New Foundations is consistent

M. Randall Holmes

July 19, 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.

1.1 Version updates

7/19/2022: Further refinement of text from the previous day.

7/18/2022: Corrected errors in the argument that there are μ elements in each type.

7/13/2022: Adding another additional section not in the main development, detailing a different approach to supports and the freedom of action theorem, similar to Peter's approach in the Lean project.

6/29/2022: I wrote a section which is not part of the official development (at least not yet) exploring Peter's preferred representation of model elements. I also fixed a typo in the subscripting in the definition of f maps (an extra subscript had survived from an earlier version).

6/27/2022: Explicit formal definition of $\equiv_\beta$ inserted due to questions about boundary cases.

6/14/2022: Corrected an automated replacement disaster in the tangled web sections left over from yesterday, and minor edits.

6/13/2022: Thank you Peter for correcting slip in statement of the definition of A^{-1}.

Later: added a comment about the fact that in these models the power set of a well-orderable set is well-orderable, which is a significant discovery.

Later: clarifications (notably in the proof of the Freedom of Action Theorem)

Later: extensive work on (hopefully) cleaning up bad notation. Different format for specifications.

6/12/2022: Further tightening and improvement of language in sections 4.3 and 4.5. There is a general (helpful!) fact about the way extensions to strong supports of index-raised versions of strong supports work which I had not properly appreciated.

Later: further fiddly edits.

6/11/2022: Correction and improvement of section 4.3. There is now only one induction argument on strong supports in the section, and the repair was needed because the one originally given first could not be proved without the second.

Later: major rewriting of section 4.3, which I think brings it much closer to the standard needed for the verification project.

6/10/2022: Replaced ι as used to represent ordinal position in a type with ι_*, to avoid confusion with the use of ι for the singleton operator which occurs elsewhere in the paper.

Added a proof that A_δ's commute with allowable permutations, supporting the fact that the unpacked coherence condition is the same as the original one.

Clarified the account of the proof that the structure satisfies predicative TTT, which was missing important details. It's still too brief, but it is a sketch of an standard argument found in other contexts.

Cleaned up the proof that the structure satisfies the axiom of union. The sketch I had before was along the right lines but there was a gap in it much too large to expect a reader to fill in.

There are other edits; this was a complete editing pass.

6/9/2022: Some debugging in section 4. Some of the detailed descriptions of applications of freedom of action guided by strong supports needed corrections.

6/8/2022: Fixed some typos (or mental slips of the sort we call typos by courtesy) and made some adjustments necessitated by the new treatment of strong supports.

6/7/2022: Major improvements in the treatment of strong supports. Removed the third components from items in supports and imposed a technical condition on the designated well-orderings of each type. **Significant changes which will require some adjustments of the blueprint** (I do not think extensive ones). Later on 6/7, corrected some howlers about litters vs near-litters in the new material. 5 pm 6/7, more editing.

6/6/2022: Added a section on tangled webs of cardinals for reference. Removed an unnecessary clause from the definition of support (which will require a modification of the blueprint).

6/5/2022: 3:30 pm Cambridge: An editing pass through the whole document. I think that if we get to the induction proofs in section 4 which work along strong supports, there will need to be special notes written for each proof.

6/5/2022, 11:11 am Cambridge: Minor fixes here and there in the adjusted text from last time; some clarification of text in the proof of freedom of action.

6/4/2022, afternoon Cambridge time: Explicitly described the procedures for extending supports to strong supports. Second posting repairing some glitches in the evening.

6/3/2022, 3 pm Cambridge time: Major revisions to the presentation of the model and its verification inspired by approaches taken in the blueprint. No essential mathematical changes; one modification is that supports are now well-orderings of triples from the outset, as in the blueprint.

5/29/2022, morning: Considering backward action of the blueprint on the paper. Introduced the concrete definition of type -1 and rephrased things much as in the blueprint.

5/25/2022, afternoon: Accidentally left out the fact $\mu > \kappa$.

5/23/2022, early afternoon: Further copy editing; no mathematical changes.

5/23/2022, morning: Fixes error in construction of f maps.

5/22/2022, evening: Fixes a typo, adds remarks motivated by our conversations on Saturday.

5/21/2022, morning: This version has some upgrades to organization and appearance which are part of preparation for its submission as a Paper. These should generally help students in the Lean project, too.

5/18/2022, 9 am Boise time: corrected a typo pairs to triples in definition of structured supports; changed cofinality requirement for μ to cofinality $> \kappa$ instead of $\geq \kappa$.

5/18/2022, 9:30 am Boise time: corrected definition of A_γ; left out a necessary application of set union. I suspect Lean would have noticed a mistake there.

5/18/2022, 9:31 am Boise time: changed cofinality back to $\leq \kappa$: inserting some discussion.

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 des 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(U) exhibits a symmetry which is important in the sequel.

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

If ϕ is a formula, define ϕ^+ as the result of replacing every variable x (free and bound) in ϕ with x^+ (and occurrences of $\emptyset^i$ with $\emptyset^{i+1}$ if this is in use). 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 ([14]), 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* ([13]). 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 ([21]), 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 ([19]).

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 (in [7]) 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 in [10]) 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 $|\iota``V| < |\mathcal{P}(V)| \le |V|$, 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 [11], 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^* and each variable x^*, 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 $\sigma('x') = \sigma('y')$ and for each atomic subformula '$x \in y$' of ϕ we have $\sigma('x') + 1 = \sigma('y')$. 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 and variable x 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 ([2]), 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 [9], 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 ([18], 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 [17], 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 $|\iota``V| < |\wp(V)| \leq |V|$ (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 [4], 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 $\texttt{type}(`x') < \lambda$; 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 $\texttt{type}(`x') = \texttt{type}(`y')$ and whose atomic sentences '$x \in y$' satisfy $\texttt{type}(`x') < \texttt{type}(`y')$, 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 $(\forall \mathbf{x}_0^\alpha : \mathbf{x}_0^\alpha \notin \emptyset^{\alpha,\beta})$ 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 $\texttt{rng}(s \lceil \{0, \dots, n-1\}) = A$. 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 that they work and 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. The development of the model does not depend on reading this section.

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 $\min(A)$. $\iota_{\{\alpha\}}(x) = x$. 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 $A \setminus \{\min(A)\}$ (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 $A \setminus \{\min(A)\}$ 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.

2.4.2 Tangled webs of cardinals: a suggestion of another approach not followed here

Nothing in the construction of a model of tangled type theory and verification that it is a model which appears below depends on anything in this section.

It is straightforward to transform a model of TST into a model of bounded Zermelo set theory (Mac Lane set theory) with atoms or without foundation (this depends on how type 0 is handled). Specify an interpretation of type 0 either as a set of atoms or a set of self-singletons. Then interpret type $i + 1$ as inhabited by sets of type i objects in the obvious way, identifying type $i + 1$ objects with objects of lower type which happen to have been assigned the same extension.

In a model of TTT, do this along some increasing sequence of types of order type ω whose range includes an infinite ordinal α. In the resulting model of bounded Zermelo set theory, let τ_A represent the cardinality of the range of ι_A as in the previous discussion (for nonempty subsets of type A all with maximum the same infinite α). Suppose further for the sake of argument that our model of TTT is λ-complete, in the sense that any subset of a type of cardinality that of λ or less is implemented as a set in each higher type. It will follow that $A \mapsto \tau_A$ is actually a function. [It is an incidentally interesting fact that the models we construct (with no dependence on this section) actually have this completeness property].

We describe the situation which holds for these cardinals.

We work in Mac Lane set theory. Choice is not assumed, and we use the Scott definition of cardinals.

Definition: If A is a nonempty finite set of ordinals which is sufficiently large, we define A_1 as $A \setminus \mathtt{min}(A)$ and A_0 as A, A_{i+1} as $(A_i)_1$.

Definition: A tangled web of cardinals of order α (an infinite ordinal) is a function τ from the set of nonempty sets of ordinals with α as maximum to cardinals such that

1. If $|A| > 1$, $\tau(A_1) = 2^{\tau(A)}$.

2. If $|A| \geq n$, the first order theory of a natural model of TST_n with base type $\tau(A)$ is completely determined by $A \setminus A_n$, the n smallest elements of A.

The bookkeeping in different versions of this definition in different attempts at a tangled web version of the proof of the consistency of NF have been different (an obvious point about the version given here is that the top ordinal α could be omitted). Another remark is that it is clear that asserting the existence of a tangled web is stronger than simple TTT: it requires $\lambda > \omega$, and the λ-completeness of course is a strong assumption in the background. All variants that I have used support versions of the following

Theorem: If there is a model of Mac Lane set theory in which there is a tangled web of cardinals τ, then NF is consistent.

Proof: Let Σ be a finite set of sentences of the language of TST. Let n be larger than any type mentioned in any formula in Σ. Partition $[\alpha]^n$ into compartments in such a way that the compartment that a set A is put into depends on the truth values of the sentences in Σ in natural models of TST_n with base type of size $\tau(B)$ where $B \setminus B_n = A$. This partition of $[\alpha]^n$ into no more than $2^{|\Sigma|}$ compartments has a homogeneous set H of size $n + 1$. The natural models of TST_n with base types of size $\tau(H)$ and base types of size $\tau(H_1)$ have the same truth values for sentences in Σ, so the model of TST with base type $\tau(H)$ satisfies the restriction of the Ambiguity Scheme to Σ, so the full Ambiguity Scheme is consistent by compactness, so TST + Ambiguity is consistent so NF is consistent.

