sense that $\chi_{X,S}(\pi(S)) = \pi(\chi_{X,S}(S))$. What requires work is to show that $\chi_{X,S}(S) = S$, from which it follows that it is in fact the intended function.

Clearly each $b \in B$ belongs to $\chi_{X,S}(S)$ as defined, because $b = \chi_{b,T_b}(T_b)$, and T_b end extends S_β.

An arbitrary $c \in \chi_{X,S}(S)$ is of the form $\chi_{b,T_b}(U)$, where U end extends S_β and of course must be in the orbit of T_b under allowable permutations.

Our strategy is to show that there is an allowable permutation π which fixes X (so that π_β"$B = B$) such that $\pi_\beta[T_b] = U$, so that $\pi_\beta(b) = c$, so $c \in B$, whence $\chi_{X,S}(S)$ as defined is equal to X as required.

We build a support $S + T_b^{\{\alpha\}}$ and a support $S + U^{\{\alpha\}}$ with parallel structure by appending T_b (respectively U) to S then removing all but the first occurrence of each repeated item. The parallelism of structure is enforced by the identity of items taken from $S \setminus S_\beta$ in both supports and the fact that U is the image of T_b under some allowable permutation.

We construct an α-allowable permutation whose action takes one of these supports to the other, which will complete the plan given above. For this we use the freedom of action theorem. We define a local bijection which sends (A, x) to y just in case a $(\beta, A, \{x\})$ in the first support corresponds to a $(\beta, A, \{y\})$ in the other, and further enforces agreement of derivatives of the permutation to be constructed with derivatives of the known permutation π' sending T_b to U at exceptions of derivatives of π' which lie in litters in T_b. This causes singleton items in the first support to be mapped to the corresponding singleton items in the other support. We have to argue that litters in the domain of $S + T_b^{\{\alpha\}}$ (which is a strong support) are mapped to the correct near-litters in the domain of $S + U^{\{\alpha\}}$. If there is a failure, there is a first one. The local cardinal of the first failure is treated correctly (because a support of it is treated correctly), so the failure must consist in the map constructed having an exception which is moved by the permutation into or out of the litter in question (if a litter L in T_b is mapped to a near-litter N in U, all elements of $N \Delta N^\circ$ are treated correctly because they are values at exceptions of the known permutation, so a failure implies an exception of the constructed permutation lying in L which is not an exception of the known map and whose singleton is not an item in T_b, and there are no such exceptions.

The constructed map fixes X because of its identity action on S, and it sends b to c because its action sends T_b to U, which is what we claimed,

The final move is to argue that there are $< \mu$ orbits in the α-allowable permutations. The idea is that the orbit in which a permutation lies is completely determined by a certain amount of combinatorial information, similarly to what happened in type 0 but a bit more complex. The orbit is specified if we know the second and third components of each item (taken from λ items in each case, the first and second components of the first item, and whether the third component is a singleton or a near-litter. If this is a singleton, we want to know the position in the support of a near-litter containing it (which will be present). If this is a near-litter and its local cardinal is an image under an f map, we can extract from information about the preceding part of the support order a subsupport which is an index-raised version of a strong support for the near-litter and so for its inverse image under the f map: as part of our specification, we take the coding function which generates that inverse image.

We give exact details. If S is a strong support (or an image of a strong support under an allowable permutation), we define its specification S^* as a well-ordering of the same length in which an item $((\beta, -1, x), A, \gamma)$ will be replaced by an item $((\beta, -1, X), A, \gamma)$ in a way that we describe. If $x = \{y\}$, we replace x with $\{\delta\}$, where δ is the position of a typed near-litter containing y in the obvious sense. If x is a near litter which does not belong to any $f^{\gamma,\beta}(y)$ with $\gamma < \mathtt{min}(A)$, then $X = \emptyset$. If $x \in f_{\gamma,\beta}(y)$ with $\gamma < \mathtt{min}(A)$, then we extract the maximal strong support T of y such that $T^{\{\alpha\}} \subseteq S$, and set $X = \chi_{y,T}$, a coding function.

There is a straightforward argument by induction on the structure of strong supports that if we have two items with strong supports which have the same specification in the sense we have just described, there is an allowable permutation (by freedom of action) whose action sends the one support to the other, and so the one item to the other.

