New Foundations is consistent

M. Randall Holmes

August 18, 2015

Change Notes:

8/18/2015: imported the revision of the "relevant FM interpretation" found in the latest series, which actually describes what happens in every clan in the main construction without oversimplification.

6/24/2015: 3 am debugging wording in various places.

6/23/2015:

7:46 am MST some typos corrected

external permutations seem to be needed to handle the elementarity argument. I have talked through this with Nathan's group and I believe I know what is needed.

corrected action of an injection of $P(\emptyset)$ into $P(A)$, requring the range to be included in $\texttt{clan}(P(A_1))$ and noting the reason.

Installed definition of external permutations and proofs of their properties, and corrected the elementarity proof.

6/22/2015: 3:30 pm there is an omission of important detail in the discussion of elementarity, which I have partly made good here.

2:30 pm Some detailed edits and corrections as I read through this. I have not examined this version so carefully in a long time. It does hold up on examination, but there are minor slips here and there, and there are places where clarification of structure is needed.

9 am Corrected one place where the definition of $B << A$ is misstated: in all occurrences, it is supposed to include $0 \in B$ (though this may be misguided or unnecessary as discussion below points out).

Corrected the definition of formal element by adding a star (under set codes).

Nathan Bowler's reading group felt that images of argument lists under allowable permutations were not necessarily argument lists because of an issue with representatives. On discussion with Nathan, I think the text as it stands is correct. Some additional exposition of the confusing point might be wanted.

adding language about a point noticed by Nathan and his group:

In this note I use the abbreviation $\texttt{clan}[A]$ for $\texttt{clan}(P(A))$ which is used in a later version.

Nathan points out that we can prove in fact that $\mathcal{P}^2(\texttt{clan}[A]) = \tau(A)$ for any clan index A.

The equation to be shown is (if $B << A = B_n$) $|\mathcal{P}^{|B|-|A|+2}(\texttt{clan}[B])| = |\mathcal{P}^2(\texttt{clan}[A])|$.

The lemma he proposes for this use is $|\mathcal{P}^{i+1}(\texttt{clan}[A])| \geq |\mathcal{P}(\texttt{clan}[A_i])|$.

$|\mathcal{P}^2(\texttt{clan}[A])| \geq |\mathcal{P}(P(A))| \geq |\mathcal{P}(\texttt{clan}(A_1))|$ gives the basis.

If $|\mathcal{P}^{i+1}(\texttt{clan}[A])| \geq |\mathcal{P}(\texttt{clan}[A_i])|$ then $|\mathcal{P}^{i+2}(\texttt{clan}[A])| \geq |\mathcal{P}^2(\texttt{clan}[A_i])| \geq |\mathcal{P}(P(A_i))| \geq |\mathcal{P}(\texttt{clan}(A_{i+1})|$. This is a minor tweak on my proof of $|\mathcal{P}^{i+2}(\texttt{clan}[A])| \geq |\mathcal{P}(P(A_i)|$

Now $|\mathcal{P}^{|B|-|A|+2}(\texttt{clan}[B])| = |\mathcal{P}(\mathcal{P}^{|B|-|A|+1}(\texttt{clan}[B]))| \geq |\mathcal{P}(\mathcal{P}(\texttt{clan}[B_{|B|-|A|}]))| = |\mathcal{P}^2(\texttt{clan}[A])|$.

On the other hand $|\mathcal{P}^2(\texttt{clan}(A))| \geq |\mathcal{P}(P(A))| \geq \mathcal{P}(\mathcal{P}^{|B|-|A|+1}(\texttt{clan}[B])| = |\mathcal{P}^{|B|-|A|+2}(\texttt{clan}(B))$

This suggested that the conditions on parent sets should be stated in a way independent of which clans are base clans: define $P(A)$ as including $\texttt{clan}(A_1)$ and as including $\mathcal{P}^2(\texttt{clan}(B))$ for each B such that $B_1 = A$. Still loopy but simpler.

5/23/2015 fixed an ugly typo in the proof re tangled webs.

3/31/2015 changed the name of the ambiguity property of tangled webs to elementarity, to avoid confusion with the type shifting ambiguity property proper to NF. Please note that though this is no longer the main document (that is `ftsversion.pdf`) I have been editing this one because someone was reading it and had useful comments.

3/28/2015 Making some changes to section 5 in response to remarks from a reader. I need to make the same changes to the nearly identical section in the current official document. Further changes to the same section for the same reason 8:20 PM.

11/3/2014 explicitly added notion of input fine type of a parent or set coder to make phrasing of parts of the main recursive definition clearer.

10/17/2014 minor edit to definition of fine type of an argument list motivated by David's question.

10/16/2014 corrected a minor typo pointed out by David.

9/20/2014 6:47 pm More editing. No conceptual changes; corrected some slips. I am inclined once again to turn to other things and give people a chance to read the text as it stands.

9/19/2014 5:45 pm Further edits. The maps σ are of course not small, since $P(\emptyset)$ is of size κ. I think that the situation is once again approaching an equilibrium where I can declare an editing moratorium.

9/19/2014 I believe it is the partial maps induced by small injective maps from coded parent sets $P(\emptyset)$ to $P(A)$ which are crucial for elementarity (and for the computation of equivalence on set codes), so I am providing a full account of these in place of the account of actions of permutations on $P(A)$'s (but I have presented them as maps from $P(\emptyset)$ to $P(A)$ rather than the reverse). I eliminated the account of the action of permutations of coded parent sets, not because it is incorrect but because it no longer appears to be used. And of course these maps will be quite familiar to readers of nfdoc; they are the same substitutions defined there, and they seem to unavoidably play some of the same roles. In particular, the substitutions have not been completely evicted from the main recursive construction because, as before, they are needed to support computation of equivalence of set codes in a well-founded manner.

I am closer to being able to stop editing for a bit.

9/18/2014 4:21 PM Rewrote the description of the actions of permutations of coded parent sets or the partial actions of small injections from one coded parent set to another. The descriptions are denser, and rely on details of parent coding, so I swapped locations of the parent code section and this one so that parent codes come first. The technical effects of this change on the elementary embedding section are installed (they are actually good; things work better with the new treatment of embeddings). The additional restriction on argument lists of set coders turns out not to be needed (which is fortunate because it is incompatible with the more ramified argument list rules we now have). The gap I reported on 9/17 definitely seems to have been fixed, but the recursive construction is now a bit more baroque and requires checking over.

early: fixed the fine typing so that coders are still generally applicable to lists of coarse type $\emptyset$ as before, with type information about items of index $\emptyset$ being ignored. There are definitely going to be a lot of fine points to fix with the more detailed typing now provided.

9/17/2014 10:36 pm repaired discussion of action of permutations on coded parent sets, which has to be changed due to the finer type information supplied about argument codes with index equal to the coarse type. The discussion needs to be improved; this version is just a sketch. Injections from any coded parent set to $P(\emptyset)$ still work to create elementary embeddings, as required for the definition of equivalence of set codes. In general, there will be quite a lot of tidying to do after today's revision to bring things back into consistency. The description of elementary embeddings in the elementarity section needs to be changed to be in line with the new treatment of embeddings.

7:24 just a remark added here. Typing of set coders needs to be added at the same place where typing of parent coders is telegraphed in advance. The finer analysis required by the fix to the latest problem may make my restriction to types in argument lists of set coders impossible, so require a more careful argument for elementarity considerations. None of this seems insuperable, just potentially annoying.

6:38 pm I am working on fixing the problem with the proof that all sets in the FM interpretation which should be codable are codable. The approach I am taking requires me to record dependencies of argument list items of the coarse type on one another where these exist (using set coders, of necessity) which complicates the recursive construction. The gap appears to be repaired, at the cost of more recursion in the construction. I cannot promise an editing moratorium, as the precise material just introduced is of a very finicky nature and may need debugging, but I may return to this desirable state soon.

I **believe** that this is fixed, but I also think that extremely paranoid type checking of the master recursive definition is in order at this point.

9/17/2014 The error in my revision of the old section 12 led me to discover a gap in the argument I gave originally in the old section 12. I do not think it is an unrepairable gap, but for the moment it is a gap; the proof that relevant iterated power sets in the FM interpretation are inhabited only by codable sets is incomplete, because I overlooked an important special situation. I am working on it, but for the moment the proof is not complete. The proof in nfdoc is subtler, and may serve to solve the problem, but I am not going to claim this until I have verified it; some work is required to translate it.

9/16/2014 2:28 pm My intention at this point is to **STOP EDITING**, except in response to comments, for at least a couple of weeks. This doesn't mean I won't correct typos, and I will certainly address material errors if I notice them. But I am going to let YOU look for them. I believe that the paper is now much closer to a production version (which does not mean it is perfect!)

6:12 PM: a caveat. Today's revision of the former chapter 12 was overambitious; it did introduce a mistake which I have to fix. The key word is "introduce"; it wasn't there before and can be corrected.

1:45 pm This version may look different but is not as different as one might think. Mostly I have moved text around. The former sections 9.10,12 are now one section (section 9), with some changes to the old section 12 material. The new sections 9,10 (formerly 9-12) have headings inserted.

9/16/2014 10:31 am Completely rewrote discussion of transformations of argument lists at the end of section 8.

9/15/2014 Brought the FM example more into line with the approach taken in the main proof.

6:35 pm introduced more structured formatting in section 8.

9/12/3014 Changed title to the obvious brief statement. Removed change notes earlier than the major fix to coded parent sets. I have done extensive editing throughout the document today. Notably, I completed the list of references.

At the end of section 7 there is a proof to be added (in no way needed for the final result, but helpful for the exposition).

In section 5, I am planning to change the definitions and proofs to more closely parallel what happens in the main construction. The permutations used ought to be all the allowable ones, and near-litters should be used in the arguments. done, 9/15.

Universal directive to highlight definitions and lemmas/theorems.

The discussion of the effects of padding argument lists with a desired initial segment on set coders needs to be more explicitly computational.

section 12 needs polish. I have a feeling that it might be reasonable to compact sections 9, 10, 12 into one. These have become very brief because the current format makes certain things which once had quite long proofs rather easier to show.

9/11/2014 another pass checking for type errors caused by the change from equivalence classes of parent codes to representatives of equivalence classes (found a few, but the paper is getting to be structured well enough that it was not too laborious). Reintroduced δ_1 as the decoding map on all parent codes, and δ is then its restriction to representative parent codes.

9/10/2014: Significant change: elements of sets $P(A)$ are now representative elements of equivalence classes included the set of parent codes of index A, which is now denoted $P^+(A)$ rather than $\bigcup P(A)$; a parent code (and so a parent of an atom) is now a code representing its equivalence class rather than itself being an equivalence class of codes. I had to do this because I had not noticed that the last change in the nature of argument lists made the equivalence class definition violate well-foundedness. The use of representative elements instead of equivalence classes is a manageable change, and in fact the modification of the text is not that profound (the abstract data type of codes is getting better defended from tweaks in the implementation). There is no change whatever in the structure of the models of type theory being described; this is all bookkeeping.

If any reader identifies type errors in the text caused by overlooking needed corrections due to this modification, I will be grateful. I am sure that some will linger for quite a while!

I discovered this problem in the course of writing a more explicit description of the action on codes and code components with indices extending A of a permutation of $P(A)$ at the end of section 8.4, which is now written.

Contents

1 The Starting Point 7

2 The Hall of Mirrors: the formulation of New Foundations 8

3 Good News and Bad News: well-known results about New Foundations 9

4 Tangled Webs of Cardinals 11

5 A relevant FM construction 13

6 The motivation of the main construction 18

7 Motivation of the coding: an analysis of part of the usual simple model of ZFA without choice 21

8 The main construction 24

8.1 General considerations 24

8.2 Parent and set coders (typing information only) 25

8.3 Argument codes and argument lists 25

8.4 Inductive assumptions and recursive constructions re argument lists 27

8.5 Parent codes 28

8.6 Set codes 30

8.7 The construction of atoms: clans and litters. The denotations of codes and their types 31

8.8 Allowable Permutations and Symmetry: the Target Model 33

9 Coded sets are symmetric; action of permutations on codes; concrete parent sets are unions of clans and iterated power sets of clans in the FM interpretation 36

10 Verification of combinatorial properties of clans stated in the example and motivational sections 39

11 Verification of elementarity properties stated in the motivational section 43

12 Conclusions to be drawn about NF 45

13 References and Index 47

1 The Starting Point

The beginning of our story is at a point which might be regarded as the end of the story of the *Principia Mathematica* of Russell and Whitehead (PM, [18]). This is the system called TST by Thomas Forster (for example, in [2], the best current monograph on the subject of NF), the simple typed theory of sets. This is a first-order theory with sorts indexed by the natural numbers 0,1,2... [and no, this does not imply prior understanding of the natural numbers] and equality and membership as primitive relations. The sorts are traditionally called "types". Atomic identity sentences $x = y$ are well-formed iff x and y are variables of the same type; atomic membership sentences $x \in y$ are well-formed iff the index of the type of y is the successor of the index of the type of x. The axioms are extensionality (objects of each positive type are equal iff they have the same elements) and comprehension (for any formula $\phi(x)$, there is an A such that $x \in A \leftrightarrow \phi(x)$, where A, obviously one type higher than x and unique by extensionality, can be denoted by $\{x : \phi(x)\}$.