Our initial approach to proving our theorem was to attempt a Frankel-Mostowski construction of a model of Mac Lane set theory with a tangled web of cardinals. We do know how to do this, but we believe from experience that constructing a model of tangled type theory directly is easier, though tangled type theory is a nastier theory to describe.

We think there is merit in giving a brief description of a situation in a more familiar set theory equivalent to (a strengthening of) the very strange situation in a model of tangled type theory. This section is also useful here because it supports the discussion in the conclusion of one of the unsolved problems which is settled by this paper.

3 The model description

In this section, 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. Elements of $\lambda \cup \{-1\}$ are called type indices; type indices other than -1 are called proper type indices. The order on the type indices is the obvious one in which -1 is less than the elements of λ, which inherit their usual order.

Nonempty finite subsets of $\lambda \cup \{-1\}$ are called *extended type indices*; ones that do not contain -1 are called *proper* extended type indices. An extended type index A may be understood as referring to the type $\mathtt{min}(A)$ in the role of type 0 in a model of $\mathrm{TST}_{|A|}$ whose higher types are indexed by the larger elements of A.

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 $\geq \kappa$.

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

We define τ_{-1} as $\{(-1, \nu, \alpha) : \nu < \mu \wedge \alpha < \kappa\}$.

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 we have this mental habit.

For each $\nu < \mu$ we define L_ν as $\{(-1, \nu, \alpha) : \alpha < \kappa\}$. We call the sets L_ν *litters*. Note that the set $\Lambda = \{L_\nu : \nu < \mu\}$ is a partition of the set τ_{-1} of size μ into subsets of size κ.

We say that a set N is a *near-litter* iff $N \subseteq \tau_{-1} \wedge (\exists \nu : |N \Delta L_\nu| < \kappa)$: a near-litter is a subset of τ_{-1} with small symmetric difference from some litter. Of course litters are near-litters.

The relation of having small symmetric difference is an equivalence relation on near-litters.

For any near-litter N we define $[N]$ as the set $\{M \subseteq \tau_{-1} : |N \Delta M| < \kappa\}$ of all near-litters M with small symmetric difference from N. We call the sets $[N]$ *local cardinals* and say that $[N]$ is the local cardinal of N. This convention is motivated by the Frege-Russell-Whitehead notion of cardinal (which is the natural one to use in TST and related theories).

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

Because the cofinality of μ is $\geq \kappa$, there are μ near-litters. One might be concerned with the fact that if μ has cofinality κ, it might have more than

μ subsets of size κ: but it still has only μ subsets of size $< \kappa$, and that is what matters for counting the near-litters: a near-litter is determined as the symmetric difference of a litter (μ of these) and a small subset (cardinality $< \kappa$) of τ_{-1} (which is of size μ) and there are only μ small subsets of τ_{-1}. If the cofinality of μ were less than κ, the cardinal arithmetic pathology mentioned as of concern could come into play.

We designate a pairwise disjoint family of subsets $\mathbf{X}_{(\beta,\gamma)}$ of the set of local cardinals, each of cardinality μ, indexed by ordered pairs (β, γ) with $-1 \le \beta < \lambda$ and $\gamma \in \lambda \setminus \{\beta\}$. These play an essential role in defining alternative extensions of the same object.

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 $\beta < \alpha$.

So, we fix $\alpha < \lambda$, a proper type index, and we assume that for all type indices $< \alpha$ the construction we are describing has already been carried out: we define τ_α and related notions.

Definition (codes): We suppose for $\beta < \alpha$ a proper type index that a β-code has already been defined as a triple (β, γ, G) where $-1 \le \gamma < \beta$ and $G \subseteq \tau_\gamma$.

A α-code is defined now as a triple (α, β, B) where $-1 \le \beta < \alpha$ and $B \subseteq \tau_\beta$.

A hypothesis of the recursion: proper types are inhabited by codes: We remark that τ_β for $-1 < \beta < \alpha$ is a (proper) subset of the collection of β-codes (a hypothesis of the recursion).

Definition (f maps): All codes $(\beta, -1, N)$ where $0 \le \beta < \alpha$ and N is a near-litter are stipulated to be elements of τ_β (this will be verified as following from inductive hypotheses given in more detail later). We refer to such objects as typed near-litters.

On each type δ, $-1 \le \delta < \alpha$, we have chosen a well-ordering $\le_\delta$ of order type μ (with corresponding strict well-ordering $<_\delta$). Technical conditions on these well-orderings are stated at the end of this section.

We define $\iota_*(x)$ for $x \in \tau_\beta$ as the order type of $\le_\beta$ restricted to $\{y \in \tau_\beta : y <_\beta x\}$ (note that ι_* is defined for elements of any type).

Where γ is a type index less than α and δ is a proper type index distinct from γ and also less than α, and $x \in \tau_\gamma$, we define $f_{\gamma,\delta}(x)$ as the local cardinal $[N]$ of the third component of the first $(\delta, -1, N)$ in $<_\delta$ such that N is a near-litter and $[N] \in \mathbf{X}_{\gamma,\delta)}$ and $\iota_*(\delta, -1, M) > \iota_*(x)$ for each $M \in [N]$ and $[N] \ne f_{\gamma,\delta}(y)$ for any $y <_\gamma x$.

Note that this definition does not really depend on α: it will give the same result at any stage of the construction beyond γ and δ.

Definition (alternative extension map): For any β-code (β, γ, G) with $\beta \leq \alpha$ a proper type index and δ a proper type index distinct from γ and less than β, and G nonempty, we define $A_\delta(\beta, \gamma, G)$ as

$$(\beta, \delta, \{(\delta, -1, N) : N \in \bigcup(f_{\gamma, \delta} \text{``} G)\}).$$

Notice that the value computed here does not depend on α: this computation at earlier stages in the recursion gives the same result at every stage whose index is at least β.

Observations and definition of A^{-1}: Notice that since the ranges of $f_{\gamma, \delta}$ and $f_{\gamma', \delta'}$ are disjoint unless $\gamma = \gamma'$ and $\delta = \delta'$, it follows that the ranges of maps A_γ with distinct indices are disjoint. Clearly each A_γ is injective, and so the union of the A_γ^{-1}'s is a (partial) function which we will call A^{-1}.

Lemma: Further, it should be clear from the definition of the f maps that no code can have infinitely many iterated images under A^{-1}: when A^{-1} is applied, typed local cardinals (sets of the form $\{(\delta, -1, M) : M \in [N]\}$) which are subsets of the extension are replaced with (singletons of) objects earlier in the well-ordering on their type than all elements of the typed local cardinal are in their own type. Look at what happens to the position of the first element in an extension under the appropriate order as A^{-1} is applied repeatedly.

Discussion of equivalence of codes: We now define an equivalence relation $\equiv_\beta$ for each type index $\beta \leq \alpha$.

It is important to notice that the definition of $\equiv_\beta$ will be the same from stage β of the recursion onward.

The relation $\equiv_{-1}$ is simply the restriction of equality to τ_{-1}.

For each proper type index β, $(\beta, \gamma, \emptyset) \equiv_\beta (\beta, \delta, D)$ iff $D = \emptyset$, and the representative member of this equivalence class is $(\beta, -1, \emptyset)$.

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

We give a formal definition.

Definition: $(\beta, \gamma, G) \equiv_\beta (\beta, \delta, D)$ is defined:

1. If G is empty, $(\beta, \gamma, G) \equiv_\beta (\beta, \delta, D)$ iff D is empty.

2. If (β, γ, G) has an even number of iterated images under A^{-1} then $(\beta, \gamma, G) \equiv_\beta (\beta, \delta, D)$ iff either $(\beta, \gamma, G) = (\beta, \delta, D)$ or $\delta \neq \gamma$ (and $\delta \neq -1$) and $A_\delta(\beta, \gamma, G) = (\beta, \delta, D)$

3. If (β, γ, G) has an odd number of iterated images under A^{-1} then $(\beta, \gamma, G) \equiv_\beta (\beta, \delta, D)$ iff either $(\beta, \gamma, G) = (\beta, \delta, D)$ or $A^{-1}(\beta, \gamma, G) \equiv_\beta (\beta, \delta, D)$

A hypothesis of the recursion: elements of proper types are representative codes:
We note further that elements of τ_β for each $\beta < \alpha$ are representative elements of their equivalence classes under $\equiv_\beta$. This will be verifiable from a more complete description of our assumptions about τ_β for $\beta < \alpha$ which is given below.

Definition (membership of the intended model of TTT): The membership relations of the intended model of tangled type theory are then defined as follows: $(\delta, \epsilon, E) \in_{TTT} (\beta, \gamma, G)$ (the triples are codes and $-1 < \delta < \beta \leq \alpha$, and both codes given are representatives of their equivalence classes) iff $\gamma = \delta$ and $(\delta, \epsilon, E) \in B$ or $\gamma \neq \delta$ and $(\delta, \epsilon, E) \in \pi_3(A_\delta(\beta, \gamma, G))$. Note that in this latter case we will certainly have $\epsilon = -1$. This does define membership of codes of lower types in type β codes which will eventually be seen not to belong to τ_β; this is harmless.