Suppose $S^* = T^*$: we discuss the construction of an allowable permutation π such that $\pi[S^*] = T^*$. It should not be a surprise that we construct the desired π as an extension (as in the freedom of action theorem) of a local bijection defined by consulting the parallel structures of S and T. If $((\beta, -1, \{x\}), A, \delta)$ and $((\beta, -1, \{y\}), A, \delta)$ appear at corresponding positions in S and T, we have π^0 send (A, x) to y. If $((\beta, -1, M), A, \delta)$ and $((\beta, -1, N), A, \delta)$ appear at corresponding positions in S and T and $((\beta, -1, \emptyset), A, \delta)$ appears at the corresponding position in $S^* = T^*$, we can provide that the map χ_A used in the freedom of action construction maps $[M]$ to $[N]$. If $((\beta, -1, M), A, \delta)$ and $((\beta, -1, N), A, \delta)$ appear at corresponding positions in S and T and $((\beta, -1, \emptyset), \chi_{y,T}, \delta)$ appears at the corresponding position in $S^* = T^*$, then we know by the inductive hypothesis that everything works before this item in the support that the action of the permutation π_A constructed so far will send $[M]$ to $[N]$. In both near-litter cases, we need to do a little more work to ensure that M is mapped exactly to N without exceptions. The idea is to extend π_A^0 so as to map each element of M which is not in M° to something in $N^\circ \cap N$, and each element of M° which is not in M to something not in N, and do the analogous things for $(\pi_A^0)^{-1}$, and then fill in orbits, which only requires countably many atoms for each orbit [this is why we take κ to be uncountable], with the rule that images and preimages chosen in the filling out process are chosen so as not to create exceptions (their images and preimages will agree with expected actions on near-litters in the supports, which is really action on their local cardinals, because all elements of the symmetric differences of near-litters with their corresponding litters are treated individually). The extension of this local bijection will have exactly the desired effect.

There are clearly $< \mu$ specifications since these are small structures built with components taken from sets of size $< \mu$. Notice the recursive dependency on the coding functions for items of lower types being taken from sets of size $< \mu$.

This completes the proof that each type is of size μ, which ensures that the construction described in the first section actually succeeds.

Note for the formal verification project: I think everything is here, but filling in details to the satisfaction of a theorem prover will be work.

4.4 The structure is a model of predicative TTT

There is then a very direct proof that the structure presented is a model of predicative TTT (in which the definition of a set at a particular type may not mention any higher type). Use E for the membership relation of the structure.

It should be evident that $xEy \leftrightarrow \pi_\beta(x)E\pi(y)$, where x is of type β, y is of type α, and π is an α-allowable permutation.

Suppose that we are considering the existence of $\{x : \phi^s\}$, where ϕ is a formula of the language of TST with $\in$ translated as E, and s is a strictly increasing sequence of types. The truth value of each subformula of ϕ will be preserved if we replace each x of type $s(i)$ with $\pi_{A_{s,i}}(x)$, where $x = s(j)$ and $A_{s,i}$ is the set of all s_k for $i \leq k \leq j+1$. The formula ϕ will contain various parameters a_i of types $s(n_i)$ and it is then evident that the set $\{x : \phi^s\}$ will be fixed by any $s(j+1)$-allowable permutation π such that π_{As,n_i} fixes a_i for each i. But this means that $(s(j+1), s(j), \{x : \phi^s\})$ is symmetric and belongs to type $s(j+1)$.

This procedure will certainly work if the set definition is predicative (all bound variables are of type no higher than that of x, parameters at the type of the set being defined are allowed).

There are easier proofs of the consistency of predicative tangled type theory; there is a reason of course that we have pursued this one.

Note for the formal verification project: We note that in order to avoid metamathematics, we actually suggest proving finitely many instances of comprehension with typed parameters from which the full comprehension scheme can be deduced. That there are such finite schemes (mod the infinite sequence of types) is well-known.