We regard this as the end of the story of PM (not necessarily an uncontroversial view) because this is a simple and natural system realizing the aims of PM. The type system of PM is considerably more complicated than that of TST for two reasons: the first reason is that Russell and Whitehead did not know how to implement ordered pairs as sets (Norbert Wiener gave the first implementation in [19], 1914), so PM contains types of n-ary relations for every n with arbitrarily complex heterogeneous input types; the second reason was that Russell and Whitehead restricted themselves initially to predicative comprehension, then made their system impredicative by adding an axiom of reducibility; TST follows Ramsey ([10]) in using fully impredicative comprehension and no axiom of reducibility. We do know that Russell tried to abandon reducibility in the second edition of PM, but we also know that much of the mathematics in PM does not survive this. TST and the system of PM with reducibility are mutually interpretable.

It is usual to add axioms of Infinity and Choice to TST, but we do not regard these as part of the basic definition. TST + Infinity (+ Choice) has the same mathematical strength as Zermelo set theory with separation restricted to bounded formulas (a system sometimes called Mac Lane set theory). This is weaker than Zermelo set theory, but easily strong enough for almost all mathematics outside of set theory. TST without Infinity is weaker than arithmetic, but this is no part of our story.

There is a very natural interpretation of TST in terms of the familiar set theory ZFC: let X_0 be any set and define X_{i+1} as the power set of X_i for each i. Interpret type i variables as ranging over X_i. Interpret equality and membership relations between each appropriate pair of types as suitable restrictions of the usual equality and membership relations. Notice that there is no requirement here that the sets interpreting the types be disjoint: we in fact cannot even ask the question in the language of TST as to whether objects of distinct types are equal.

2 The Hall of Mirrors: the formulation of New Foundations

The next chapter in the story is an observation made by Russell about PM (under the heading of "systematic ambiguity") and made in a much sharper form by Quine about TST. The system is extremely symmetric, in the sense that there is nothing in the system to distinguish the sorts from one another (to be exact, there is nothing to distinguish the type of individuals, which has no given structure, from any other type). In TST, this can be stated in a very elegant way. For any sentence ϕ, let ϕ^+ be a sentence obtained by raising the type of each variable in ϕ by one (without creating any identifications between variables). Since all variables in ϕ are bound, the exact way that the new variables are chosen does not matter. We observe then that if ϕ is provable, so is ϕ^+, and if we define a mathematical construction using a set abstract $\{x : \phi\}$ and our scheme of variable type raising sends x to y, $\{y : \phi^+\}$ will be a precisely analogous mathematical construction one type higher. This is the "hall of mirrors" aspect of TST: for example, it is natural in TST to define the number three as in effect the set of all sets with three elements, following Frege and also PM, but we then get a new version of the number three in each type $i + 2$.

Quine made the stronger suggestion that we should simply identify all the types. This gives the theory which is generally called New Foundations (NF) after the title of the paper [9], 1937, where he made the suggestion. NF is the first order single-sorted theory with equality and membership whose axioms are obtained from those of TST by dropping all distinctions of sort between the variables (without introducing any identifications between variables). That is, the axioms are extensionality (objects with the same elements are the same) and stratified comprehension ($\{x : \phi(x)\}$ exists for ϕ a stratified formula), where we say that ϕ is stratified iff there is a function σ from variables to natural numbers with the property 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 $\sigma(x)+1 = \sigma(y)$. Notice that the function σ (called a stratification of ϕ) is acting on x and y as bits of text, not on their values, so we should perhaps put the variables in quotes (but do not here do this).

It is a persistent criticism of NF that it is a syntactical trick. Of course, as phrased here, it does look that way. It is possible to give a finite axiomatization of NF, which eliminates the notion of stratification from the explicit definition of the theory (though the very first thing one would do with such a formulation of the theory would be to prove stratified comprehension as a meta-theorem). The standard reference is Hailperin, ([1]).

In terms of the interpretation of TST suggested above, this is a very strange proposal. If type $i + 1$ is represented by exactly the same set as type i, it is certainly not represented by the power set of the set representing type i, which is a larger set by Cantor's theorem.

3 Good News and Bad News: well-known results about New Foundations

Specker showed in [15], 1962, that NF is equiconsistent with TST + the Ambiguity Scheme which asserts $\phi \leftrightarrow \phi^+$ for each sentence ϕ, which is not surprising given the motivation of the theory.

Specker also showed, much more surprisingly, in [14], 1953, that Choice is inconsistent with NF, which implies that Infinity is a consequence of NF (as if the universe were finite, every partition, being finite, would have a choice set). [Quine's argument for Infinity in the original NF paper is fallacious]. This shows that NF is stronger than expected, but it also shows that it is very strange, and caused considerable doubt about this theory.

On the good news side of things, Jensen showed in [8], 1969, that NFU (New Foundations with urelements) is consistent. We outline his approach. The idea is to replace TST with TSTU, in which the axiom of extensionality is weakened to apply only to objects with elements, so that each positive type contains at most one element with each nonempty extension, but may contain many elements with empty extension (urelements or atoms). Note that the individuals of type 0 are not atoms, or at least are not considered as atoms: we simply do not ask the question as to what elements they have.

The results of Specker can be extended to show that NFU (New Foundations with the weaker form of extensionality and full stratified comprehension) is equiconsistent with TSTU + Ambiguity.

We now argue, following Jensen, for the consistency of TSTU + Ambiguity. We work in some familiar set theory (we use nothing like the full power of ZFC, but we may suppose that to be our working theory). Let X_0 be a set and define X_{i+1} as the power set of X_i for each i. Let each type i variable in the language of TSTU be interpreted as ranging over elements of X_i; interpret equality and membership in TSTU as equality and membership suitably restricted. Thus far we have actually interpreted TST. Now we throw in a refinement. Let s be a strictly increasing sequence of natural numbers. An alternative interpretation takes variables of type i as ranging over elements of X_{s_i}, takes equality between type i objects as equality suitably restricted, and interprets membership of type i objects in type $i+1$ objects as holding where the type $i+1$ object is an element of X_{s_i+1} which has the type i object as an element; note that this interpretation treats all elements of $X_{s_{i+1}} \setminus X_{s_i+1}$ as urelements (it should be noted that the relation interpreting membership of type i objects in type $i+1$ objects will not necessarily agree with the relations interpreting membership between other successive types). It is straightforward to establish that this gives an interpretation of TSTU for each increasing sequence s. Now choose a finite set Σ of formulas of the language of TSTU mentioning no types with index higher than $n-1$. Define a partition of the n element subsets A of the natural numbers determined by the truth values of the sentences in Σ in interpretations of TSTU determined by maps s which have A as an initial segment of their range (the truth value of sentences in Σ is entirely determined by the first n

elements of the range of the function s used). This is a partition of the n element subsets of $\mathbb{N}$ into finitely many parts (no more than $2^{|\Sigma|}$) and so has an infinite homogeneous set H which is the range of a strictly increasing map h. In the interpretation of TSTU determined by the map h, we have ambiguity for all formulas in Σ. This implies by compactness that full Ambiguity is consistent with TSTU, and by the results of Specker that NFU is consistent.

It is useful to note one could use instead of the sets X_i, sets X_α indexed by elements of any limit ordinal (taking unions at limit indices), whereupon the sequences s would be sequences of ordinals below the limit (still indexed by natural numbers); for example, the stages of the cumulative hierarchy up to any limit level could be used. This is relevant to establishing the consistency of strong extensions of NFU.

It is clear that if Choice holds in our working set theory, Choice will hold in all the approximations to NFU obtained by the method above, and so is consistent with NFU, and that if all X_i's are finite, the approximations of NFU obtained by the method above will not satisfy Infinity, and so NFU does not prove Infinity. If X_0 is infinite, Infinity will hold, of course. NFU by itself is weaker than Peano arithmetic, as it happens. We will not consider theories weaker than NFU + Infinity, which is a quite competent mathematical theory equivalent in strength to TST + Infinity or to Mac Lane set theory.

Other fragments of NF have been shown to be consistent, but the strategy we will follow to show the consistency of full NF follows in its broadest outlines the strategy of Jensen for NFU (though this turns out to be quite hard to do).

4 Tangled Webs of Cardinals

We articulate a hypothesis about cardinals in ZFA (ZFC sans choice and with extensionality weakened to allow atoms) whose consistency with ZFA implies the consistency of NF. Note that we can use Scott's trick ([12]) to define cardinals in this theory; the usual definition using initial ordinals will not work when choice is not assumed. The treatment here is derived from our [4], but the notation is greatly improved.

Let λ be an infinite limit ordinal (it could be taken to be ω for the purposes of merely proving Con(NF) but we aim for more generality).

We define TST_n as the theory obtained by restricting the language of TST to mention only types with index less than n and having as its axioms exactly the axioms of TST which can be expressed in the restricted language. A natural model of TST_n is one in which type i is represented by a set X_i, with X_{i+1} being the power set of X_i for each appropriate i, and the equality and membership relations of the natural model on each suitable pair of types being the obvious restrictions of the usual equality and membership relations. It is a crucially important observation that the first order theory of a natural model of TST_n is completely determined by the cardinality of the set X_0 representing type 0 in the model (it is straightforward to construct an isomorphism between natural models with base sets representing type 0 of the same size).

We define a *tangled web of order* λ as a function τ sending nonempty finite subsets of λ to cardinals with two properties:

naturality: If A has at least two elements, $2^{\tau(A)} = \tau(A \setminus \{\mathtt{min}(A)\})$

elementarity: For each n, if A has at least n elements, the first-order theory of a natural model of TST_n with type 0 having cardinality $\tau(A)$ is completely determined by the smallest n elements of A.

It may not be immediately evident, but the definition of a tangled web of cardinals is precisely motivated by the desire to replicate the consistency proof of Jensen for NF.

We argue that the existence of a tangled web of cardinals implies the consistency of NF. Let Σ be a finite set of sentences of the language of TST mentioning no variable of type $\geq n$. Partition the finite subsets A of λ by considering the truth values of the sentences in Σ in natural models of TST with base type of cardinality $\tau(B)$ where B has at least n elements and the smallest n elements of B are exactly the elements of A. By Ramsey's theorem there is an infinite homogeneous subset H of λ for this partition. Note that for any subset B of H with at least n elements, the theory of a model of TST_n with base set representing type 0 of cardinality $\tau(B)$ will assign the same truth values to sentences in Σ. It follows that for any finite subset B of H with more than n elements, we find that ambiguity holds for all sentences in Σ in a model with base set representing type 0 of size $\tau(B)$: notice that the power set of this set will be of size $\tau(B \setminus \{\mathtt{min}(B)\})$, and the argument of this expression is also a subset of H with at least n elements. Thus the theory of the natural model of TST_n with base type T of size $\tau(B)$ agrees with the model of TST_n with base type $\mathcal{P}(T)$ about the truth value of each sentence in Σ, from which it follows that $\phi \leftrightarrow \phi^+$ holds in the model of TST_{n+1} with base type T for each ϕ in Σ. Thus we find that ambiguity for Σ is consistent with TST_m for all $m > n$, so with TST. Thus full Ambiguity is consistent with TST by compactness, and NF is consistent by the results of Specker.

Our program is thus to construct a model of ZFA with a tangled web of cardinals in it. We do this using the FM technique of constructing models of ZFA with a high degree of symmetry (in particular, generally not satisfying Choice).

5 A relevant FM construction

The phenomena illustrated in this section happen in each "clan" in the construction which follows.

We review the requirements for the Frankel-Mostowski construction (refer to [7]). Any permutation π of the atoms in ZFA can be extended to a class permutation of the universe by the convention $\pi(A) = \pi``A$. A Frankel-Mostowski interpretation is determined by a group G of permutations of the atoms (always considered to be extended to the universe as indicated) and a collection F of subgroups of G (a "normal filter") satisfying the following conditions:

1. $G \in F$,
2. $H \in F \wedge H \subseteq K \rightarrow K \in F$,
3. $H \in F \wedge K \in F \rightarrow H \cap K \in F$,
4. $\pi \in G \wedge H \in F \rightarrow \pi H \pi^{-1} \in F$ (normality condition).
5. Further, the group of permutations in G such that $\pi(a) = a$ should belong to F for each atom a.

We then say that a set A is F-symmetric iff the group of permutations in G which fix A belongs to F. The objects in the domain of the FM interpretation are the atoms and the sets which are hereditarily F-symmetric. The membership relation of the FM interpretation is the restriction of the membership relation of our ambient ZFA to this domain. The grand theorem which we are using but not proving asserts that this class structure satisfies ZFA as well (but generally not choice).