Theorem: If X and Y are β-codes ($\beta \leq \alpha$) which are representatives of their equivalence classes, and γ is a proper type index less than β,

$$(\forall Z \in \tau_\gamma : Z \in_{TTT} X \leftrightarrow Z \in_{TTT} Y) \rightarrow X = Y).$$

This should be evident from the method of construction. Each representative code has an extension in each lower type, and no two representative codes can have the same extension in any lower type.

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 of cardinality no greater than μ. There must be a very strong restriction on what sets can appear as third components of codes in the model.

Note for the Lean workers: The recursive definition of α-codes does need to take τ_β for $\beta < \alpha$ into account (the restriction of the third component of a code to subsets of τ_β is essential, but all one needs to know about τ_β is that it is of size μ and contains all typed near-litters). A treatment of the first section could be free-standing, with a parameter in the definition of codes a function sending each β to a subset of size μ of the β-codes containing all type β near-litters, to serve as τ_β.

Then the function taking α to the set of α-symmetric representative codes might be fed in as that parameter. I don't know if this is helpful...

3.4 Permutations, symmetry and the model definition

We try to improve intelligibility of the notion of "allowable permutation" that we now define by providing a preliminary notion.

Stipulation: We postulate as part of the hypotheses of the recursion that $(\beta, \gamma, \{x\}) \in \tau_\beta$ for any $x \in \tau_\gamma$, for any $\gamma < \beta < \alpha$. This will be verifiable from a more complete description of our assumptions about τ_β for $\beta < \alpha$ which is given below. We refer to such objects as typed singletons; when $\gamma = -1$, we may refer to them as typed atoms.

Definition (structural permutation): We define the notion of β-structural permutation for each type index $\beta \leq \alpha$. It should be evident that the definition given for β-structural permutation will give the same result at each stage of the recursive construction with index $\geq \beta$.

A -1-structural permutation is simply a permutation of type -1.

If β is a proper type index, a permutation π of β-codes is β-structural iff there is for each type index $\gamma < \beta$ a γ-structural permutation π_γ such that for any β-code (β, γ, G) we have $\pi(\beta, \gamma, G) = (\beta, \gamma, \pi_\gamma "G)$.

Notice that if π is a β-structural permutation, each π_γ is definable from π by $\pi_\gamma(x) = \bigcup(\pi_3(\pi(\beta, \gamma, \{x\})))$. Strictly speaking, this defines π_γ only on elements of τ_γ, but the definition could readily be extended to all γ-codes in the event that this were needed. Note further (a fact we do not use directly, though it must be noticed) that we are implicitly assuming that all π_γ for $\gamma < \beta$ map elements of τ_γ to elements of τ_γ.

Definition (notation for derivatives of a permutation): We provide extended notation for the lower indexed structural permutations which β-structural permutation depends, for any proper type index $\beta \leq \alpha$. For any finite set A of type indices with β as its largest element, define π_A as π if $A = \{\beta\}$ and otherwise as $(\pi_{A \setminus \{\mathtt{min}(A)\}})_{\mathtt{min}(A)}$. We refer to permutations π_A as derivatives of π.

Definition (allowable permutation): For each $\beta \leq \alpha$, we define a β-allowable permutation as a β-structural permutation with certain additional properties.

A -1-allowable permutation is a permutation π of τ_{-1} with the property that for any near-litter N, $\pi "N$ is a near-litter. Notice that a -1-allowable permutation determines a permutation of the local cardinals in a natural way.

For $\beta > -1$, a β-allowable permutation is a β-structural permutation π such that each π_γ for $\gamma < \beta$ is γ-allowable, and further for any codes X, Y, $X \equiv_\beta Y \leftrightarrow \pi(X) \equiv_\beta \pi(y)$ (this is referred to as the coherence condition).

Discussion (unpacking the coherence condition): This coherence condition can be unpacked.

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

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

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

that is,

$$(\beta, \gamma, \{\pi_\gamma(g)\}) \equiv_\beta (\beta, \delta, \{(\delta, -1, (\pi_\delta)_{-1} \text{``} N) : N \in f_{\gamma,\delta}(g)\}),$$

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

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

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

Allowable permutations commute with A_δ: We show this.

$$\pi(A_\delta(\beta, \gamma, G)) = \pi(\beta, \delta, \{(\delta, -1, N) : N \in \bigcup f_{\gamma,\delta} \text{``} G\})$$

$$= (\beta, \delta, \{(\delta, -1, (\pi_\delta)_{-1} \text{``} N) : N \in \bigcup f_{\gamma,\delta} \text{``} G\})$$

$$= (\beta, \delta, \bigcup_{g \in G} \{(\delta, -1, (\pi_\delta)_{-1} \text{``} N) : N \in f_{\gamma,\delta}(g)\})$$

$$= (\beta, \delta, \bigcup_{g \in G} \{(\delta, -1, N) : N \in \{(\pi_\delta)_{-1} \text{``} M : M \in f_{\gamma,\delta}(g)\}\})$$

$$= (\beta, \delta, \bigcup_{g \in G} \{(\delta, -1, N) : N \in f_{\gamma,\delta}(\pi_\gamma(g))$$

$$= (\beta, \delta, \{(\delta, -1, N) : N \in \bigcup f_{\gamma,\delta} \text{``} (\pi_\gamma \text{``} G)\}) = A_\delta(\beta, \gamma, \pi_\gamma \text{``} G)$$

$$= A_\delta(\beta, \gamma, \pi_\gamma \text{``} G) = A_\delta(\pi(\beta, \delta, G))$$

Consequences of this calculation: That each allowable permutation commutes with A_δ implies that each allowable permutation also commutes with A^{-1}, and this shows that the unpacked coherence condition implies and so is exactly equivalent to the original coherence condition. It also shows that representative codes are mapped by allowable permutations to representative codes.

Definition (support and symmetry): An β-support is a well-ordering of a small set of pairs of the form $((\gamma, -1, x), A)$ where in each pair, $\gamma \leq \beta$, x is a singleton or near-litter, and A is an extended type index with maximum β and minimum γ.

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 $\{((\pi_A(x), A), (\pi_B(y), B)) : ((x, A), (y, B)) \in S\}$.

We say that S is a β-support of X if X is a β-code, S is an β-support, and for any β-allowable permutation π, if $\pi[S] = S$ then $\pi(x) = x$.

Observation (cardinality of the set of supports): It is a useful observation that because μ has cofinality $\geq \kappa$, there are no more than μ (and so exactly μ) near-litters, and similarly there are exactly μ supports.

Definition (the types of our structure): We then stipulate that the elements of τ_β for $\beta < \alpha$ have been constructed precisely as the representatives of equivalence classes of type β codes that have supports: such codes are said to be symmetric. It should be evident that typed near-litters are symmetric [they have supports which are decorated versions of their own singletons], and typed singletons of symmetric objects are symmetric [take the support of the singleton element and add the type of the singleton to all the extended type indices in the support], as we assumed above. It should also be clear that τ_β is defined in the same way at every stage with index at least β.

We then define τ_α in the same way as the set of all representatives of equivalence classes of type α-codes that have supports, and will refer to α-codes that have supports as symmetric α-codes.

It should also be evident that (β, γ, G) (for $\beta \leq \alpha$) will always be symmetric if $|G| < \kappa$ [take the union of the γ-supports of elements of G and add α to all the second components of elements of this union]: all small subsets of a type are realized in each higher type.

Applying permutations to objects with support: Quite standard techniques show that if π is an β-allowable permutation and $X \in \tau_\beta$ has β-support S, then $\pi(X)$ has β-support $\pi[S]$. It follows from this that an allowable permutation on β-codes restricts to a permutation on τ_β, as we presumed in the definition of allowable permutation.

Construction of designated orders on the types with technical conditions: 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. This order needs to satisfy technical conditions, which are satisfied by all $\leq_\beta$ with $\beta \leq \alpha$: we provide for reasons to be discussed later that

1. any $(\beta, -1, L)$, L a litter, precedes each $(\beta, -1, \{a\})$ for $a \in L$ in $<_\beta$;
2. any near-litter N which is not a litter is preceded in the order $<_\beta$ by N° and all elements of $N \Delta N^\circ$;
3. designate a β-support for each $x \in \tau_\beta$ which is not a typed atom or typed near-litter: we require that $\iota_*(x)$ strictly exceed $\iota_*(y)$ for each (y, A) in the domain of the designated support of x.

We assume that such designated supports and conditions on the order $\leq_\beta$ are present, as a hypothesis of the recursion, for $\beta < \alpha$, and construct them for α.

This implies that for any x in any τ_β, elements (y, A) of the designated support of $f_{\beta,\gamma}(x)$ have $\iota_*(y) < \iota_*(x)$.

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. Supports are now introduced as well-orderings from the outset.

3.5 Outline of an alternative approach to defining the model elements [under construction, not part of the main development as yet]

note to Lean project: This is (informally) an implementation of the approach Peter has in mind, but it may differ in details.