4.5 Impredicativity: verifying the axiom of union

What remains to complete the proof is that typed versions of the axiom of set union hold. That this is sufficient is a fact about predicative type theory. If we have predicative comprehension and union, we note that for any formula ϕ, $\{\iota^k(x) : \phi(x)\}$ will be predicative if k is taken to be large enough, then application of union k times to this set will give $\{x : \phi(x)\}$. $\iota(x)$ here denotes $\{x\}$. It is evidently sufficient to prove that unions of sets of singletons exist.

So what we need to show is that if $(\alpha, \beta, \{(\beta, \gamma, \{g\}) : g \in G\})$ is symmetric, then (β, γ, G) is symmetric.

Suppose that $(\alpha, \beta, \{(\beta, \gamma, \{g\}) : g \in G\})$ is symmetric. It then has a strong support S. We claim that S_β (same notion defined above) is a β-support for (β, γ, G).

Suppose that $\pi[S_\beta] = S_\beta$.

Any $g \in G$ has a strong support T which extends $(S_\beta)_\gamma$.

Construct using freedom of action technology a permutation π^* which acts as the identity on $S \setminus S_\beta$, such that π^*_β agrees with π on S_β [so in fact π^* will fix $(\alpha, \beta, \{(\beta, \gamma, \{g\}) : g \in G\})$) and $(\pi^*_\beta)_\gamma$ agrees with π_γ on T, both on items in T and on the orbits under π of exceptions of π which are in litters in T. It will follow that π^* fixes $(\alpha, \beta, \{(\beta, \gamma, \{g\}) : g \in G\})$ and that $(\pi^*_\beta)_\gamma$ has the same value as π_γ at g, which means that $\pi_\gamma(g) \in G$ (and the same things follow for the inverse of π) which verifies that that S_β (same notion defined above) is a β-support for (β, γ, G), so the axiom of union holds in the interpreted TTT.

The application of the freedom of action theorem works because no movement of typed atoms of type γ stipulated by the behaviour of π_γ can force movement of elements of $S \setminus S_\beta$, because this would have to be mediated by the action of π on S_β, which fixes all elements of S_β.

Note for formal verification project: This is a high level description which will probably acquire more detailed text if we get to it. The whole idea is here, I'm not saying there is a gap. But I have a strong suspicion that unwinding the details will induce more text.

5 Conclusions, extended results, and questions

[I have copied in the conclusions section of an older version, but what it says should be about right, and may require some revisions to fit in this paper. I also added the bibliography, which again is probably approximately the right one.]

This is a rather boring resolution of the NF consistency problem.

NF has no locally interesting combinatorial consequences. Any fact about sets of a bounded standard size which holds in ZFCA will continue to hold in models constructed using this strategy with the parameter κ chosen large enough. That the continuum can be well-ordered or that the axiom of dependent choices can hold, for example, can readily be arranged. Any theorem about familiar objects such as real numbers which holds in ZFCA can be relied upon to hold in our models (even if it requires Choice to prove), and any situation which is possible for familiar objects is possible in models of *NF*: for example, the Continuum Hypothesis can be true or false. It cannot be expected that *NF* proves any strictly local result about familiar mathematical objects which is not also a theorem of ZFCA (or even of ZFC).

Questions of consistency with NF of global choice-like statements such as "the universe is linearly ordered" cannot be resolved by the method used here (at least, not without major changes).

NF with strong axioms such as the Axiom of Counting (introduced by Rosser in [9], an admirable textbook based on *NF*), the Axiom of Cantorian Sets (introduced in [2]) or my axioms of Small Ordinals and Large Ordinals (introduced in my [4] which pretends to be a set theory textbook based on *NFU*) can be obtained by choosing λ large enough to have strong partition properties, more or less exactly as I report in my paper [5] on strong axioms of infinity in NFU: the results in that paper are not all mine, and I owe a good deal to Solovay (unpublished conversations and [11]).

That NF has α-models for each standard ordinal α should follow by the same methods Jensen used for NFU in his original paper [7]. No model of NF can contain all countable subsets of its domain; all well-typed combinatorial consequences of closure of a model of TST under taking subsets of size $< \kappa$ will hold in our models, but the application of compactness which gets us from TST + Ambiguity to NF forces the existence of externally countable proper classes, a result which has long been known and which also holds in NFU.