We fix a regular uncountable cardinal κ. We will refer to sets of cardinality less than κ as "small" and all other sets as "large". We suppose that P is a large set (that is, $|P| \geq \kappa$) and there is a collection of atoms $\mathbb{A}$ equinumerous with $P \times \kappa$ (which are not necessarily all of the atoms present): let f be a bijection from $P \times \kappa$ onto $\mathbb{A}$ and denote $f(p, \alpha)$ as p_α.

We refer to the set $\{p_\alpha : \alpha < \kappa\}$ as $\texttt{litter}(p)$ and refer to such sets as litters. A near-litter is defined as a subset of $\mathbb{A}$ with small symmetric difference from a litter. p is called the "parent" of $\texttt{litter}(p)$, and of any near-litter with small symmetric difference from $\texttt{litter}(p)$, and of any atom p_α.

We let G be the group of permutations π of the atoms (which may include atoms not in $\mathbb{A}$) which have the property that $\pi(\texttt{litter}(p))\Delta\texttt{litter}(\pi(p))$ is small for each $p \in P$, and which further have the property that the action of G on P is restricted to an unspecified G_0. Another way of putting this is that a permutation is in G iff it moves only a small number of elements of $\mathbb{A}$ in each litter in a way which is not expected from the action of the permutation on P (and actions on P may be further restricted in unspecified ways). The identity is clearly such a permutation, and inverses of such permutations and compositions of such permutations are clearly such permutations. It is worth noting that near-litters are mapped to near-litters, and only a small number of elements of any near-litter are moved in a way not deducible from the way their parent is moved.

Let S be a small set of atoms in $\mathbb{A}$ and near-litters, such that the intersections of distinct near-litter elements of S are large. Let G_S be the collection of permutations in G which fix each element of S (where we apply the rule $\pi(A) = \pi``A$ in the case of the set elements of S). The filter F consists of all subgroups of G which include a G_S as a subgroup.

The only condition on F which requires much effort is the normality condition, and it does not really require much. Suppose that H is an element of F, and so includes some G_S. We claim that $\pi H \pi^{-1}$ includes $G_{\pi(S)}$. Certainly $\pi H \pi^{-1}$ includes $\pi G_S \pi^{-1}$. We claim that $\pi G_S \pi^{-1}$ includes $G_{\pi(S)}$. Suppose that $\sigma \in G_{\pi(S)}$: we would like to show that $\pi^{-1}\sigma\pi \in G_S$: but this is immediate. For $x \in S$ (whether atom or near-litter), $\pi(x) \in \pi(S)$ is fixed by σ, so $\pi^{-1}\sigma\pi(x) = x$ as required.

It is worth noting that we could have required that our supports consist just of atoms and litters, but the use of near-litters simplifies the proof of the normality condition. From a given support, we obtain a support consisting entirely of atoms and litters by replacing each near-litter element N with the litter L with small symmetric difference from N and in addition the atoms in $L \Delta N$ (which are called the anomalous elements for N).

We assume additional conditions on the action of permutations in G:

Existence of a special relation on $\mathbb{A}$: There is a strictly well-founded transitive relation $\leq$ on $P \cup \mathbb{A}$ such that the preimage under $\leq$ of each element p_α of $\mathbb{A}$ is $\{p\}$ and for each $p \in P$, the collection of atoms $\leq p$ and litters $\texttt{litter}(q)$ with $q \leq p$ is a support for p, which we will call its strong support. A strong support of a general object is obtained by taking a support of the object made up of atoms and litters and adding the strong support of each parent of an atom or litter in the original support.

A condition on free action of permutations in G: For any small collection Σ_1 of atoms in $\mathbb{A}$ and litters and small collection Σ_2 of elements of $\mathbb{A}$ such that each element of Σ_1 has strong support not meeting Σ_2, any permutation of Σ_2 can be extended to a permutation in G fixing each element of Σ_1.

observation that these conditions are trivially realizable: Notice that these conditions can be made to hold by making all elements of P pure sets or at any rate requiring that no element of P has an element of $\mathbb{A}$ in its transitive closure: in this case every element of P has empty support and the permutations in G can be taken to be those which move each litter to a near-litter with the same parent, which clearly has the second property.

We demonstrate some properties of the resulting FM interpretation.

All small subsets of the domain of the FM interpretation are sets of the FM interpretation, which support equal to the union of the supports of their elements.

Any subcollection of the domain of the FM interpretation with small symmetric difference from a set of the FM interpretation will be a set of the FM interpretation, with support equal to the union of the supports of the elements of the small symmetric difference and the support of the set from which the difference is taken.

We say that a set C is κ-amorphous iff for any $B \subseteq C$, either B or $C \setminus B$ is small. We say that the cardinal of a κ-amorphous set is a κ-amorphous cardinal.

Every litter $\texttt{litter}(p)$ is a set of the FM interpretation with support its own singleton. Further, every litter is a κ-amorphous set in the FM interpretation. Let C be a subset of $\texttt{litter}(p)$ with strong support S, such that neither C nor $\texttt{litter}(p) \setminus C$ is small. We can then choose two atoms p_α, p_β from the litter, one in C and one not in C, neither belonging to the strong support S. There will be a permutation in G exchanging p_α and p_β and fixing all elements of S. This permutation moves C without moving any element of its support S, which is a contradiction.

Not only is every subset of a litter either small or co-small, but every subset of the collection $\mathbb{A}$ of atoms of interest has small symmetric difference from either a small or a co-small union of litters. We prove this in stages.

Suppose that a subset C of $\mathbb{A}$ cuts a large number of litters (that is, there is a large collection of litters L such that $L \cap C$ and $L \setminus C$ are both nonempty). Suppose further that C has strong support S. Any litter cut by C is cut into a small part and a large part. We can choose a litter cut by C which is not in S and no element of the small part of which belongs to S (because we are only ruling out a small collection of litters); we can then choose from this litter an element of C which is not in S and a non-element of C which is not in S. A permutation in G exchanging these two atoms and fixing all elements of S will exist and will move C but not any element of S, which is a contradiction. This shows that any subset of $\mathbb{A}$ in the FM interpretation has small symmetric difference from a union of litters which is a set.

Suppose that a union of litters C is a set of the FM interpretation including a large collection of litters and excluding a large collection of litters, with a strong support S. We can then choose two litters, one included in C and one not included in C, neither of which is in S, and choose an element from each litter which does not belong to S. A permutation in G which interchanges these two atoms and which fixes each element of S will exist and will move C to a non-union of litters (so certainly move it) and will not move any element of S. This is a contradiction.

If $B \subseteq \mathbb{A}$ is a set of the FM interpretation, we give a precise description of the union of a large collection of litters with small symmetric difference from B. We know that for all but a small collection of litters L, either L is included in B or L is disjoint from B. We also know that for each litter L, exactly one of the sets $L \setminus B$ and $L \cap B$ is large. The union of litters C that we specify is the union of all litters L such that $L \cap B$ is large. The symmetric difference of L and B is the union of the small collection of nonempty $L \cap B$'s for litters L not included as subsets in C and the small collection of nonempty $L \setminus B$'s for L included as a subset in C. This is a small union of small sets, and so is small. Moreover, C

is a set in the FM interpretation, as either the collection of litters included in C or its complement is a small set of litters which can serve as support for C.

We have completed the description of the subsets of $\mathbb{A}$ in the FM interpretation, being exactly those sets with small symmetric difference from the union of a small or co-small collection of litters.

We add a remark which is useful below. Note that a large collection of atoms which is a set of the FM interpretation must have large intersection with some litter. Otherwise, it would have to have small intersection with each of a large collection of litters, and we have seen above that a set of the FM interpretation cannot cut each of a large collection of litters.

We argue that if B and C are subsets of $\mathbb{A}$ which are of the same cardinality in the FM interpretation, $B \Delta C$ is small. We also observe that if B and C are large sets of the FM interpretation with small symmetric difference, it is evident that they are of the same cardinality in the FM interpretation. Suppose that B and C are of the same cardinality in the FM interpretation and $B \Delta C$ is large: this is to be witnessed by a bijection f with strong support S. We may suppose without loss of generality that $C \setminus B$ is large (the case where $B \setminus C$ is large is handled symmetrically). The preimage of $C \setminus B$ is large, and so has large intersection with some litter. The image of the intersection of the preimage of $C \setminus B$ and this litter is large, and so has large intersection with some litter. So we have a large subset of a litter in the preimage of $C \setminus B$ mapped to a large subset of a litter. Choose two elements of the large subset in the preimage, not belonging to S and not mapped to elements of S. A permutation interchanging their images and fixing all elements of S will exist and will move f but not move any element of S, which is a contradiction.

It follows that it is reasonable to define $|\texttt{litter}(p)|$ for any $p \in P$ as the collection of subsets of $\mathbb{A}$ with small symmetric difference from $\texttt{litter}(p)$, as this is precisely the collection of subsets of $\mathbb{A}$ with the same cardinality as the litter (in the internal sense of the FM interpretation).

We have shown that the power set of $\mathbb{A}$ in the FM interpretation is extremely impoverished. In particular, it certainly conveys no set theoretical information about the structure of P in the ground interpretation. We show that $\mathcal{P}^2(\mathbb{A})$, on the other hand, contains a subset the same size as $\mathcal{P}(P)$ in the sense of the FM interpretation (under the further assumption that P remains a set in the FM interpretation, which is again clearly true if P is a pure set and may be true under other conditions). This is unsurprising if we consider the size of this set in the ambient ZFA, but one must note that neither $\mathbb{A}$ nor $\mathcal{P}(\mathbb{A})$ should be expected to contain a set the same size as P in the sense of the FM interpretation.

The crucial result is that the map $(p \in P \mapsto |\texttt{litter}(p)|)$ is a set of the FM interpretation. To see this, observe that any pair $(p, |\texttt{litter}(p)|)$ in this set is actually fixed by any $\pi \in G$, since elements of P are fixed, and sets with small symmetric difference from elements of $\texttt{litter}(p)$ are mapped exactly to sets with small symmetric difference from $\texttt{litter}(p)$. This implies further that the map $(B \subseteq P \mapsto \bigcup\{|\texttt{litter}(p)| : p \in B\})$ is a set: to see that the correct invariance holds, it is useful to recall that the cardinalities of litters are pairwise disjoint sets. The map $(B \subseteq P \mapsto \bigcup\{|\texttt{litter}(p)| : p \in B\})$ is the promised injection from $\mathcal{P}(P)$ into $\mathcal{P}^2(\mathbb{A})$. It is a set in the FM interpretatiion because it is invariant under all permutations in G.

The reason that this construction is interesting is that it shows how to allow structure not visible to the FM interpretation but concealed in type 0 of a model of TST (visible to the ground interpretation) to unfold not in type 1 (completely nondescript here) but in type 2. This technique can be used to cause unexpected structure to unfold at any desired type level in a model of TST, as the reader may discern in the construction that follows.

6 The motivation of the main construction

The properties of the FM construction of the previous section motivate our main construction. The idea is to iterate this FM construction, creating many sets of atoms of this peculiar structure.

In this section we will give an advance description of features of the structure we will build; experience suggests that it is very hard to understand the motivation for the actual construction that follows without detailed understanding of what the target is intended to be like.

We will specify a limit ordinal λ and a regular uncountable cardinal κ. As above, we will refer to sets smaller than κ as small and all other sets as large.

We will construct "coded parent sets" $P(A)$, which are pure sets, for each finite subset A of λ. Details of these sets are given in the main construction. These sets are pairwise disjoint.

We will provide a set of atoms the same size as $\kappa \cup \bigcup_{A \in \mathcal{P}^{<\omega}(\lambda)}(P(A) \times \kappa)$. Where $\mathtt{atom}$ is a fixed bijection from this set to the atoms, we will write an element of $\mathtt{atom}``\kappa$ in the form $\mathtt{atom}(\alpha)$, and write an $\mathtt{atom}(p, \alpha)$ using the notation p_α, for $p \in P(A)$ and $\alpha < \kappa$.

Letters A, B will generally represent finite subsets of λ. A_1 denotes $A \setminus \{\mathtt{min}(A)\}$; A_0 denotes A and A_{i+1} denotes $(A_i)_1$. Obviously A_i is only defined if $|A| \leq i$.

The set $\{p_\alpha : p \in P(A) \wedge \alpha < \kappa\}$ will be denoted by $\mathtt{clan}(P(A))$, and such sets will be called clans. The notation $\mathtt{clan}(A)$, which I am bound to write by mistake now and then, means the same thing.

The set $\{p_\alpha : \alpha < \kappa\}$ will be denoted by $\mathtt{litter}(p)$, and such sets will be called litters. A subset of a clan with small symmetric difference from a litter will be called a near-litter.