This section proposes an alternative presentation to the last two sections. The section on atoms, litters, and local cardinals precedes this section immediately if it is used.

We define an α-pretangle as a function t defined at each $\beta < \alpha$ and possibly at -1 (always defined at -1 if $\alpha = 0$), with $t(\beta)$ in each case a set of β-pretangles, and $t(-1)$ (if it exists) a subset of τ_{-1}. The set of α-pretangles will be denoted by τ_α^0.

We will eventually see that $\tau_\alpha \subseteq \tau_\alpha^0$, so model elements at type α will be (selected) α-pretangles.

The membership relation has an easy definition: for $\beta < \alpha$, $x \in \tau_\beta^0$ and $y \in \tau_\alpha^0$ we define $x \in_{TTT} y$ as $x \in y(\beta)$. The price of the ease of definition is that of course for pretangles this relation is not extensional.

We can define structural permutations easily on α-pretangles: an α-structural permutation is a permutation π of τ_α^0 with the property that for each $-1 \leq \beta < \alpha$ there is a β-structural permutation τ_β such that $\pi(x)(\beta) = \tau_\beta$ "$x(\beta)$ (this only holds for $\beta = -1$ if $x(-1)$ is defined, of course). A -1-permutation is a permutation π of τ_{-1} such that for any litter L, π"L is a near-litter.

We then define π_A for π a α-structural permutation and A a subset of $\lambda \cup \{-1\}$ with α as maximum element as π if $A = \{\alpha\}$ and as $(\pi_{A \setminus \{\min(A)\}})_{\min(A)}$ otherwise, just as above but with more generality.

It then seems to be necessary to introduce the inductive hypothesis that τ_β has already been defined for each $\beta < \alpha$, is the domain of a well-ordering $\leq_\beta$ of type μ (this holds for $\beta = -1$ as well; technical conditions on these well-orderings are stated at the end of this section [above in the present incomplete state of things]) and is extensional (for any subset G of τ_γ for $-1 \leq \gamma < \beta$ there is at most one element x of τ_β such that $x(\gamma) = G$ and there is for any near-litter N and $\beta < \alpha$ an $N_\beta \in \tau_\beta$ such that $N_\beta(-1) = N$ [actually the stronger condition is required that maps A'_δ defined below are total on the extensions that are present]. We refer to objects N_β as typed near-litters.

Definition (f maps): We define $\iota_*(x)$ for $x \in \tau_\beta$ as the order type of $\leq_\beta$ restricted to $\{y \in \tau_\beta : y <_\beta x\}$ (note that ι_* is defined for elements of any type).

Where γ is a type index less than α and δ is a proper type index distinct from γ and also less than α, and $x \in \tau_\gamma$, we define $f_{\gamma,\delta}(x)$ as the local

cardinal $[N'(-1)]$ of the value at -1 of the first N' in $<_\delta$ such that $N'(-1)$ is a near-litter and $[N'(-1)] \in \mathbf{X}_{(\gamma,\delta)}$ and $\iota_*(M_\delta) > \iota_*(x)$ for each $M \in [N'(-1)]$ and $[N'(-1)] \neq f_{\gamma,\delta}(y)$ for any $y <_\gamma x$.

Note that this definition does not really depend on α: it will give the same result at any stage of the construction beyond γ and δ.

Definition (alternative extension map): For any β-pretangle x with $x(\gamma) = G \neq \emptyset$ with $\beta \leq \alpha$ a proper type index and δ a proper type index distinct from γ and less than β, and G nonempty, we define $A_\delta(x)$ as the element y of τ_β such that

$$y(\delta) = \{N_\delta : N \in \bigcup(f_{\gamma,\delta}\text{``}G)\}),$$

and we presume that there is such an object y (this will follow from the eventually stated inductive hypotheses about symmetry of τ_β).

We make what we presume more explicit: let δ be a proper type index; we define $A'_\delta(G)$ as $\{N_\delta : N \in \bigcup(f_{\gamma,\delta}\text{``}G)\}$ for any nonempty subset G of τ_γ with $\gamma \neq \delta$, and stipulate that this map is total on extensions in the sense that if G is in the range of some element of τ_β (β dominating both γ and δ) and in the domain of A'_δ then $A'_\delta(G)$ is in the range of some element of τ_β.

Definition (extensional pretangle): We say that x is an extensional β-pretangle iff x is a β-pretangle and there is $\gamma < \beta$, which is equal to -1 if -1 is in the domain of x, such that $x = A_\delta(x)$ for each $\delta \neq \gamma$ in the domain of x, or $\gamma = -1$ and every $x(\delta)$ is empty.

We refer to $x(\gamma)$ as a distinguished extension of x.

Observation: We claim that the distinguished extension of an extensional β-pretangle is unique.

Suppose $x(\gamma)$ is a distinguished extension (and nonempty; uniqueness is obvious in the empty case). Then the minimal value of ι_* on $x(\gamma)$ is dominated by the minimal value of ι_* on each other extension $x(\delta)$ of x because of the order constraints on the f maps: in defining $x(\delta)$, each element u of $x(\gamma)$ is replaced by typed near-litters in type δ whose values under ι_* must dominate $\iota_*(u)$, and so dominate the minimal value of ι_* on $x(\gamma)$.

This is enough to be able to determine γ uniquely given x.

Stipulation: We stipulate that every element of τ_β for $\beta < \alpha$ is a β-extensional pretangle.

Discussion and more precise stipulation: Observe that if $G \subseteq \tau_\gamma$ is the distinguished extension of an element of τ_β, and δ is not γ or -1, $\{N_\delta : N \in \bigcup(f_{\gamma,\delta}\text{``}G)\}$ cannot be the distinguished extension of an element of τ_β, because it is forced to be another extension of the same element.

We sharpen this condition, defining distinguishable extensions: an extension H is distinguishable if it is not of the form $A'_\delta(G) = \{N_\delta : N \in \bigcup(f_{\gamma,\delta}\text{``}G)\}$ for any G, and further it is distinguishable if it is of the form $A'_\delta(A'_\epsilon(G))$ for some distinguishable extension (probably in a different type), and all distinguishable extensions are determined in this way.

It is useful to note that the functions A'_γ (supposed restricted to nonempty extensions) are injective, have disjoint ranges, and they send a set G with a given minimum of its elementwise image under ι_* to a set with a larger such minimum. So the union of the inverses of A'_γ's, which we might call A^{-1}, is a defined partial function and the distinguishable extensions are exactly those with an even number of iterated images under A^{-1} (including zero as an important special case); nothing has an infinite number of iterated images under A^{-1}, so this definition works.

We further stipulate that the distinguished extension of each element of τ_β is distinguishable if it is nonempty.

Observation: Extensionality has been enforced for the interpreted membership of the structure by our stipulations so far.

Preliminary discussion of permutations: We now need to develop the notion of β-allowable permutation for $\beta < \alpha$. The basic idea is that an β-allowable permutation is a β-structural permutation which sends extensional β-pretangles to extensional β-pretangles. Consider the extensional β-pretangle with $x(\gamma) = \{y\}$ (this will be its distinguished extension, obviously). $x(\delta) = \{N_\delta : N \in f_{\gamma,\delta}(y)\}$ defines the non-distinguished extensions. We must have $\pi(x)(\gamma) = \{\pi_\gamma(y)\}$, and we must have $\pi(x)(\delta) = \{N_\delta : N \in f_{\gamma,\delta}(\pi_\gamma(y))\}$ for one reason, and $\pi(x)(\delta) = \pi_\delta\text{``}\{N_\delta : N \in f_{\gamma,\delta}(y)\} = \{((\pi_\delta)_{-1}\text{``}N)_\delta : N \in f_{\gamma,\delta}(y)\}$ for another reason, so we must have $f_{\gamma,\delta}(\pi_\gamma(y)) = \{(\pi_\delta)_{-1}\text{``}N : N \in f_{\gamma,\delta}(y)\} = [(\pi_\delta)_{-1}\text{``}(f_{\gamma,\delta}(y)^\circ)]$.

It should be reasonably clear that this forces any extensional β-pretangle to be sent to an extensional β-pretangle.

Definition (allowable permutation): A β-structural permutation π is β-allowable if it is a β-structural permutation, each π_γ is γ-allowable, and $f_{\gamma,\delta}(\pi_\gamma(y)) = \{(\pi_\delta)_{-1}\text{``}N : N \in f_{\gamma,\delta}(y)\} = [(\pi_\delta)_{-1}\text{``}(f_{\gamma,\delta}(y)^\circ)]$ holds for all appropriate γ, δ.

Remarks: It should be reasonably clear that the elementwise action of an allowable permutation commutes with the maps A'_γ, whence it follows that extensional β-pretangles with distinguishable distinguished extensions are sent to extensional β-pretangles with distinguishable distinguished extensions.

Further note to Lean group: The rest of the development based on this approach should be clear enough. Some notational reconciliation would be needed. Comments encouraged from Peter or the students.

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

Definition: For an extended type index A, let A_1 denote $A \setminus \{\texttt{min}(A)\}$

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

If S is an α-support and $\beta < \alpha$, $S_{(\beta)}$ is defined as the largest support U such that $U^{\{\alpha\}} \subseteq S$ and U is a β-support.