We mention some esoteric problems which our approach solves. The Theory of Negative Types of Hao Wang (TST with all integers as types, proposed in [14]) has ω-models; an ω-model of NF gives an ω-model of TST immediately. This question was open.

In ordinary set theory, the Specker tree of a cardinal is the tree in which the top is the given cardinal, the children of the top node are the preimages of the top under the map $(\kappa \mapsto 2^\kappa)$, and the part of the tree below each child is the Specker tree of the child. Forster proved using a result of Sierpinski that the Specker tree of a cardinal must be well-founded (a result which applies in ordinary set theory or in NF(U), with some finesse in the definition of the exponential map in NF(U)). Given Choice, there is a finite bound on the lengths of the branches in any given Specker tree. Of course by the Sierpinski result a Specker tree can be assigned an ordinal rank. The question which was open was whether existence of a Specker tree of infinite rank is consistent. It is known that in NF with the Axiom of Counting the Specker tree of the cardinality of the universe is of infinite rank. Our results show that Specker trees of infinite rank are consistent in ZFA [this requires discussion in the tangled type theory approach used here, but is still true]. We are confident that our permutation methods can be adapted to ZFC using forcing in standard ways to show that Specker trees of infinite rank can exist in ZF.

We believe that NF is no stronger than TST + Infinity, which is of the same strength as Zermelo set theory with separation restricted to bounded formulas. Our work here does not show this, as we need enough Replacement for existence of $\beth_{\omega_2}$ at least. We leave it to others to tighten things up and show the minimal strength that we expect holds.

Another question of a very general and amorphous nature which remains is: what do models of NF look like in general? Are all models of NF in some way like the ones we describe, or are there models of quite a different character?

I am not sure that all references given here will be used in this version.

References

[1] Forster, T.E. [1995] Set Theory with a Universal Set, exploring an untyped Universe Second edition. Oxford Logic Guides, Oxford University Press, Clarendon Press, Oxford.

[2] Henson, C.W. [1973a] Type-raising operations in NF. Journal of Symbolic Logic 38 , pp. 59-68.

[3] Holmes, M.R. “The equivalence of NF-style set theories with ”tangled” type theories; the construction of omega-models of predicative NF (and more)”. *Journal of Symbolic Logic* 60 (1995), pp. 178-189.

[4] Holmes, M. R. [1998] Elementary set theory with a universal set. volume 10 of the Cahiers du Centre de logique, Academia, Louvain-la-Neuve (Belgium), 241 pages, ISBN 2-87209-488-1. See here for an on-line errata slip. By permission of the publishers, a corrected text is published online; an official second edition will appear online eventually.

[5] Holmes, M. R. [2001] Strong Axioms of infinity in NFU. Journal of Symbolic Logic, 66, no. 1, pp. 87-116.
(“Errata in ‘Strong Axioms of Infinity in NFU’ ”, JSL, vol. 66, no. 4 (December 2001), p. 1974, reports some errata and provides corrections).

[6] Jech, Thomas, *Set theory*, Academic Press 1978, pp. 199-201.

[7] Jensen, R.B. “On the consistency of a slight(?) modification of Quine’s NF”. *Synthese* 19 (1969), pp. 250-263.

[8] Quine, W.V., “New Foundations for Mathematical Logic”. *American Mathematical Monthly* 44 (1937), pp. 70-80.

[9] Rosser, J. B. [1978] Logic for mathematicians, second edition. Chelsea Publishing.

[10] Scott, Dana, “Definitions by abstraction in axiomatic set theory”, *Bull. Amer. Math. Soc.*, vol. 61, p. 442, 1955.

[11] Solovay, R, “The consistency strength of NFUB”, preprint on `arXiv.org`, `arXiv:math/9707207 [math.LO]`

[12] Specker, E.P. “The axiom of choice in Quine’s new foundations for mathematical logic”. *Proceedings of the National Academy of Sciences of the USA* 39 (1953), pp. 972-975.

[13] Specker, E.P. [1962] “Typical ambiguity”. *Logic, methodology and philosophy of science*, ed. E. Nagel, Stanford University Press, pp. 116-123.

[14] Wang, H. [1952] Negative types.