There will be a map δ, which we will call the decoding map, whose restriction to each $P(A)$ is an injection, with the property that $\delta``P(\emptyset) = \mathtt{atom}``\kappa$, while for nonempty A,

$$\delta``P(A) = \mathtt{clan}(P(A_1)) \cup \bigcup_{B<<A} \mathcal{P}^{|B|-|A|+1}_{\mathtt{symm}}(\mathtt{clan}(P(B)),$$

where $B << A$ means that B contains 0 and is a proper downward extension of A. $\mathcal{P}_{\mathtt{symm}}$ denotes the power set in the FM interpretation to be described.

Notice that for any nonempty A which has a proper downward extension B containing 0, $P(B)$ will be at least as large as $P(A)$ in the sense of the ground interpretation, because it contains a copy of $\mathtt{clan}(P(B_1))$, which is at least as large as $P(B_1)$, which in turn contains a copy of $\mathtt{clan}(P(B_2))$, and so on: A is equal to some B_i. On the other hand, $P(A)$ is at least as large as $P(B)$ in the sense of the ground interpretation, because it contains a set the same size as $\mathcal{P}^{|B|-|A|+1}_{\mathtt{symm}}(\mathtt{clan}(P(B))$. This implies that these iterated power sets of the FM interpretation must be very impoverished, missing many sets of the ground interpretation.

We refer to p as the (coded) parent of p_α, $\mathtt{litter}(p)$ or any near-litter with small symmetric difference from $\mathtt{litter}(p)$ and $\delta(p)$ as the concrete parent of each of these things. $P(A)$ is the (coded) parent set of $\texttt{clan}(P(A))$, and $\delta``P(A)$ is its concrete parent set. A is the index of $\texttt{clan}(P(A))$.

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

A permutation is said to be *allowable* iff it fixes each clan, fixes each set $\delta``P(A)$, and maps each litter $\texttt{litter}(p)$ to a near-litter with small symmetric difference from $\texttt{litter}(\pi^*(p))$, where $\pi^*(p)$ is the element of $P(A)$ mapped by δ to $\pi(\delta(p))$ (and so maps near-litters to near-litters as well). An atom p_α such that $\pi(p_\alpha) \not\in \texttt{litter}(\pi^*(p))$ is called an exception of π. An allowable permutation has only a small collection of exceptions in each litter.

A support set is a small set of atoms and near-litters. A set C has support S iff every allowable permutation which fixes each element of S also fixes C. A set is said to be symmetric iff it has a support. The FM interpretation in which we are interested will consist of the atoms and the hereditarily symmetric sets.

We will show in the main development that the clans have the property exhibited in the previous example (the double power set lemma), that in the FM interpretation, $\mathcal{P}^2(\texttt{clan}(P(A))$, the double power set of a clan, contains a set the same size as $\mathcal{P}(\delta``P(A))$, the power set of the concrete parent set of the clan.

This is enough to establish that we can coherently define τ in such a way that $\tau(B_i) = |\mathcal{P}^{i+2}(\texttt{clan}(P(B))|$ for each B containing 0 and B_i nonempty, which will enforce the first property of a tangled web (that $\exp(\tau(A)) = \tau(A_1)$).

We verify this. What requires verification is that if $B << A = B_i$ and $C << A = C_j$, $|\mathcal{P}^{i+2}(\texttt{clan}(P(B))| = |\mathcal{P}^{j+2}(\texttt{clan}(P(C))|$.

We prove that $\mathcal{P}^{i+2}(\texttt{clan}(B))$ contains a set the size of $\mathcal{P}(\delta``P(B_i))$. That this is true in the case $i = 0$ is an assumption we have made already. Suppose that $\mathcal{P}^{k+2}(\texttt{clan}(B))$ contains a set the size of $\mathcal{P}(\delta``P(B_k))$ and that B_{k+1} exists. $\delta``P(B_k)$ contains $\texttt{clan}(B_{k+1})$, so $\mathcal{P}^{k+3}(\texttt{clan}(B))$ contains a set the size of $\mathcal{P}^2(\delta``P(B_k))$ which contains a set of size $\mathcal{P}^2(\texttt{clan}(P(B_{k+1}))$ which includes a set the size of $\mathcal{P}(\delta``P(B_{k+1})$, completing the proof by induction of the claim.

It follows that if $B << A = B_i$ and $C << A = C_j$, $\mathcal{P}^{i+2}(\texttt{clan}(P(B))$ contains a set the size of $\mathcal{P}(\delta``P(B_i)) = \mathcal{P}(\delta``P(A))$. Now $\delta``P(A)$ includes $\mathcal{P}^{|C|-|A|+1}(\texttt{clan}(P(C)) = \mathcal{P}^{j+1}(\texttt{clan}(P(C))$, so $\mathcal{P}(\delta``P(A))$ and so $\mathcal{P}^{i+2}(\texttt{clan}(P(B))$ include a set the size of $\mathcal{P}^{j+2}(\texttt{clan}(P(C))$, and because the situation is symmetrical, $\mathcal{P}^{j+2}(\texttt{clan}(P(C))$ contains a set the same size as $\mathcal{P}^{i+2}(\texttt{clan}(P(B))$, so these sets are the same size.

Of course we have as yet established nothing, as we need to show subsequently that we can actually construct a system in which the assumptions we have made in this development are true.

Motivating the elementarity condition for a tangled web: In addition, we want the first order theory of the model of TST$_n$ with base type of size $\tau(B_i) = |\mathcal{P}^{2+i}(\texttt{clan}(P(B)))|$ to depend only on the first n elements of B_i, for any zero-minimal B. The top type of this model is $\mathcal{P}^{n+i+1}(\texttt{clan}(P(B)))$.

Our method of achieving this is to arrange for every element of the model of TST$_{n+2}$ with base type $\texttt{clan}(B)$ and top type $\mathcal{P}^{n+1}(\texttt{clan}(P(B)))$ to be represented by applying one of a suite of constructions f determined entirely by $B \setminus B_n$ to a small amount of data L depending on a small list of elements of $P(B_n)$, in such a way that two such constructions $f[L]$ and $f'[L']$ will have relations of identity or membership between their referents unperturbed by any bijective replacement of the small list of elements of $P(B_n)$ in use in L and L' with different elements of $P(B_n)$ or with elements of a parent set with another index set which would make sense in the context, so the elements of $P(B_n)$ are treated as indiscernibles. This is actually done by making the sets represented highly symmetrical. This has the effect of making the first order theory of the model of TST_{n+2} with base type $\texttt{clan}(B)$ depend only on the first n elements of B (the set $B \setminus B_n$ which determines the set of basic constructions). This is sufficient to show the desired result for a model with base type any $\tau(A)$ (not just a zero-minimal one). Let A be zero-minimal and consider the model of TST_n with base type of size $\tau(A_i)$. The top type of this model is $\mathcal{P}^{n+i+1}(\texttt{clan}(P(A)))$. We arrange as above for this to depend only on the first $n+i$ elements of A. But we can argue that it actually depends only on the first n elements of A_i, because we can use the results above to replace the first i elements of A with any desired finite collection of small enough ordinals, obtaining a B with $B_j = A_i$, without affecting the theory of the types with sizes at or above $\tau(A_i) = \tau(B_j)$, because the cardinalities of these types will not be changed, and as noted above the theory of a natural model of TST_n depends only on n and the cardinality of its base type.

7 Motivation of the coding: an analysis of part of the usual simple model of ZFA without choice

We work in ZFA with a set of atoms $\mathbb{A}$ of unspecified size.

We indicate a representation of sets in the power sets $\mathcal{P}^n(\mathbb{A})$.

The representations will be codes (f, L) which we will write $f[L]$, where L will be an injection from a natural number to $\mathbb{A}$ (viewed as an argument list) and f will code an operation on the argument list.

The coders of rank 0 will be of the form (n, m), where $n < m$ are natural numbers and $(n, m)[L]$ is a code iff the length of L is m.

The coders of rank $i + 1$ will be of the form (E, m), where E is a set of coders of rank i where for each $e \in E$, $\pi_2(e) \geq m$. $(E, m)[L]$ will be a code iff the length of L is m.

We now define the denotation function δ.

$\delta((n, m)[L]) = L(n)$.

$\delta((E, m)[L]) = \{\delta(e[M]) : e \in E \wedge L \subseteq M\}$.

The sets $\delta(f[L])$ in any $\mathcal{P}^n(\mathbb{A})$ are exactly the sets which are hereditarily of finite support with respect to permutations of the atoms.

To see this, first prove by an easy induction that if π is a permutation of the atoms, $\pi(\delta(f[L])) = f[\pi \circ L]$. We leave this to the reader.

Then we prove the main result by induction. There is nothing to prove for the base case of the atoms.

Suppose that all elements of $\mathcal{P}^n(\mathbb{A})$ are in the range of δ. Let D be a subset of $\mathcal{P}^n(\mathbb{A})$ with support S. Let L be an argument list with range S. Now define E as the set of all coders g such that there is an argument list M extending L such that $\delta(g[M]) \in D$. We claim that $\delta((E, m)[L]) = D$, where m is the length of L.

This is straightforward. Every element of D is of the form $\delta(g[M])$ by construction (any codable object has a code with L as an initial segment, as argument lists are easily padded and reordered; all elements of D are codable by inductive hypothesis). Now the only question is whether all elements of $\delta((E, m)[L])$ belong to D. Such an element is of the form $\delta(g[M])$, M extending L, where there is a $g[M']$ with M' also extending L such that $\delta(g[M]) \in D$. But then we can construct a permutation of the atoms extending the map sending each $M'(n)$ to $M(n)$, and the action of this permutation will send $\delta(g[M'])) \in D$ to $\delta(g[M])$, and will fix all elements of the range of L by construction, so it fixes D, so $\delta(g[M]) \in D$.

Further, this can be used to give a proof that the first order theory of the power sets here coded is independent of the cardinality of the set of atoms. Notice that the truth value of a sentence $\delta(f[L]) \in \delta(g[M])$ or $\delta(f[L]) = \delta(g[M])$ depends only on the identities of f and g and the set $\{(i, j) : L(i) = M(j)\}$. This is clear from invariance under permutations.

We make the stronger claim that any such sentence depends only on the structure of the system of codes, not on the size of the set of atoms. Notice that the codes are pure sets and do not depend on any knowledge of the atoms at all. This is clear at type 0 (equality of referents of codes $f[L]$ and $g[M]$ where f and g are projection operators is clearly dependent on the identities of f and g and the set $\{(i,j) : L(i) = M(j)\}$ in a way that is entirely independent of the size of the set of atoms). Further it is clear that if equality of referents of type i codes $f[L]$ and $g[M]$ is computable only from the identities of f and g and the set $\{(i,j) : L(i) = M(j)\}$ in a way that is entirely independent of the size of the set of atoms, it follows that membership of type i codes in type $i+1$ codes is similarly computable: let the type i code be c and the type $i+1$ code be $(E,m)[L]$: we need only determine whether there is a type i code $g[M]$ with M extending L and $g \in E$ which has the same referent as c. Similarly then we can show that equality of two type $i+1$ codes depends only on the structure of the system of codes and not on the size of the set of atoms: that $(E,m)[L]$ and $(E',m')[L']$ have the same reference reduces to showing that for each $g[M]$ with $g \in E$ and M extending L there is a $g'[M']$ which must have the same referent as $g[M]$, with $g' \in E'$ and M' extending L', and vice versa.

Thus the truth value of any sentence just involving constants is only dependent on the system of codes and not the size of the set of atoms. Now we argue that if the truth value of $\phi(c)$ for each code c can be computed independently of the size of the set of atoms, so can that of $(\exists x.\phi(x))$. If $(\exists x.\phi(x))$ is true, then there is a witness, which has a code c, so $\phi(c)$ witnessed its truth already independently of the size of the set of atoms. If $(\exists x.\phi(x))$ is false, then (every object being codable) the fact that each $\phi(c)$ is false witnesses the situation and is entirely determined without reference to the size of the set of atoms.

Compare the coding here with the (much more complicated) set coders in the main construction and view the proof of elementary equivalence of models with sets of atoms of different sizes here as a baby example toward understanding the reasons why the main construction satisfies the elementarity property of tangled webs.

Further expanding on this, we give a similar coding for our first FM example, which shares more features with the set coding in the main construction.

A code will be of the form $f[L]$ where f is a coder and L is an argument list, an injective function from a small ordinal to atoms and near-litters (sets of atoms with small symmetric difference from litters), with distinct near-litters in its range disjoint, and with the further property that if $L(\alpha)$ is an atom, there is an $L(\beta)$ which is a near-litter containing $L(\alpha)$ with $\beta < \alpha$. The type of an argument list L will be a list τ of the same length with $\tau(\alpha) = 0$ if $L(\alpha)$ is a litter and $\tau(\alpha) = (1,\beta)$ if $L(\beta)$ is a near-litter containing $L(\alpha)$.