Definition (strong support): A δ-strong support of an object X is a δ-support S of X with certain additional properties.

1. If $((\beta, -1, x), A) \in S$ then x is a singleton or a litter.

2. If $((\beta, -1, \{x\}), A) \in S$, then $((\beta, -1, L), A) <_S ((\beta, -1, \{x\}), A)$, where L is the litter containing x.

3. If $((\beta, -1, L), A) \in S$ and $[L] = f_{\gamma, \beta}(y)$, then there is a [strong] 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 $\leq_S ((\beta, -1, L), A)$.

Definition: We say that a support S' extends a support S if $\texttt{dom}(S) \subseteq \texttt{dom}(S')$: we might change the order in the extension process.

Observation: a support can have all near-litters converted to litters: 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 for which x is a near-litter and not a litter with $((\beta, -1, x^\circ), A)$ and $((\beta, -1, \{y\}), A)$ for each y in the symmetric difference of x and x°.

We can formally define the order S°: the domain of S° consists of all elements $((\beta, -1, x), A)$ for which x is a singleton or a litter [we call these preserved elements of the domain] and elements $((\beta, -1, y), A)$ such that for some x, $((\beta, -1, x), A)$ is in the domain, x is a near-litter and not a litter, and either $y = x^\circ$ or $y = \{z\} \subseteq x \Delta x^\circ$ (we call these new elements of the domain: notice that nothing prevents a new element from also being a preserved element). We refer to $((\beta, -1, x), A)$ as the archetype of $((\beta, -1, y), A)$.

For any code X which is preserved and not new, we define X' as X. For any X which is preserved and new, we define X' as the earlier in the order S of X and its archetype. For any code X which is new and not preserved, we define X' as the archetype of X. We define $X \, S^\circ \, Y$ as $X' \, S \, Y'$ when

29

$X' \neq Y'$. If $X' = Y'$ and Y is a typed litter (there is only one possible value for Y in this case), $X \, S \, Y$, and if X is a typed litter, $Y \, S^\circ \, X$. If $X' = Y'$ and X' and Y' are both typed singletons, use lexicographic order on the codes for the type -1 elements of the third components of X and Y to determine their order in S°.

Observation (any support can be converted to a strong support): Apply the previous result to put a β-support into a form containing only typed singletons and typed near-litters.

Augment the support by adding, wherever $((\delta, -1, L), A) \in S$ and $[L] = f_{\gamma,\delta}(y)$, the set $T^{A\setminus\{\beta\}}$, where T is the designated γ-support of y, to our support (and transform to a support consisting only of typed litters and typed singletons; note that a near-litter in a designated support is replaced by items preceding it in the order).

Then put the support in the order determined first by ι_* applied to first projections, then by any desired order on extended type indices.

The support which results will be strong by conditions we have placed on the f maps and the orders on the types. A litter precedes its elements, and every element of $f_{\gamma,\delta}(y)$ follows all elements of the designated γ-support of y, however decorated with extended type indices.

Observation (extension of index-raised supports): If S is a strong γ-support and $\beta > \gamma$, it is not necessarily the case that $S^{\{\beta\}}$ is a strong β-support. The difficulty is that there may be type γ typed litters which become non-flexible $((\gamma, -1, N)$ with $[N] = f_{\delta,\gamma}(y)$ for some $\delta < \beta$, $y \in \tau_\delta$. However, $S^{\{\beta\}}$ can be extended to a strong support without adding any new elements of type $\geq \beta$: attention to the closure conditions reveals this. Things of type greater than γ may be added, but they are always of type less than β.

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.

Definition (local bijection): An α-local bijection 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 $\pi_A^0(x) = \pi^0(A, x)$ and state the further condition that each map π_A^0 is injective and has domain the same as its range, and that the intersection of the domain of π_A^0 with any litter is small (empty being an important case of small).

Definition (exception of a permutation): 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 $\pi^{-1}(x) \notin (\pi^{-1}``L)^\circ$. For a proper type index α and π an α-allowable permutation and A an extended type index with maximum α, we say that (A, x) is an exception of π iff x is an exception of $\pi_{A \cup \{-1\}}$.

Definition: For each litter L we define a well-ordering $\leq_L$ of order type κ with coordinated strict well-ordering $<_L$, by providing that

$$(-1, \nu, \beta) \leq_{L_\nu} (-1, \nu, \gamma)$$

if and only if $\beta \leq \gamma$.

Definition: An A-flexible local cardinal is a local cardinal which is not in the range of $f_{\gamma, \min(A)}$ for any $\gamma < \min(A_1)$. If A has one element, all local cardinals are A-flexible.

Theorem (freedom of action): For any α-local bijection π^0 and specification for each proper extended type index A with maximum α of a permutation χ_A of the A-flexible local cardinals, there is a uniquely determined α-allowable permutation π such that

1. $\pi_{A \cup \{-1\}}$ extends π_A^0 for each A

2. and $\pi_3(\pi_A(\min(A), -1, L)) \in \chi_A([L])$ holds when A has one element or when $[L]$ is not of the form $f_{\gamma, \min(A)}(y)$ for some $\gamma < \min(A_1)$,

3. and where the elementwise image under $\pi_{A \cup \{-1\}}$ of a litter L is a near-litter N, the restriction of $\pi_{A \cup \{-1\}}$ to $L \setminus \mathrm{dom}(\pi_A^0)$ is the unique bijection from $L \setminus \mathrm{dom}(\pi_A^0)$ to $N^\circ \setminus \mathrm{dom}(\pi_A^0)$ which is strictly increasing with respect to the orders on L and N°,

4. and satisfying a further technical condition: the permutation π obtained from π^0 has no exceptions which are not elements of the domain of π^0 (this is actually a consequence of the previous condition but worth stating separately).

Proof of the Freedom of Action Theorem: 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 assume that the result holds for all $\beta < \alpha$.

When computing the value for $\pi_A(x)$ at any element $((\beta, -1, x), A)$ in the strong support S we assume (ind hyp) that we have already computed $\pi_B(y)$ for each $((\beta, -1, y), B) <_S ((\beta, -1, x), A)$. Moreover, we assume that the value obtained for $\pi_B(y)$ is the same value that would be obtained by computation along any strong support of y.

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

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

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_A^0 to the elements of N° not in the domain of π_A^0 which is strictly increasing in the sense that it sends larger objects in the sense of $<_L$ to larger objects in the sense of $<_{N^\circ}$. Independence of the support used for x follows from independence of the support used for L.

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

If $[L]$ is not of the form $f_{\gamma, \beta}(y)$ for $\gamma < \texttt{min}(A_1)$ and $y \in \tau_\gamma$, we compute $\pi_A((\beta, -1, L))$ as

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

we map L to the near-litter in $\chi_A([L])$ with the exact modifications required by the local bijection. Independence of the support used for L is obvious.

If $[L]$ is of the form $f_{\gamma, \beta}(y)$ for a $\gamma < \texttt{min}(A_1)$ and $y \in \tau_\gamma$ then we proceed just as above but we take the action on $[L]$ from a different source: 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

$$(\beta, -1, \pi_A^0 \text{“} L \cup (f_{\gamma,\beta}(\pi_{A\setminus\{\beta\}\cup\{\gamma\}}(y))^\circ \setminus \pi_A^0 \text{“}(\tau_{-1} \setminus L))),$$

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

We can compute $\pi_{A\setminus\{\beta\}\cup\{\gamma\}}(y)$ along its strong γ-support by inductive hypothesis that we know how to carry out this procedure for $\gamma < \alpha$. The data supplied for this computation has extended type indices A systematically replaced with $A \cap (\gamma + 1)$. In the case of typed litters of type γ in the support, we observe that the index of the relevant χ_A reduces to $\chi_{\{\gamma\}}$ and this does not seem to give us the ability to compute χ_A for local cardinals which are not A-flexible. But we have already by recursive hypothesis computed the values of $\pi_{A\setminus\{\beta\}\cup\{\gamma\}}$ at these litters, and we can simply supply those values for $\chi_{\{\gamma\}}$ as data in the calculation of $(\pi_{A\setminus\{\beta\}\cup\{\gamma\}}(y)$ (and can indifferently assign values for $\chi_{\{\gamma\}}$ in any arbitary way where we have not computed them already; for us to get the right values of $.\pi_{A\setminus\{\beta\}\cup\{\gamma\}}(y)$ it is sufficient to have correct values for it and its derivatives on a support of y).

Independence of the support used for L can be handled thus: suppose we calculate the value for L along two different supports, for earlier items in each of which we have independence of support. Merge the two supports, handling repeated items by preserving the first occurrence. A computation along this support gives a value which must agree with each of the subcomputations because it agrees on a support for y in each case.

To verify that this computes a permutation, note that we can compute its inverse, by the same procedure applied to the inverse data.

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. Setting up the recursive definition of the computation may be nasty.

4.3 Alternative approach to supports and freedom of action (not part of the main line, or not yet)

We define supports differently in this subsection, so there will be some redevelopment, which is for the moment entirely local to this subsection. We use codes as in the main development for elements of the model structure; it could easily be adapted to the pretangle section approach. By "X has code (β, γ, G)" we mean either (in the main development) that X is this code, or in the pretangle development that G is the distinguished extension of X (γ can be determined from G as either the type of the members of G or -1 if G is empty).

We use the notation π_3 for the third projection of a triple to extract extensions from codes. We introduce the notation $\pi_3^*(x)$ for the sole element of $\pi_3(x)$ in the case where $\pi_3(x)$ is a singleton (in particular, a typed atom).

Definition (condition): We define a β-*condition* for $\beta \leq \alpha$ as a set of pairs (x, A), where A is a finite subset of λ with $\max(A) = \beta$ and x has code $(\min(A), -1, B)$, with B either a singleton (so x is a typed atom) or a near-litter (so x is a typed near-litter). Note that A is proper here (it does not contain -1).

Definition (support): We define a β-*support* for $\beta \leq \alpha$ as a small set of β-conditions. It is convenient to stipulate as a further condition on supports that if (x, A) and (y, A) belong to a support and both x and y are typed near-litters, that $\pi_3(x)$ and $\pi_3(y)$ are disjoint.

Definition (action of an allowable permutation on a support): If π is a β-allowable permutation and S is a β-support, we define $\pi[S]$ as

$$\{(\pi_A(x), A) : (x, A) \in S\}.$$

Definition (support of): If X has code (β, γ, G), $\beta \leq \alpha$, and S is a β-support, we say that S is a support of X iff for any allowable permutation π, π will fix X if it is the case that for every $(x, A) \in S$, $\pi_A(x) = x$.

Definition (dependency): We define a relation of dependency D on conditions: $(x, A)D(y, B)$ if either

1. $A = B$ and $|\pi_3(x)| = 1$ and $\pi_3(y)$ is a litter and $\pi_3(x) \subseteq \pi_3(y)$ (equivalently, $\pi_3^*(x) \in \pi_3(y)$) ["elements of a litter depend on the litter"], or

2. $A = B$, $\pi_3(y)$ is a litter, and $\pi_3(x) \neq \pi_3(y)$ is a near-litter and $\pi_3(x)^\circ = \pi_3(y)$ ["near-litters which are not litters depend on the litter they are near"], or

3. for some $\gamma < \min(A_1)$, $(y, (B \cap \min(A)) \cup \{\gamma\})$ is an element of the designated γ-support of $f_{\gamma, \min(A)}^{-1}([\pi_3(x)])$ ["non-flexible near-litters depend on the elements of the designated support of the unique preimage under an f map of their local cardinals"].

34

The bracketed statements are merely suggestive, as the scare quotes should suggest, as they are statements about atoms and near-litters rather than typed atoms and typed near-litters decorated with extended type indices. A near-litter being flexible or not is of course context dependent; but the scare quoted statements are intended to be suggestive of what is going on rather than precise.

Note that if $(x, A)D(y, B)$ we always have $\iota_*(x) > \iota_*(y)$.

Definition (order on a support): If S is a β-support, we define the order $<_S$ on S by $(x, A) <_S (y, B)$ iff the two conditions belong to S as $\iota_*(x) < \iota_*(y)$.

Definition (strong support): We say that a β-support S is strong if whenever $(x, A) \in S$ and $(x, A)D(y, B)$, we have $(y, B) \in S$. Note that $(y, B) <_S (x, A)$ will hold in this situation.

Observations: Note that the definition of strong support, combined with the disjointness condition in the definition of supports, ensures that if (x, A) is in a strong support and $\pi_3(x)$ is a near-litter, $\pi_3(x)$ is a litter. Note further that any support can be extended to a strong support: this can be enforced in ω stages of adding missing items, adding a small collection of new items at each stage.

General Remark: This treatment of supports and strong supports could replace the one given above and is probably simpler. There is some elegance in supports being well-orderings, as it makes it simpler to state the definition of "support of".

Definition (approximation): A β-approximation is a map π^0 from finite subsets of λ with maximum element β such that each $\pi^0(A)$ (which we write π_A^0) is a function with the following properties:

1. The domain and range of π_A^0 are the same.
2. Each domain element x has code of the form $(\mathtt{min}(A), -1, B)$ where B is either a singleton or a near-litter which is a subset of a litter (i.e, (x, A) is a condition and $\pi_3(x) \subseteq \pi_3(x)^\circ$).
3. $\pi_3(x)$ and $\pi_3(\pi_A^0(x))$ have the same cardinality.
4. For each A, the collection $\{\pi_3(x) : x \in \mathtt{dom}(\pi_A^0)\}$ is pairwise disjoint and covers any litter with which its union has large intersection.

We say that π^0 approximates a β-allowable permutation π just in case $\pi_A(x) = \pi_A^0(x)$ whenever the latter is defined.

Definition (flexibility): A typed near-litter x is A-flexible if it is of type $\mathtt{min}(A)$ and $[\pi_3(x)]$ is not in the range of any $f_{\gamma,\mathtt{min}(A)}$ for $\gamma < \mathtt{min}(A_1)$.

Definition (exception, exact approximation): A -1-allowable permutation π has *exception* x if, L being the litter containing x, we have either $\pi(x) \notin (\pi``L)^\circ$ or $\pi^{-1}(x) \notin (\pi^{-1}``L)^\circ$.

A β-approximation π^0 *exactly approximates* a β-allowable permutation π iff π^0 approximates π and for every exception x of a $\pi_{A \cup \{-1\}}$ (A not containing -1) we have x' with code $(\min(A), -1, \{x\})$ in the domain of π_A^0.

Theorem (freedom of action): A β-approximation π^0 wlll exactly approximate some β-allowable permutation π if it satisfies the additional conditions that for any litter L, the set of domain elements of π_A^0 whose third components intersect L is small, and that any domain element x of π_A^0 which is a typed near-litter is A-flexible.

Proof: For purposes of this proof we choose for every pair (L, M) of subsets of litters a bijection $\pi_{L,M}$ from L to M. This is not an application of AC: it can be done concretely using the indexing of elements of litters, but the details are not important.

We also choose a well-ordering of the conditions: choose a well-ordering of finite subsets of λ (we don't care which one) and set $(x, A) \leq (y, B)$ iff either $\iota_*(x) < \iota_*(y)$ or $\iota_*(x) = \iota_*(y)$ and $A \leq B$.

We also choose an extension of π_A^0 to suitable near-litter subsets of all A-flexible typed litters (the conditions dictate what near-litters should be added to the domain, uniquely, since the litters need to be exactly covered by domain elements in a suitable sense); we do this without notational comment, simply assuming that π_A^0 is defined for each A at a near-litter included in each A-flexible litter, which can be arranged harmlessly (for example, one could have π_A^0 act as the identity on the new A-flexible typed near-litters, but we do not require this).

We choose an approximation π^0 satisfying the conditions of the theorem. We carry out a recursive calculation on the order on conditions: we compute each $\pi_A(x)$ for the π exactly approximated by π^0 on the assumption that this has already been carried out for all $(y, B) \leq (x, A)$. We assume in computing the β-allowable permutation π that the theorem is already established for all $\gamma < \beta$.

We indicate how to carry out the calculation at (x, A). Let $\gamma = \min(A)$, If x is a typed atom, (x, A) is preceded by (L, A), where $\pi_3(L)$ is the litter including $\pi_3(x)$, and we have already computed $\pi_A(L)$ by hypothesis of the recursion. Let L^- be the set of all $y \in \pi_3(L)$ such that for no $z \in \mathrm{dom}(\pi_A^0)$ is $y = \pi_3^*(z)$. We define π_A on each $(\gamma, -1, \{z\})$ for $z \in \pi_3(L)$: if $(\gamma, -1, \{z\})$ is in the domain of π_A^0, we define $\pi_A(\gamma, -1, \{z\})$ as $\pi_A^0(\gamma, -1, \{z\})$. Otherwise, we define $\pi_A(\gamma, -1, \{z\})$ as $(\gamma, -1, \{\pi_{L^-, M^-}(z)\})$, where M^- is the set of all $y \in \pi_3(\pi_A(L))$ such that for no $z \in \mathrm{dom}(\pi_A^0)$ is $y = \pi_3^*(z)$. We have indicated how to compute π_A at every typed atom whose third component is included in the third component of L, and so certainly at x.

We exclude (x, A) from consideration when x is a typed near-litter, except when its third component is a subset of a litter. This will not cause any

difficulties: computing every derivative of π at every typed atom and litter subset is sufficient to compute π for every object in the type structure.

If x is A-flexible, $x = (\gamma, -1, N)$ where N is a near-litter. We have an $M \subseteq N^\circ$ at which $\pi_A^0(\gamma, -1, M)$ is defined. For every $z \in N^\circ \setminus M$, $\pi_A^0(\gamma, -1, \{z\})$ is defined. We define a convenient map π_* on N°, sending z to the element of $\pi_3(\pi_A^0(\gamma, -1, \{z\}))$ when this is defined, and otherwise to $\pi_{M,\pi_3(\pi_A^0(\gamma,-1,M))}(z)$. We then define $\pi_A(\gamma, -1, N)$ as $(\gamma, -1, \pi_*``N)$.