A type 0 code will be of the form (α,τ), where $\tau(\alpha)$ is of the form $(1,\beta)$, and $(\alpha,\tau)[L]$ is well-formed just in case L is of type τ, and $\delta((\alpha,\tau)[L]) = L(\alpha)$

A type $i+1$ code will be of the form (E,τ), where E is a set of type i codes each of which has second component extending τ (not necessarily properly). $(E,\tau)[L]$ is a code just in case L is of type τ, and $\delta((E,\tau)[L]) = \{\delta(g[M]) : g \in E \wedge L \subseteq M\}$.

The proof that the type i codes represent exactly the sets in the ith iterated power set of the set of atoms which have small support in atoms and near-litters under permutations moving only a small number of atoms goes similarly to the proof above, though it has details which are more complex. (NOTE: to be supplied)

8 The main construction

We now commence the main construction.

Parameters of the construction: We will specify a limit ordinal λ and a regular uncountable cardinal κ.

Definition (small and large sets): As above, we will call sets smaller than κ small sets, and all other sets we will call large sets. Simply for proving Con(NF) it is sufficient for λ to be ω and κ to be ω_1, but more general results can be shown if we give the argument with more generality.

We will begin by defining sets $P(A)$ for each finite subset A of the limit ordinal λ, which will eventually be seen to be parent sets of clans, but we do not introduce any considerations about atoms until the definition of the parent sets is complete (any statements about atoms or sets of atoms are purely to indicate motivation). All elements of the sets $P(A)$ are pure sets. The definition of the sets $P(A)$ is by a complex recursion with other sets.

Convention: Letters A, B will generally represent finite subsets of λ. These will now and then be referred to as *clan indices*.

Definition (operations on clan indices): A_1 denotes $A \setminus \{\mathtt{min}(A)\}$; A_0 denotes A and A_{i+1} denotes $(A_i)_1$. Obviously A_i is only defined if $|A| \leq i$.

We say that B extends A downward when A, B are clan indices and each element of $B \setminus A$ is less than each element of A. We say that B strictly extends A downward iff B extends A downward and $B \neq A$.

We write $B << A$ when B strictly extends A downward and $0 \in B$.

8.1 General considerations

There are three different sorts of codes, parent codes (elements of sets $P^+(A)$; the notation is simply intended to suggest that $P^+(A)$ is a larger set than $P(A)$), argument codes [which are intended to represent atoms and near-litters in the supports of coded objects], and set codes.

Elements of any $P(A)$ are representatives of equivalence classes in $P^+(A)$ under a global equivalence relation $\sim$ on parent codes (we will see that the $P^+(A)$'s are disjoint, so we do not need to index this relation; there will also be natural equivalence relations on the other classes of codes, which we will always be able to identify from context, so we will use $\sim$ symbolically and "equivalent" verbally for all of these relations).

$P(\emptyset) = P^+(\emptyset)$ is the set of small ordinals, and equivalence in this case is equality.

Our parent and set codes (except for the parent codes of empty index) are of the form (f, L), which we write $f[L]$ to suggest representation of function application, where f is a *coder* appropriate to the kind of code (and the index) and L is an argument list, which is an injective function from a small ordinal to argument codes satisfying additional conditions.

8.2 Parent and set coders (typing information only)

Each parent or set coder has an input fine type as a component. Fine types will be explained below.

A notation $f[L]$ is a parent code in $P^+(A)$ if and only if L is an argument list of coarse type A_1, f is a parent coder belonging to $\Pi(\mathtt{min}(A))$ (this set is defined below) and L has as a fine type the input fine type of f. Coarse and fine types of argument lists will be explained below. Further details will be given after argument lists are explained. Notice that to any argument list L of a given coarse type A we can apply any parent coder in a set $\Pi(\alpha)$ where α is less than all elements of A and where L has the input fine type of the coder, obtaining a code in $P^+(A \cup \{\alpha\})$.

A notation $f[L]$ is a set code iff L is an argument list of coarse type A and has as a fine type the input fine type of f and f is a set coder belonging to $\Sigma(B)$, where B is a clan index such that all elements of A are greater than all elements of B and $A \cup B$ is zero-minimal (note that B will either be zero-minimal or else empty in case A is zero-minimal). [$f[L]$ is intended in this case to denote an element of $\mathcal{P}^{|B|+1}(\mathtt{clan}(P(A \cup B)))$ – motivational, no part of the definitions].

8.3 Argument codes and argument lists

Argument codes, argument lists, and the types of argument lists are developed in a series of definitions.

Definition (argument codes): Argument codes are of two sorts, codes for atoms and codes for near-litters.

An *argument code for an atom* of index A is an element of $P(A) \times \kappa$ [the intention, which is no part of the definition, is that (p, α) for $p \in P(A)$ will denote the atom p_α].

An *argument code for a near-litter* with index A is a triple $(C, D, \{1\})$ where $C \in P(A)$ and D is a small subset of $P(A) \times \kappa$. The function of the final label $\{1\}$ is to prevent any argument code for a near-litter from being an argument code for an atom (if we just used (C, D), an argument code for a litter with second component empty would also be an argument code for an atom with ordinal index zero, which would cause endless pointless technical annoyances).

Definition (extension of an argument code): The *extension* of an argument code for a near-litter $(C, D, \{1\})$ is defined as the symmetric difference of $C \times \kappa$ and D [The intention, which is no part of the definition, is that the denotation of a near-litter code will be the same as the set of denotations of the elements of its extension].

Definition ($P^*(A)$): The set of argument codes for near-litters with index A is denoted by $P^*(A)$.

Definition (argument list, coarse type of an argument list): An argument list is an injective function L from a nonzero small ordinal to argument codes with certain properties.

1. For any argument list, $L(0)$ belongs to some $P^*(A)$, where we call A the *coarse type* of the argument list.
2. Every element of the range of L has index which extends A downward (not necessarily properly).
3. Any two distinct elements of the range of L which belong to the same $P^*(B)$ have disjoint extensions.
4. For any α and B such that $L(\alpha) \in P(B) \times \kappa$, there is $\beta < \alpha$ such that $L(\beta) \in P^*(B)$ and $L(\alpha)$ belongs to the extension of $L(\beta)$.
5. For any α and B strictly downward extending A such that $L(\alpha) \in P^*(B)$, there is a parent coder $g \in \Pi(\mathtt{min}(B))$ and a strictly increasing function M from a small ordinal to α such that $g[(\gamma \mapsto L(M(\gamma)))] \sim \pi_1(L(\alpha))$.
6. For any α such that $L(\alpha) \in P^*(A)$, either $\pi_1(L(\alpha))$ is a parent code for an atom (or a small ordinal) or $\pi_1(L(\alpha))$ is a parent code for a set of the form $(f, G, \tau, \mathtt{min}(A))[N]$ where there is a set coder g and strictly increasing function M from a small ordinal to α such that the set code $g[(\gamma \mapsto L(M(\gamma)))]$ is equivalent to the set code $f[(\gamma \mapsto N(G(\gamma)))]$. (To understand details of this clause, look forward to the definition of parent coders for sets).

Definition (fine type of an argument list) : A fine type τ of an argument list L of coarse type A is a function with the same domain as L, which satisfies the following conditions.

1. If $\tau(\alpha) = 0$, $L(\alpha) \in P^*(A)$ and $\pi_1(L(\alpha))$ is a parent code for an atom (or a small ordinal if $A = \emptyset$).
2. If $\tau(\alpha) = (1, \beta)$, then for some B, $L(\alpha) \in P(B) \times \kappa$, $\beta < \alpha$, $L(\beta) \in P^*(B)$, and $L(\alpha)$ belongs to the extension of $L(\beta)$.
3. If $\tau(\alpha) = (2, g, M)$ then M is a strictly increasing function from a small ordinal to α and either
   (a) for some B strictly downward extending A, $L(\alpha) \in P^*(B)$, $g \in \Pi(\mathtt{min}(B))$, and $g[(\gamma \mapsto L(M(\gamma)))]$ is equivalent to $\pi_1(L(\alpha))$,
   (b) or g is a set coder, $L(\alpha) \in P^*(A)$ and $\pi_1(L(\alpha))$ is a small ordinal (in the case $A = \emptyset$) or a parent code for a set of the form $(f, G, \tau, \mathtt{min}(A))[N]$ where $g[(\gamma \mapsto L(M(\gamma)))]$ is equivalent to $f[(\gamma \mapsto N(G(\gamma)))]$ (notice that f and g are set coders in this second case; notice that if $A = \emptyset$ type information about near

litter codes of index $\emptyset$ is ignored). (To understand details of the second case, look forward to the definition of parent coders for sets).

4. Each $\tau(\alpha)$ is of one of these forms.

The definition reveals that every argument list has a fine type. Fine types are not unique. Note that the fine type does not reveal the coarse type. However, it is possible to compute the set $P(B) \times \kappa$ or $P^*(B)$ to which each $L(\alpha)$ belongs using just the coarse type A for L and a fine type τ for L. This is evident where $\tau(\alpha) = 0$ or $\tau(\alpha) = (1, \gamma)$ (assuming that we have been able to determine the set to which each $L(\beta)$ for $\beta < \alpha$ belongs); if $\tau(\alpha) = (2, g, M)$ note that we can determine the set $P^+(B \cup \{\beta\})$ to which $g[(\gamma \mapsto L(M(\gamma)))]$ belongs from the $\Pi(\beta)$ to which g belongs and the coarse type B of $(\gamma \mapsto L(M(\gamma)))$, which we have already computed by inductive hypothesis (or if g is a set coder we know that an $L(\alpha)$ should be in $P^*(A)$).

[The motivation of the definition of argument list and argument list type (which is no part of this definition) is that an argument list is to have a type which dictates which positions are occupied by atoms and which by near-litters, and in particular (type 0) which are occupied by near-litters of the coarse type. Each atom in the list must be preceded by a near-litter which contains it, and the type of the list includes the information as to which position the parent of each atom is found in (types $(1, \beta)$). Each near-litter in the list must be preceded by arguments for a parent code for its concrete parent, and the type includes the parent coder to be used to encode the parent of the near-litter at each position and the positions earlier in the list from which the arguments are to be taken (types $(2, g, M)$).]

Note that extending an argument list cannot change its coarse type.

8.4 Inductive assumptions and recursive constructions re argument lists

Definition: We define the *delta function on argument lists* at a pair of argument lists L, M by

$$\Delta(L, M) = ((\alpha, \beta) \mapsto (|L^\circ(\alpha) \backslash M^\circ(\beta)|, |L^\circ(\alpha) \cap M^\circ(\beta)|, |M^\circ(\beta) \backslash L^\circ(\alpha)|)),$$

where L°, M° are constructed by replacing each argument code for a near-litter code in the range of each list with its extension and each argument code for an atom in the range of each list with its singleton.

Lemma (to be shown by induction on the construction): All the equivalence relations on codes of the form $f[L]$ (either parent codes or set codes) satisfy the *indiscernibility condition* that the truth of an equivalence $f[L] \sim g[M]$ is completely determined, as long as the prerequisite condition is met that the coarse types of L and M are the same, by f, g, and $\Delta(L, M)$. This condition will be verified by induction in the course of the construction. [Notice that this gives enough information to determine whether an coded atom belongs to a coded near litter and whether two coded near-litters have the same parent.]

Action of an injection $P(\emptyset)$ into a coded parent set: Let σ_0 be an injection from $P(\emptyset)$ into $P(A)$.

Define an action (called σ) sending code components of any index B all of whose elements are less than all elements of A to code components of index $A \cup B$.

It is worth noting that if σ is initially taken to be partial, we can define a partial action in the same way.

On $P(\emptyset)$ σ of course agrees with σ_0. On any $P(B) \times \kappa$, $\sigma(p, \alpha) = (\sigma(p), \alpha)$. On any $P^*(B)$, $\sigma(C, D, \{1\}) = (\sigma(C), \sigma``D, \{1\})$. Any set code $f[L]$ of appropriate index is sent to $f[\sigma \circ L]$. Any parent code $f[L]$ for which the coarse type of L properly extends A downward is sent to the representative code equivalent to $f[\sigma \circ L]$.

Notice that if σ is known to be injective on the ranges of L and M, then $f[L] \sim g[M] \leftrightarrow f[\sigma \circ L] \sim g[\sigma \circ M]$: this is the induction step of the argument that σ is injective on argument codes and set codes and maps equivalent parent codes to equal representative parent codes, and is injective on representative parent codes.

The map σ preserves fine type of an argument list of coarse type $\emptyset$, except for possible difficulties with the images in $P^*(A)$ of arguments from $P^*(\emptyset)$ [which can be avoided if one restricts the range of σ_0 to codes representing atoms].