We are left with the case where $[x] = f_{\delta,\gamma}(u)$ for some $u \in \tau_\delta$, with $\delta < \min(A_1)$. In this case we know that there is a support T of u such that all elements of $\{(v, C \cup (A \cap \gamma)) : (v, C) \in T\}$ are $\leq (x, A)$ (T being the designated δ-support of u). Thus we have already computed $\pi_{C \cup (A \cap \gamma)}(v)$ for each element of this set by hypothesis of the recursion. Thus for each element v of the δ-support T we have a desired value of a derivative of the δ-permutation $\pi_{(A \cap \gamma) \cup \{\delta\}}$. We extend these values to a δ-approximation: we have a value for each typed atom in T and for each litter in T with atoms in T removed. We need to extend this so that the domain and range of the approximation are the same. To do this, we fill out orbits of typed atoms, the only constraint being that atoms in litters in T must be mapped (by the approximation and its inverse) compatibly with the way the litters are supposed to be mapped. Only countably many atoms for each atom in T need to be added; the constraints clearly can be met. Extending the map on flexible near-litters presents no difficulty. We get a δ-approximation whose domain includes all typed atoms (including the new ones added) and apply the inductive hypothesis to get a δ-permutation which it approximates. We need to verify that it sends non-flexible near-litters in T to the right values. It certainly sends a non-flexible litter in T to a near-litter with the correct local cardinal, because its appropriate derivative sends the preimage of that local cardinal under the appropriate f map to the right value, because its derivatives act correctly on a support of the preimage. Now if it did not send the non-flexible litter to the correct near-litter, this would imply existence of an exception of the appropriate derivative, and all exceptions of the δ-permutation we have constructed which belong to (or do not belong to the litter in T under consideration are recorded in the approximation and mapped consistently with our expectations about where the litter should be mapped. So in fact the action of the δ-approximation we have constructed is the same on T as we expect the action of $\pi_{(A \cap \gamma) \cup \{\delta\}}$, so we can apply the δ-permutation we have constructed to u to get the only possible value of $\pi_{(A \cap \gamma) \cup \{\delta\}}(u)$, and the inverse

[This is not finished, but is also not part of the main line of development]

4.4 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.

Definition (coding functions): 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.

Definition (the specification of a support): A support which is an image of a strong support under an allowable permutation we may call a nice support. For each nice support S we define a combinatorial object S^* which we call its *specification*. We will show below that what it specifies is the orbit in the action of allowable permutations on supports to which it belongs.

For S a support, we define S_ϵ as the element x of its domain such that the restriction of S to $\{y : y <_S x\}$ is of order type ϵ. We define $S_{<\epsilon}$ as $\{y : y <_S S_\epsilon\}$.

The specification S^* is a well-ordering of the same length as S. We describe the elements of its domain.

1. If S_ϵ is $((\beta, -1, \{x\}), A)$, then S_ϵ^* is $(0, \beta, \delta, A)$ such that S_δ is $((\beta, -1, N), A)$, for N a near-litter with x belonging to N. There is exactly one such δ by the definitions of support and strong support.

   One can state an internal condition on specifications that if S_ϵ^* is $(0, \beta, \delta, A)$, then S_δ^* must have first component 2 or 3.

2. If S_ϵ is $((\beta, -1, N), A)$ and N is a near-litter, and either $|A| = 1$ or $[N]$ is not in the range of any $f_{\gamma,\beta}$ for $\gamma < \min(A_1)$, then S_ϵ^* is $(1, \beta, \emptyset, A)$.

3. If S_ϵ is $((\beta, -1, N), A)$ and N is a near-litter, and $[N] = f_{\gamma,\beta}(x)$ for $\gamma < \min(A_1)$, and $x \in \tau_\gamma$ then S_ϵ^* is $(2, \beta, \chi_{x,(S_{<\epsilon})(\gamma)}, A)$: the third component is the coding function with largest possible domain taken from S which yields x.

   One can state an internal condition on specifications that if S_ϵ^* is $(2, \beta, \chi, A)$, where χ is a coding function with outputs of type γ, then domain elements of χ have specification $(S_{<\epsilon}^*)_{(\gamma)}$ (restrict to earlier elements of the specification, then remove the top element from each extended type index and retain only those which then have top element γ).

Observation: On the inductive hypothesis that there are $< \mu$ γ-coding functions with domain containing a strong support for each $\gamma < \alpha$, we observe that there are $< \mu$ specifications of β-supports for $\beta \leq \alpha$.

Lemma: The specification of a nice β-support exactly determines the orbit in the action of β-allowable permutations on supports to which it belongs.

Proof of Lemma: It is straightforward to see that if S is a nice β-support and if π is a β-allowable permutation, that $(\pi[S])^* = S^*$. The relationships between items in the support recorded in the specification are invariant under application of allowable permutations.

It remains to show that if S and T are nice supports, and $S^* = T^*$, there is an allowable permutation ϕ such that $\pi[S] = T$.

We construct π using the Freedom of Action Theorem.

If we have $S_\epsilon = ((\beta, -1, \{x\}), A)$, we will have $T_\epsilon = ((\beta, -1, \{y\}), A)$ for some y, and we will set $\pi_A^0(x) = y$ as part of the construction of the local bijection to be used.

If we have $S_\epsilon = ((\beta, -1, M), A)$ for M a near litter and either $|A| = 1$ or $[M]$ is not in the range of any $f_{\gamma, \beta}$ for $\gamma < \min(A_1)$, then $T_\epsilon = ((\beta, -1, N), A)$ for N a near litter, with analogous properties, and we set $\chi_A([M]) = [N]$ as part of the data for application of the Freedom of Action Theorem.

If we have $S_\epsilon = ((\beta, -1, M), A)$ for M a near litter with $[M] = f_{\gamma, \beta}(x)$, where $\gamma < \min(A_1)$, then S_ϵ^* is $(2, \beta, \chi_{x, (S_{<\epsilon})_{(\gamma)}}, A)$ and T_ϵ^* is $(2, \beta, \chi_{y, (T_{<\epsilon})_{(\gamma)}}, A)$, and they are the same.

If all earlier items in S are mapped by appropriate derivatives of a fixed allowable permutation π_0 to the corresponding items in T, then

$$\pi_0(x) = \pi_0(\chi_{x, (S_{<\epsilon})_{(\gamma)}}((S_{<\epsilon})_{(\gamma)}))$$

$$= \chi_{x, (S_{<\epsilon})_{(\gamma)}}(\pi_0[(S_{<\epsilon})_{(\gamma)}])$$

$$= \chi_{x, (S_{<\epsilon})_{(\gamma)}}((T_{<\epsilon})_{(\gamma)})$$

$$= \chi_{y, (T_{<\epsilon})_{(\gamma)}}((T_{<\epsilon})_{(\gamma)}) = y$$

and this implies that $\pi(M) \Delta N$ is small: we need to augment the local bijection to prevent anomalies, and there is a way to do this.

We want to ensure that elements of $M \setminus M^\circ$ and elements of $M^\circ \setminus M$ are in the domain of π_A^0 and sent to elements of N, and similarly elements of M are chosen to be mapped by π_A^0 to elements of $N \setminus N^\circ$ and $N^\circ \setminus N$. Some additional work must be done. For each new element introduced to the domain of π_A^0, we have the obligation to fill in its complete orbit in π_A^0. The restriction we must obey as we do this is that any element of a near-litter in S must be mapped by π_A^0 to an element of the corresponding near-litter in T and any element of a near-litter in T must be mapped by $(\pi_A^0)^{-1}$ to

an element of the corresponding near-litter in S. Since only countably many new values are needed to fill in each orbit and κ is uncountable, there is no obstruction to doing this. Note that atoms already in the domain of π_A^0 are already constrained to behave in this way. The map eventually constructed by Freedom of Action will send M to N because it maps elements of $M \Delta M^\circ$ to and elements of $N \Delta N^\circ$ to appropriate values individually, and all other values in M must be mapped to values in N (and elements of N mapped from elements of M) because the map constructed by Freedom of Action has no exceptions not in the domain of the local bijection.

So we have completed the description of what we need to do to construct the needed permutation.

Since the specifications precisely determine the orbits in nice supports under allowable permutations, and there are $< \mu$ specifications (on stated hypotheses) there are $< \mu$ such orbits.

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.

Analysis of coding functions for type 0: 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.

Analysis of the general case:

Our inductive hypothesis is that for each $\beta < \alpha$ there are $< \mu$ β-coding functions.

For each $g \in \tau_\gamma$, $\gamma < \alpha$, let S_g be the strong support obtained by closing up the designated support for g following the observation above on why supports can be extended to strong supports.

We specify an object $X \in \tau_\alpha$ and a strong α-support S for X, and develop a recipe for the coding function $\chi_{X,S}$ which can be used to see that there are $< \mu$ α-coding functions (assuming of course that we know that things worked out correctly for $\beta < \alpha$).

$X = (\alpha, \beta, B)$, where B is a subset of τ_β.

For each $b \in \tau_\beta$, we define $\{b\}_\alpha$ as $(\alpha, \beta, \{b\})$. We describe the computation of a strong α-support T_b for $\{b\}_\alpha$ from the strong β support S_b. We compute this by a process of closing up $S_b^{\{\alpha\}}$: the only situation we are concerned about is items $(N, \{\gamma, \alpha\})$ in the support where $[N] = f_{\delta,\gamma}(d)$, $\delta < \alpha$. In this case we insert a suitably tagged version of S_d. The resulting support will be a strong α-support for $\{b\}_\alpha$; note that it will contain no type α near-litters, there being no mechanism which would add them in our closure process. This process terminates because in adding a support S_d we are always adding things whose position in the designated order on their type is before the position of d in its type.