Notice that if we select two argument lists of coarse type L and M and choose σ so that all elements of the ranges of both lists are in its range, we will have $\Delta(L, M) = \Delta(\sigma^{-1}(L), \sigma^{-1}(M))$, where the latter argument lists are of coarse type $\emptyset$.

8.5 Parent codes

We now give the full definition of parent coders and parent codes.

We repeat basic typing information. An $f[L]$ will belong to $P^+(A)$ iff f is a parent coder from the set $\Pi(\mathtt{min}(A))$ to be defined and L is an argument list of coarse type A_1 and has as a fine type the input fine type of the coder f.

Definition (parent coders and codes, equivalence of parent codes): We describe the set $\Pi(\beta)$, where β is an ordinal or takes the special value -1. Elements of $\Pi(\beta)$ are of two kinds, parent coders for atoms and parent coders for sets.

1. A *parent coder for an atom* in $\Pi(\beta)$ is of the form (α, τ, β), where τ is a fine type (the input fine type of the coder), α is in the domain of τ, and $\tau(\alpha)$ is some $(1, \gamma)$ such that $\tau(\gamma) = 0$.

A parent code $(\alpha, \tau, \beta)[L]$ is well-formed iff τ is a fine type of L and the coarse type A of L has all elements greater than β: it will belong to $P^+(A \cup \{\beta\})$. Equivalence $(\alpha, \tau, \beta)[L] \sim (\alpha', \tau', \beta)[L']$ is defined as holding iff $L(\alpha) = L'(\alpha')$. Note that the indiscernibility condition obviously holds in this case.

[The intent here, which is no part of this definition, is that $(\alpha, \tau, \beta)[L]$ denote the atom represented by the argument code $L(\alpha)$ in the obvious way, and that this atom belong to $\texttt{clan}(A)$ if the list is of coarse type A. The code also indicates which parent set it is to be placed in].

2. A *parent coder for a set* in $\Pi(\beta)$ is of the form (f, G, τ, β), where $\beta > 0$ (there are no parent coders for sets in $\Pi(0)$ or $\Pi(-1)$). The component τ is a fine type (the input fine type of the coder). The component G is a strictly increasing function from a small ordinal to the domain of τ. f is a set coder in a $\Sigma(B)$ where all elements of B are less than β. The argument list $(\gamma \mapsto L(G(\gamma)))$ must have coarse type obtained from the coarse type of L by adding the single element β, for any list L of fine type τ (this is a feature of τ not depending on choice of L).

   A parent code $(f, G, \tau, \beta)[L]$ is well-formed iff L is of fine type τ and the coarse type A of L has all elements greater than β: it will belong to $P^+(A \cup \{\beta\})$. Equivalence $(f, G, \tau, \beta)[L] \sim (f', G', \tau', \beta)[L']$ is defined as holding iff

$$f[\alpha \mapsto L(G(\alpha))] \sim f'[\alpha \mapsto L'(G'(\alpha))]$$

   holds. The indiscernibility property holds here if it holds for the set coders involved, which are simpler in a suitable sense.

Parent codes for atoms (parent codes where the coder is a parent coder for an atom) are not equivalent to parent codes for sets (parent codes where the coder is a parent coder for a set).

Observation (nonempty argument lists and disjointness of $P^+(A)$'s): We require that argument lists be nonempty for a technical reason. We can determine the minimal element of the index A such that a code $f[L]$ belongs to $P^+(A)$ from the last component of its coder. We can determine the next largest element of the index, and inductively determine all elements of the index, if there is required to be an argument in the range of L. This ensures that the $P^+(A)$'s are disjoint.

Construction (coded parent sets): For each clan index A, the set $P(A)$ contains exactly one element of each equivalence class under $\sim$ of elements of $P^+(A)$, chosen to be of minimal set theoretical rank (and that the sets $P(A)$ are disjoint follows from the disjointness of the larger sets $P^+(A)$). Notice that representatives can be chosen as soon as equivalence classes are inhabited, as we can compute equivalence between given codes without requiring information about codes not yet constructed: this is one of the functions of the indiscernibility property.

8.6 Set codes

Definition (set codes and coders): We now define set coders and codes.

Definition (set code): A set code is a pair of the form $f[L]$ where the coarse type of L is A, f, a set coder, belongs to a set $\Sigma(B)$ with all elements of B less than all elements of A, and L has as a fine type the input fine type of the coder f. The sets $\Sigma(B)$ are defined below: B will be zero-minimal or empty.

Definition, set coder, non-basis case: An element f of $\Sigma(B)$, B zero-minimal, is a pair of a subset F of $\Sigma(B \setminus \mathtt{max}(B))$ and a fine argument list type τ (the input fine type of the coder) such that each element of F has input fine type extending the type τ^* described in the next paragraph (not necessarily properly).

If one takes any argument list L of fine type τ and coarse type A with all elements of A greater than all elements of B, drops all elements of the range except those of index equal to $A \cup \{\mathtt{max}(B)\}$ or extending $A \cup \{\mathtt{max}(B)\}$ downward, and reindexes, one obtains L^* of type τ^*; this does not depend on the choice of L but is best described this way. We do require for well-formedness that τ^* be nonempty. If B is empty, $L^* = L$.

Definition, set coder, basis case: An element f of $\Sigma(\emptyset)$ is a pair of a subset F of the set $\Pi(-1)$ and a fine type τ (the input fine type of the coder) such that each element of F has input fine type extending τ. An element of $\Pi(-1)$ takes the form $(\alpha, \tau, -1)$ and satisfies the same formal rules for well-formedness and equivalence as any parent code for atoms.

Definition (well-typedness of set codes): We have $(F, \tau)[L]$ well-defined iff L has fine type τ.

Definition (equivalence of set codes): We define $(F, \tau)[L] \sim (F', \tau')[L']$ deviously. This is of course false if the coarse types of L and L' are different. Assuming that these types are the same, choose lists L_0 and L'_0 such that $\Delta(L_0, L'_0) = \Delta(L, L')$, while the coarse types of L_0 and L'_0 are both $\emptyset$ (this uses the delta function on argument lists defined above; the procedure for constructing L_0 and L'_0 is also described above, at the end of section 8.4). We define $(F, \tau)[L] \sim (F', \tau')[L']$ iff each code $g[M]$ with $g \in F$ and M extending L_0^* is equivalent to some $g'[M']$ with $g' \in F'$ and M' extending $(L'_0)^*$, and vice versa. For the base case, remove the stars.

Definition (formal element): A code $g[M]$ is a *formal element* of a code $(F, \tau)[L]$ iff $g \in F \wedge L^* \subseteq M$. It is worth noting that set codes are equivalent iff they have the same formal elements up to equivalence; whether $f[L]$ is a formal element of $g[M]$ is determined, after obvious requirements re coarse type, by f, g, and $\Delta(L, M)$. This is shown as a side effect of the indiscernibility property.

Verification of indiscernibility: The equivalence of a $g[M]$ to some $g'[M']$ is determined (on the assumption that indiscernibility applies) by g, g' and the set of possible values of $\Delta(M, M')$ for well-typed argument lists M, M' extending $L_0^*, (L_0')^*$, which can be seen to depend on nothing other than g, g' and $\Delta(L_0, L_0')$. It follows that the indiscernibility property will hold for $(F, \tau)[L]$ if it holds for all codes with coders belonging to F. At the basis, it holds for coders in $\Pi(-1)$.

The reason for the elaboration is that we need the items in argument lists in the codes $g[M]$ and $g'[M']$ to be simpler in a suitable sense than the terms $(F, \tau)[L], (F', \tau')[L']$ for which equivalence is being computed, which we cannot be sure of if we do not make the change of coarse type. We thus bound the maximum ordinal which can appear in clan indices involved in M; this bound will not drop if the coarse type of L and L' was already $\emptyset$, but it will drop when this computation is repeated for $g[M]$'s themselves (of course if the $g[M]$'s are codes for atoms their equivalence is computable).

The indiscernibility property holds for all codes by a structural induction. That it holds for parent codes for atoms is obvious and serves as the base case; that it holds for parent codes for sets follows from it holding for set codes; that it holds for set codes with argument lists with coarse type nonempty follows from it holding for set codes with argument lists with coarse type $\emptyset$ by the way the computation is defined; that it holds for set codes $f[L]$ with coarse type empty for sets in a power set with a given index follows from it holding for set codes with in an iterated power set of a clan with index one less (and with clan indices involved bounded by the maximum of the index of f). When the index of iterated power sets becomes zero, we are back in the realm of codes for atoms where we are secure.

8.7 The construction of atoms: clans and litters. The denotations of codes and their types

Atoms postulated: We assume the existence of exactly as many atoms as elements of $\kappa \cup ((\bigcup_{A \in \mathcal{P}^{<\omega}(\lambda)} P(A)) \times \kappa)$, with a bijection $\texttt{atom}$ from this set to the atoms.

Definition (notation for atoms): An atom $\texttt{atom}(\alpha)$ for α a small ordinal will just be written so. The other atoms $\texttt{atom}(p, \alpha)$ will be written p_α.

Definition (clans): The collection $\texttt{atom}``(P(A) \times \kappa)$ will be denoted by $\texttt{clan}(P(A))$ or even $\texttt{clan}(A)$. Such sets will be called clans.

Definition (litters, near-litters): The collection $\{p_\alpha : \alpha < \kappa\}$ will be termed $\texttt{litter}(p)$. Such sets are called litters. A subset of a clan with small symmetric difference from a litter will be called a near-litter.

Definition (notions of parenthood): The parent of an atom p_α is p. The parent of a litter $\texttt{litter}(p)$ is p. The parent of a near-litter is the parent of the litter from which it has small symmetric difference. Parents are representatives of equivalence classes of codes under a relation which turns out to be that of having the same referent: we refer to this referent as a concrete parent, as opposed to the parent proper which we might call by contrast the coded parent.

Definition (anomalous elements for a near-litter): The elements of the symmetric difference of a near-litter and the litter with the same parent are called anomalous elements for the near-litter (they are not necessarily elements *of* the near-litter).

Definition (denotation functions): Each parent code or set code has a denotation which belongs to a clan or to an iterated power set of a clan. The function δ_1 returns the denotation of a parent code and the function δ_2 returns the denotation of a set code.

1. $\delta_1(\alpha) = \texttt{atom}(\alpha)$. The concrete parents of elements of $\texttt{clan}(P(\emptyset))$ are elements of $\texttt{atom}``\kappa$.
2. $\delta_1((\alpha, \tau, \gamma)[L]) = p_\beta$, where $(p, \beta) = L(\alpha)$. Note that equivalence of parent codes for atoms clearly corresponds to equality of the coded atoms. This will give an atom in $\texttt{clan}(P(A_1))$ understood as the concrete parent of an atom of $\texttt{clan}(P(A))$, where A_1 is the coarse type of L and γ is the smallest element of A, or an atom in $\texttt{clan}(A)$ belonging to a base type, where A is the coarse type of L and zero-minimal, and γ is -1.
3. $\delta_1((f, G, \tau, \gamma)[L]) = \delta_2(f[\alpha \mapsto L(G(\alpha))])$. Equivalence of parent codes for sets will correspond to equality of sets if set codes have the corresponding property. This will give a set understood as the concrete parent of an atom in $\texttt{clan}(P(A))$, where A_1 is the coarse input type of L and γ is the minimum element of A, belonging to $\mathcal{P}^{|B|+1}(\texttt{clan}(P(A \cup B)))$, where f belongs to $\Sigma(B)$.
4. $\delta_2((F, \tau)[L]) = \{\delta_2(g[M]) : g \in F \wedge L^* \subseteq M\}$, where L^* is defined as above, or $\delta_2((F, \tau)[L]) = \{\delta_1(g[M]) : g \in F \wedge L \subseteq M\}$ in the special case of subsets of base clans. That equivalence of set codes corresponds to equality of sets denoted is evident by induction on membership (and the facts about parent codes of atoms at the base). Where (F, τ) belongs to $\Sigma(B)$ and A is the coarse type of L, the denotation will belong to $\mathcal{P}^{|B|+1}(\texttt{clan}(P(A \cup B)))$.

Observation, symmetry of codable sets: Note that the sets in the range of δ_2 are highly symmetrical.

Observation and Definition (decoding map): It is straightforward to show by induction that equivalent parent codes are mapped to the same value by δ_1. The restriction of δ_1 to the elements of the sets $P(A)$ (which we call just δ) is referred to as the *decoding map*.

Observation re actions on coded parent sets: It is straightforward to verify that the action on referents of codes induced by the action on codes of an injective map from a $P(\emptyset)$ to a $P(A)$, as described above, sends elements of sets to elements of their images under the action (but not all elements of a set which is an image under the action are themselves images under the action) and acts on the parents of atoms leaving the index unaffected.