So for each $b \in B$, we have $\{b\}_\alpha = \chi_{\{b\}_\alpha, T_b}(T_b)$ for T_b taken from a family of $< \mu$ coding functions. We further provide that each T_b end extends S (we can do this by starting with the support $S + S_b^{\{\alpha\}}$ obtained by appending S_b to S then removing duplications).

I need to verify the claim that there are $< \mu$ coding functions $\chi_{\{b\}_\alpha, T_b}(T_b)$ in play. It is given that T_b is computed from S_b, but what we actually need is something like the specification of T_b being computable from the specification of S_b. There are $< \mu$ specifications for supports S_b by the inductive hypothesis. Any specification built the way we describe uses an S_b to start, which has one of $< \mu$ possible specifications. Each further refinement involves choosing an S_d (with one of $< \mu$ possible specifications) to insert at stated positions (describable with $< \kappa$ information) to produce a further partial specification. There will be $< \kappa$ such insertions. There are $< \mu$ possible descriptions of such processes of insertion, including descriptions of actually ill-founded processes (of course, an actual construction of a T_b will be well-founded). So there are $< \mu$ possible specifications for supports T_b constructed as above.

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 $\chi_{\{b\}_\alpha, T_b}$. There are $< \mu$ coding functions of this kind, and we have shown above 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 $\bigcup \pi_3(\chi_{\{b\}_\alpha, T_b}(U'))$ for $b \in B$ and U' end extending U. Clearly this definition depends only on the orbit of S and the set of coding functions T_b derived from B as described above. Before we know that this is actually the coding function desired, we will write it as $\chi_{X,S}^*$.

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 $\chi^*_{X,S}(S) = X$, 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 $b = \bigcup \pi_3(\chi_{\{b\}_\alpha, T_b}(T_b))$, and T_b end extends S.

An arbitrary $c \in \chi^*_{X,S}(S)$ is of the form $\bigcup \pi_3(\chi_{\{b\}_\alpha, T_b}(U))$, where U end extends S and of course must be in the orbit of T_b under allowable permutations, so some $\pi_0[T_b] = U$. Now observe that $\pi_0[S] = S$, so $\pi_0(X) = X$, so $(\pi_0)_\beta``B = B$. Further $(\pi_0)_\beta(b) = c$, so in fact $c \in B$ which completes the argument. The assertion $(\pi_0)_\beta(b) = c$ might be thought to require verification: the thing to observe is that $c = \bigcup \pi_3(\chi_{\{b\}_\alpha, T_b}(U)) = \bigcup \pi_3(\pi_0(\chi_{\{b\}_\alpha, T_b}(S)) = \bigcup \pi_3(\pi_0(\{b\}_\alpha)) = \bigcup \pi_3(\pi_0(\alpha, \beta, \{b\})) = \bigcup \pi_3(\alpha, \beta, \{(\pi_0)_\beta(b)\}) = (\pi_0)_\beta(b)$

This completes the proof: any element of a type is determined by a support (of which there are μ) and a coding function whose domain includes a strong support (there are $< \mu$ of these, so a type has no more than μ elements (and obviously has at least μ elements).

Note for the formal verification project: I think the latest revisions are closer to the standard needed for the Lean verification project.

4.5 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 $\in_{TTT}$ of the structure defined above. 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 u of type $s(i)$ with $\pi_{A_{s,i}}(u)$, where $A_{s,i}$ is the set of all s_k for $i \leq k \leq j + 1$ [x being of type $s(j)$, and there being no variables of type higher than $s(j + 1)$]: $\pi_{A_{s,i}}(x)E\pi_{A_{s,i+1}}(y)$ is equivalent to $(\pi_{A_{s,i+1}})_{s(i)}(x)E\pi_{A_{s,i+1}}(y)$, which is equivalent to xEy by the observation above. 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 $\pi_{A_{s,n_i}}$ fixes a_i for each i. But this means that $(s(j+1), s(j), \{x : \phi^s\})$ is symmetric and belongs to type $s(j+1)$: we can merge the supports of the a_i's (with suitable raising of indices) into a single $s(j+1)$-support. Notice that we assumed the predicativity condition that no variable more than one type higher than x appears (in the sense of TST).

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.

It should be noted that the construction given here is in a sense a Frankel-Mostowski construction, though we have no real need to reference the usual FM constructions in ZFA here. Constructions analogous to Frankel-Mostowski constructions can be carried out in TST using permutations of type 0; here we are doing something much more complicated involving many permutations of type -1 which intermesh in precisely the right way. Our explanation of our technique is self-contained, but we do acknowledge this intellectual debt.

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. For the project, a list should be provided here.

4.6 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 $S_{(\beta)}$ (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 γ-support T which extends $(S_{(\beta)})_{(\gamma)}$. Extend $T^{\{\beta\}}$ to a strong β-support T^*: notice that this will not involve adding any new elements of type $\geq \beta$, though elements of type greater than γ may be added.

Our plan is to use freedom of action technology to construct a permutation π^* whose action on S is the identity and whose action on $T^{\{\alpha,\beta\}}$ precisely parallels the action of π on $T^{\{\beta\}}$.

If this is accomplished, then the action of π^* fixes S and so fixes $(\alpha, \beta, \{(\beta, \gamma, \{g\}) : g \in G\})$, while at the same time $(\pi^*_\beta)_\gamma$ agrees with π_γ on G. This implies that $\pi_\gamma(g) \in G$ (and the same argument applies to π^{-1}) so π fixes $\{(\beta, \gamma, \{g\}) : g \in G\}$.

The support $S + (T^*)^{\{\alpha\}}$ obtained by juxtaposition and deletion of all but the first occurrence of repeated items is a strong support. Build a permutation π^* using the freedom of action theorem whose derivatives send each element of S to itself and whose derivative π^*_β sends each atomic item in T^*, or exception of a derivative of π belonging to a litter in T^*, or element of the orbit under π of any items of the last two kinds, to their image under the appropriate derivative of π, and any flexible litter item in T^* to its image under an appropriate derivative of π. Non-flexible litter items in T^* are then sent to their images under the appropriate derivative of π because their local cardinals are handled correctly because the action on a support is handled correctly, and the other conditions ensure that there can be no exceptions of the new permutation which do not correspond to exceptions of π. This permutation has the effects described above, so π fixes $(\beta, \gamma, \{g\})$ as desired.

Note for formal verification project: This is converging to a full description at the level needed for formalization...

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 stratified fact about sets of a bounded standard size which holds in ZFC 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 ZFC 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 stratified result about familiar mathematical objects which is not also a theorem 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). One statement which seems to be about big sets can be seen to hold in our models: the power set of any well-orderable set is well-orderable, and more generally, beth numbers are alephs. We indicate the proofs: a relation which one of our models of TTT thinks is a well-ordering actually is a well-ordering, because the models are countably complete; so a well-ordering with a certain support has all elements of its domain sets with the same support (a permutation whose action fixes a well-ordering has action fixing all elements of its domain), and all subsets of and relations on the domain are sets with the same support (adjusted for type differential), and this applies further to the well-ordering of the subsets of the domain which we find in the metatheory. Applying the same result to sets with well-founded extensional relations on them proves the more general result about beth numbers. This form of choice seems to allow us to use choice freely on any structure one is likely to talk about in the usual set theory. It also proves, for example, that the power set of the set of ordinals (a big set!) is well-ordered.

NF with strong axioms such as the Axiom of Counting (introduced by Rosser in [12], an admirable textbook based on *NF*), the Axiom of Cantorian Sets (introduced in [3]) or my axioms of Small Ordinals and Large Ordinals (introduced in my [5] 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 [6] on strong axioms of infinity in NFU: the results in that paper are not all mine, and I owe a good deal to Solovay in that connection (unpublished conversations and [16]).

That NF has α-models for each standard ordinal α should follow by the same methods Jensen used for NFU in his original paper [9]. 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 [20]) 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 in this paper can be used to show that Specker trees of infinite rank are consistent in bounded Zermelo set theory with atoms or without foundation (this takes a little work, using the way that internal type representations unfold in TTT and a natural interpretation of bounded Zermelo set theory in TST; a tangled web as described above would have range part of a Specker tree of infinite rank). A bit more work definitely gets this result in ZFA, and 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 $\beth_{\omega_1}$ at least. We leave it as an interesting further task, possibly for 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? There are very special assumptions which we made by fiat in building our model of TTT which do not seem at all inevitable in general models of this theory.

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] Hailperin, finite axiomatization

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

[4] 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.

[5] 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.

[6] 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).

[7] Kemeny thesis on strength of TST

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

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

[10] Quine on ordered pairs

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

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

[13] Russell, Principles of Mathematics

[14] Russell and Whitehead, Principia Mathematica

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

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

[17] 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.

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

[19] Tarski, first description of TST

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

[21] Wiener, Norbert, paper on Wiener pair