Definition (notions of parenthood, coded and concrete): For any atom p_α, we refer to p as the coded parent of the atom and $\delta(p)$ as the concrete parent of the atom. Similarly, $\delta_1``P^+(A) = \delta``P(A)$ may be termed the concrete parent set of $\texttt{clan}(P(A))$, and $P(A)$ itself referred to as the coded parent set of the clan (and A as the index of the clan).

Observation: Note that we obtain the relation

$$\delta``P(A) \subseteq \texttt{clan}(P(A_1)) \cup \bigcup_{B<<A} \mathcal{P}^{|B|-|A|+1}(\texttt{clan}(P(B)))$$

as our motivation would suggest we want ($B << A$ meaning that B is a proper downward extension of A which is zero-minimal). The inclusion is proper: only sets in the range of δ_2 are included. We hope to find that the codable sets include all sets in these iterated power sets in the sense of a suitable FM interpretation.

Definition: We now define denotations for argument codes.

1. An argument code for a near-litter $(C, D, \{1\})$ has referent $\delta_3(C, D, \{1\}) = \texttt{litter}(C)\Delta\{p_\alpha : (p, \alpha) \in D\}$.
2. An argument code for an atom (p, β) denotes $\delta_3(p, \beta) = p_\beta$.

8.8 Allowable Permutations and Symmetry: the Target Model

Definition (map on codes induced by a permutation): If π is a permutation of atoms, extended to sets by the usual rule $\pi(D) = \pi``D$, fixing all clans and fixing all concrete parent sets $\delta``P(A)$, we define $\pi^*(x)$, for $x \in P(A)$, as the unique element of $P(A)$ which is mapped by δ to $\pi(\delta(x))$.

Definition (allowable permutation): A permutation of atoms is *allowable* iff it fixes each clan and each concrete parent set $\delta``P(A)$ and sends each litter $\texttt{litter}(p)$ to a near-litter with parent $\pi^*(p)$ (remember that a parent of a litter is a code). Another way of putting this is that atoms in any given clan are mapped to atoms in the same clan, and further that for each p, the collection of atoms p_α such that $\pi(p_\alpha)$ is not in $\texttt{litter}(\pi^*(p))$ (called the set of exceptions of π with parent p) is small. The action of an allowable permutation is extended from atoms to sets by the usual rule $\pi(D) = \pi``D$.

Definition (external permutation): An external permutation relative to a parent set $P(A)$ is a map acting on objects with codes with index downward extending A which fixes each relevant clan and concrete parent set and maps each litter $\texttt{litter}(p)$ to a near-litter with parent $\pi^*(p)$. An external permutation π extends to sets with codes of the relevant index (elements of $\mathcal{P}^{|B|+1}(\texttt{clan}(P(A \cup B))$ where $B << A$) by the usual rule $\pi(D) = \pi``D$.

Definition (support sets, support, symmetry): A support set is a small set of atoms and near-litters. A set D has support S iff S is a support set and any allowable permutation fixing each element of S also fixes D. A set with a support is said to be symmetric. The objects of the FM interpretation are the atoms and the hereditarily symmetric sets. That the allowable permutations satisfy the normality condition needed for the FM condition is straightfoward to establish (see note below on this).

Construction (the FM interpretation specified): The model of ZFA with a tangled web of cardinals is the FM model consisting of the hereditarily symmetric sets with respect to allowable permutations and supports as described.

The system of allowable permutations clearly satisfies the requirements for an FM interpretation other than the normality condition. That $\pi G_S \pi^{-1}$ includes $G_{\pi(S)}$ is readily shown (just as above in the initial FM example), completing the verification that we get an FM model. The use of near-litters rather than litters greatly simplifies these normality proofs.

Observation re values of allowable permutations at support elements: It is useful to note that if two allowable permutations give the same values at all support elements of an object, they have the same value at that object. Consider the effect of applying one of the permutations, then the inverse of the other: this cannot move the object because it moves none of the support elements. Further, note that knowing the value of a permutation at a litter amounts to knowing its value at the parent and its value at each exception or image of an exception which lies in the litter (revealing what is mapped into or out of the litter). This indicates that to know the value of a permutation at an object with a given support, one needs to know no more than a small number of exceptions of the permutation.

Construction (argument list giving a certain support): For any support S, there is an argument list L such that the appropriate action of any allowable permutation fixing all elements of S fixes any $\delta_i(f[L])$. Modify the collection S to an S^* containing only litters proper and satisfying certain closure conditions: S^* will contain each atom in S, each anomalous element for a near-litter in S, and each litter with the same parent as a near-litter in S; choose a coarse type A such that every index of a clan which contains or includes an element of S is a downward extension of A; S^* will contain a litter containing each atom in S^*, and will contain each atom, anomalous element of a litter, or litter with the same parent as an object denoted by an argument code in the argument list of the parent of a litter in S^*, excluding atoms in $\texttt{clan}(P(A_1))$. The collection of argument codes denoting the elements of S^* has the right closure properties to be the range of an argument list L, and clearly any allowable permutation whose action fixes all of the elements of S^* fixes any set with support S.

Construction (equivalent code with a given initial segment of its list): For any fixed argument list L and code $f[M]$, we can construct an equivalent code $f[M']$ such that $L \subseteq M'$. We first indicate how to construct the list M': simply appending M to L may create some conflicts which need to be repaired. Any atom code in M which appears in L can simply be deleted from M as the occurrence in L will meet all requirements. A near-litter code in M whose extension meets the extension of a near-litter code in L should be deleted if it has the same first component as the near-litter code in L and otherwise should be modified so that the elements of the extension shared with the argument code in L are removed. A problem remains if elements of the extension of a deleted near litter code in M are present later in M and do not belong to the extension of any near litter code in L. In this case add a suitable near-litter code (one with the parent taken from these codes for anomalous atoms and with extension modified if necessary to avoid the extension of argument codes in L), and add additional items as necessary to close up the range suitably (as in the previous paragraph, but instead of using codes for litters, use codes for litters modified when necessary to either have disjoint or coincident extension with elements of the range of L). Items prerequisite for new elements of M' which are not found in L are to be added after all elements of L and before old elements of M.

Now observe that for any parent code for an atom, we can construct an equivalent code with the new argument list by changing an index, as all atoms present in M are present in M'. That we can construct a parent code $f'[M']$ for sets equivalent to an $f[M]$ follows from the same claim for a set code. Suppose $f[M]$ is a set code, with $f = (F, \tau)$. We may suppose that we can find a $g'[N']$ with N' extending $(M')^*$ equivalent to each $g[N]$ with N extending M^*, $g \in F$, by induction on the index of the iterated power sets to which coded sets belong; at the basis we are dealing with parent codes for atoms again (we are again constructing a code equivalent to a given code with a desired initial segment to its argument list). The set of coders g' gives the first component of the needed set coder.

9 Coded sets are symmetric; action of permutations on codes; concrete parent sets are unions of clans and iterated power sets of clans in the FM interpretation

Compute allowable (or external) permutations on coded objects argumentwise :

Computation of the image of $\delta_1(f[L])$ or $\delta_2(f[L])$ under π is very simple: replace each argument code $L(\alpha)$ with an argument code denoting $\pi(\delta_3(L(\alpha)))$, then apply the appropriate decoding function. A straightforward induction establishes that a list is sent to a well-formed list with the same type by this operation, and that the result of evaluating the transformed code is $\pi(\delta_i(f[L]))$ (for $i = 1, 2$). This is extremely direct: verify that each of the typing conditions continues to hold, using induction on complexity to verify the requirement that argument codes for near-litters involve suitable parent codes in their first components.

The induced calculations on argument codes are simple. The action on an argument code for an atom is simply obvious: decode the atom, apply the permutation, and encode. For an argument code for a near-litter, it is hardly more difficult: compute the extension, decode it elementwise, apply the permutation elementwise, then note that when encoded elementwise this will be the extension of a uniquely determined argument code for a near-litter, which is what you return. Notice that the only information one actually needs is the action on $P^*(A)$ where A is the coarse type of the list, the action on atoms for which codes appear in the list, and the small collection of cases where the permutation maps atoms into or out of near-litters coded in the list.

Calculations of values of external permutations at codes $f[L]$ with coarse type downward extending A go in the same way, with the special proviso for action of the external permutation on a near-litter with parent in $P(A)$ (or atom with parent in $P(A)$).

Coded sets are symmetric: Further, we see that a coded set $\delta_2(f[L])$ is in all cases symmetric, because it has as its support the image under δ_3 of the range of L: an allowable permutation fixing all elements of this image clearly fixes the denotation of $f[L]$.

Further, note that the range of an argument list considered as a support has strong closure conditions.

Observation and Definition: If L is of coarse type A, the image under δ_3 of its range includes near-litters containing each atom it contains and supports of parents of all near-litters it contains other than parents of near-litters coded in $P^*(A)$. We call a support satisfying these conditions a *strong support* (of index A).

Definition (small permutation of atoms which respects clans): An injective map from atoms to atoms with domain and range the same small set, which sends an atom in any given clan which is in its domain to an atom in the same clan, is a small permutation of atoms which respects clans.

Lemma (Substitution Property, definition of substitution extension): For any small permutation π_0 of atoms which respects clans and permutation χ of $P(\emptyset)$, it is possible to find an allowable permutation π extending π_0 (called a *substitution extension* of π_0 [and χ]) such that all exceptions of π are elements of the field of π_0 and χ coincides with the action of π^* on $P(\emptyset)$.

We indicate how to compute the value of the substitution extension at an atom p_α. If the atom is in the range of π_0 we apply π_0. Otherwise (assuming that we already know how to compute $\pi^*(p)$) we choose any bijection from the elements of $\texttt{litter}(p)$ which are not in the field of π_0 to the elements of $\texttt{litter}(\pi^*(p))$ which are not in the field of π_0 and apply this map (fixing it for all future application to atoms in those near litters).

Computation of the value of π^* at any code p for a parent only requires calculations for elements of the argument list of the code p, some of which are atoms (apply the indicated procedures) and some of which are simpler codes for which we can assume the calculations already completed. What is actually done is to determine the action induced by the permutation on the argument list, then choose the representative of the equivalence class under $\sim$ containing the resulting code. The action on elements $P(\emptyset)$ is determined by χ.

Extend the resulting map on atoms to sets. It will have only the intended exceptions.

Lemma (Substitution Property, for external permutations): For any small permutation π_0 of atoms which respects clans and acts only on clans with index downward extending A, and arbitrary permutation χ of $P(A)$, it is possible to find an external permutation π extending π_0 (called a *substitution extension* of π_0 [and χ]) such that all exceptions of π are elements of the field of π_0 and the action of π^* on $P(A)$ coincides with the action of χ.

Proof: The proof goes exactly as above, except that consideration of $P(A)$ replaces consideration of $P(\emptyset)$. It should be noted that application of an external permutation to a code $f[L]$ with argument list of coarse type A may corrupt the fine type of arguments from $P^*(A)$, but the value of the resulting code can still be computed with the same coder f, because elements depend only on application of coders in the set component of f to L^*, whose fine type is not changed (and of course there is no difficulty if the coarse type of A is a base type and f is a projection operator).

All h.s. sets in appropriate iterated power sets of clans are codable: We say a set or atom is codable iff it is in the range of δ_1 or δ_2.

In this section we show that the intersection of $\delta``P(A)$ and $\mathcal{P}^{|B|-|A|+1}(\texttt{clan}(B))$, for $B << A$, is exactly the set $\mathcal{P}^{|B|-|A|+1}(\texttt{clan}(B))$ of the FM interpretation.

Note that $\mathcal{P}^0(\texttt{clan}(B))$ has all elements codable, since they are atoms.

Note that a symmetric subset of $\mathcal{P}^0(\texttt{clan}(B))$ has small symmetric difference from a small or co-small union of litters, and such sets are clearly codable.

Now suppose that all elements of $\mathcal{P}^m(\texttt{clan}(B))$ in the FM interpretation are codable for m less than or equal to a fixed positive n ($0 < n \leq |B|$). Let C be a symmetric (and so hereditarily symmetric) subset of $\mathcal{P}^n(\texttt{clan}(B))$. We need to show that C is codable. Let S be a strong support of C (the range of an argument list L; any support can be extended to the range of an argument list). Build codes for each element of C using extensions of the argument list L^* obtained by restricting L suitably (to items with index extending B_{n-1} downward, not necessarily properly; one should note that L itself may not have been restricted sufficiently to be an argument list for a code for C; it might contain items not downward extending an element of B_n; further note that we are using the fact that we can construct a code for a given codable object extending any argument list meeting the index restrictions). We claim that the set F of all the coders used in these codes, paired with the fine type τ of L, applied to L, gives a code $(F, \tau)[L]$ for C. That $\delta_2(F, \tau)[L]$ contains all elements of C is evident from the construction. The question is whether it contains anything else. A general element of $\delta_2(F, \tau)[L]$ is of the form $\delta_2(g[M])$ where M extends L^* , where there is an element $\delta_2(g[M']) \in C$ where M' also extends L^*.

It would be sufficient to show that there is an allowable permutation fixing all elements of S which sends $g[M']$ to $g[M]$.

The basic idea is to construct the allowable permutation by matching the lists M' and M (combining this with matching L to L and fixing other elements of the original support which are excluded from the argument list by coarse type consideratiions) and generating a small injective map from which to define a substitution extension. Certainly wherever atom codes are in matching positions in M' and M we expect to map the atom coded at a given position in M' to the atom coded at the same position in M. Where near-litter codes appear in corresponding positions, we rely on matching their argument lists to get correspondences of atoms. Where near-litter codes appear which do not depend on preceding lists of arguments, their parents are atoms and we can suitably extend the injection on which we will base the substitution extension.

The permutation constructed in this way sends $g[M']$ to $g[M]$ and fixes all elements of S, so it also fixes C, so $g[M]$ is in C.

10 Verification of combinatorial properties of clans stated in the example and motivational sections

We verify combinatorial properties of litters and clans analogous to those seen in the initial example of an FM construction. In the headings, "local set" abbreviates "set of the FM interpretation", and "locally" abbreviates "in the FM interpretation".

Litters are local sets: Clearly litters are sets of the FM interpretation with support the same as that of their parents (the support of an atom being its own singleton).

Small sets of local sets are local sets: All small sets of symmetric sets are symmetric sets with support the union of the supports of their elements.

Definition (nearness to a support): We say that a litter is near a support set S if it either has small symmetric difference from a near-litter in S or contains an anomalous element for a near-litter in S. A litter which is not near S cannot contain an element of S.

Litters are locally κ-amorphous: We verify that litters are κ-amorphous in the FM interpretation. Suppose that $\texttt{litter}(p)$ has large disjoint subsets D and E. Choose p_α in D and p_β in E neither of which are in either of the respective strong supports S, T of D, E (understood as decoded ranges of argument lists). Construct an allowable permutation which permutes p_α and p_β, fixes all atoms belonging to S or T and all anomalous elements for near-litters appearing in S or T, and has no other exceptions. If this map moved any element of S or T, the element moved would be a near-litter, and we can choose the one for which a code appears earliest in the relevant argument list, which could only be moved if it contains an exception or image of an exception of the permutation (since codes for elements of a support of its parent appear earlier in the list and so are fixed, and anomalous elements for the near-litter are also fixed), which is impossible: the only candidate exceptions are elements of the supports, which are not moved and so cannot be mapped into or out of a litter, and p_α and p_β: if they are non-anomalous elements of the relevant litter in the support, that element is $\texttt{litter}(p)$ and is thus fixed by the permutation, and p_α and p_β are not exceptions at all.

All the proofs of the combinatorial properties are like this one in following the proofs in the example FM construction above closely, but with help from the substitution property and the properties of strong supports.

Local sets cannot cut a large collection of litters: We verify that a subset D of a clan cannot cut a large collection of litters (E is cut by D iff $E \cup D$ and $E \setminus D$ are both nonempty). Suppose that there is such a set D

with strong support S. Choose a litter included in the clan which is not near S and which is cut by D. Choose an atom from this litter which is in D and an atom from this litter which is not in D. A substitution extension of the map transposing the two atoms and fixing every atom in S and anomalous element for a near-litter of S moves the set D but fixes every element of its support which is impossible. We verify the claim that any element of the support is fixed. Suppose otherwise: then there would be a first element of the support which was not fixed (in an argument list with range S), which has to be a near-litter, and whose concrete parent has to be fixed (since its support elements all appear earlier and so are fixed), so some atom must be mapped either into the associated litter from another litter or out of the associated litter from within it. But the only possible exceptions of this map are either themselves support elements fixed by the map or are mapped to another atom in the same litter.

From this it follows that every symmetric subset of a clan has small symmetric difference from a union of litters which is a symmetric set.

A large union of litters which is a local set is co-small: We verify that a union of a large collection of litters in a clan whose complement relative to the clan is also a large union of litters cannot be symmetric. Suppose otherwise. Then there is such a set D with strong support S. Choose a litter from the clan which is included in D and one which is excluded from D, such that neither of them is near S. The substitution extension of a map transposing two atoms, one chosen from each of these litters, and fixing all atoms in S and anomalous elements for near-litters in S will move D but not move any element of the support of D, for the usual reasons: consider the first element of an argument list with range D which is moved: it must be a near-litter and must contain an exception or the image of an exception; but all exceptions of the map which are moved by the map are chosen so as not to lie in or map into any litter in the support. This is impossible.

Complete description of local subsets of a clan: It follows that every subset of a clan has small symmetric difference from a small or cosmall union of litters. Moreover, all such sets are actually symmetric.

Description of subsets of clans locally equinumerous to litters: We prove that a subset of a clan the same size as a litter in a clan has small symmetric difference from the litter. Suppose otherwise: let C be a subset of a clan and $\texttt{litter}(p)$ be a litter in the same clan and suppose that they are the same size, witnessed by a bijection f with strong support S. Suppose first that the symmetric difference of the two sets is large. If $\texttt{litter}(p) \setminus C$ is large, then $C \cap \texttt{litter}(p)$ is small by amorphousness of litters, so $C \setminus \texttt{litter}(p)$ is large and in bijection with a large subset of $\texttt{litter}(p)$, and this is the only case we need to consider. We can choose two elements of the large subset of $\texttt{litter}(p)$ neither of which are in S, and whose images under the bijection are neither in S nor belong to litters near S (or which are neither one of them in S and belong to the same near-litter, if the large image set happens to be covered by a small collection of near-litters) and then transpose the two elements of $\texttt{litter}(p)$ while fixing the two elements of C and all atomic elements of S. We cannot have moved any element of S, for the same sorts of reasons indicated above, but we have clearly moved f; this is absurd. Since the two sets must have small symmetric difference, there is a symmetric bijection between them with small symmetric difference from the identity (this is where we use the fact that κ is uncountable). It is then easy to see that $\pi(\texttt{litter}(p))$ is $\texttt{litter}(\pi^*(p))$ for any allowable π.

Proof of the double power set lemma: We define $|\texttt{litter}(p)|$ as the collection of subsets of the clan which includes $\texttt{litter}(p)$ with small symmetric difference from $\texttt{litter}(p)$. We observe that it is then quite clear for the same reasons given above that the map taking each $\delta(p)$ to $|\texttt{litter}(p)|$ is symmetric, and so is the map taking an arbitrary subset D of $\delta``(P(A))$ to the union of all cardinals $|\texttt{litter}(p)|$ such that $\delta(p) \in D$. This map is a bijection because the cardinals are disjoint sets. So we are able to establish that $\mathcal{P}^2(\texttt{clan}(P(A)))$ contains a set the same size as $\mathcal{P}(\delta``P(A))$: that is, we have proved the double power set lemma.

Lemma (double power set lemma): For any clan index A, $\mathcal{P}^2(\texttt{clan}(P(A)))$ contains a set the same size as $\mathcal{P}(\delta``P(A))$ (in the FM interpretation).

The argument for the first property of tangled webs is complete: Once this is verified, the argument in the motivational section for coherence of the equation $\tau(A_i) = \mathcal{P}^{2+i}(\texttt{clan}(A))$ for every zero-minimal A is supported. We repeat the relevant text from the motivational section.

This is enough to establish that we can coherently define τ in such a way that $\tau(B_i) = |\mathcal{P}^{i+2}(\texttt{clan}(P(B))|$ for each B containing 0 and B_i nonempty, which will enforce the first property of a tangled web (that $\exp(\tau(A)) = \tau(A_1)$).

We verify this. What requires verification is that if $B << A = B_i$ and $C << A = C_j$, $|\mathcal{P}^{i+2}(\texttt{clan}(P(B))| = |\mathcal{P}^{j+2}(\texttt{clan}(P(C))|$.

We prove that $\mathcal{P}^{i+2}(\texttt{clan}(B))$ contains a set the size of $\mathcal{P}(\delta``P(B_i))$. That this is true in the case $i = 0$ is an assumption we have made already. Suppose that $\mathcal{P}^{k+2}(\texttt{clan}(B))$ contains a set the size of $\mathcal{P}(\delta``P(B_k))$ and that B_{k+1} exists. $\delta``P(B_k)$ contains $\texttt{clan}(B_{k+1})$, so $\mathcal{P}^{k+3}(\texttt{clan}(B))$ contains a set the size of $\mathcal{P}^2(\delta``P(B_k))$ which contains a set of size $\mathcal{P}^2(\texttt{clan}(P(B_{k+1}))$ which includes a set the size of $\mathcal{P}(\delta``P(B_{k+1})$, completing the proof by induction of the claim.

It follows that if $B << A = B_i$ and $C << A = C_j$, $\mathcal{P}^{i+2}(\texttt{clan}(P(B))$ contains a set the size of $\mathcal{P}(\delta``P(B_i)) = \mathcal{P}(\delta``P(A))$. Now $\delta``P(A)$ includes $\mathcal{P}^{|C|-|A|+1}(\texttt{clan}(P(C)) = \mathcal{P}^{j+1}(\texttt{clan}(P(C))$, so $\mathcal{P}(\delta``P(A))$ and so $\mathcal{P}^{i+2}(\texttt{clan}(P(B))$ include a set the size of $\mathcal{P}^{j+2}(\texttt{clan}(P(C))$, and because the situation is symmetrical, $\mathcal{P}^{j+2}(\mathtt{clan}(P(C))$ contains a set the same size as $\mathcal{P}^{i+2}(\mathtt{clan}(P(B))$, so these sets are the same size.

Definition (the claimed tangled web): We define $\tau(B_i) = |\mathcal{P}^{i+2}(\mathtt{clan}(P(B))|$ for each B containing 0 and B_i nonempty. The argument just above shows that this does not depend on the choice of B.

11 Verification of elementarity properties stated in the motivational section

The model of TST_n with base type $\tau(A)$ needs to be shown to have first-order theory depending only on $A \setminus A_n$, the collection of the first n elements of A (where A has at least n elements, of course). It is sufficient to show this for all B zero-minimal which extend A downward: if this is assumed already shown, we will know that the first order theory of the mode of $TST_{n+|B|-|A|}$ with base type $\tau(B)$ is determined by $B \setminus B_{n+|B|-|A|}$, and we can determine from the results above that in fact it is completely determined by the elements of $A \setminus A_n$, because B can be replaced with any other zero-minimal downward extension of A (say C) without any effect on the theory of the part of this model with base type of size $\tau(A)$, as the results above show that the cardinality of the relevant type is not changed by changing B out for C.

Now we assume that B is zero-minimal and our aim is to show that the first order theory of the model of TST_n with base type of size $\tau(B) = |\mathcal{P}^2(\mathtt{clan}(B))|$ (so we can simply take the base type to be $\mathcal{P}^2(\mathtt{clan}(B)))$ is determined by $B \setminus B_n$. The top type of this model is $\mathcal{P}^{n+1}(\mathtt{clan}(B))$. We note that all elements of $\mathcal{P}^{n+1}(\mathtt{clan}(B))$ in the FM model are of the form $\delta_2(f[L])$ for f an element of $\Sigma(B \setminus B_n)$ and L an argument list of coarse type B_n.

We consider the case where $B_n = \emptyset$. Obviously if this case works, so do all others. Choose C such that $C \setminus C_n = B \setminus B_n = B$. Choose an injection σ_0 from $P(B_n) = P(\emptyset)$ into $\mathtt{clan}(C_{n+1}) \subseteq P(C_n)$. This induces a transformation of code components of index D dominated by C_n to code components of index $C_n \cup D$ (which we will call σ); we claim that this action induces an elementary embedding (which we will also call σ) from the model of TST_{n+2} with base type $\mathtt{clan}(B)$ to the model of TST_{n+2} with base type $\mathtt{clan}(C)$.

The fact that this map is an elementary embedding follows from the fact that every object in both natural models of type theory is codable and the indiscernibility property. Each element of any type in the model indexed by B has a representation via the coding which can be converted to the representation of an object in the model indexed by C simply by replacing all elements of $P(B_n) = P(\emptyset)$ injectively with elements of $\mathtt{clan}(C_{n+1})$ using the master injection which defines the embedding (the restriction of range elements in $P(C_n)$ to atoms avoids incurring complex fine type obligations).

Equality and membership statements between objects represented by set codes in the two models (with suitable parent codes standing in for base type atoms) are completely determined by the coders in the two codes and the delta function applied to the arguments; this follows from the indiscernibility property of codes and the way set codes are defined: so the truth values of quantifier-free sentences in the language of set theory about objects in the model indexed by B and corresponding objects in the model indexed by C will be the same.

Suppose that $\phi(x, y)$ satisfies $\phi(x, y) \leftrightarrow \phi(\sigma(x), \sigma(y))$. We want to show that this holds for $(\exists x.\phi(x, y))$.

If $(\exists x.\phi(x, y))$ is true, then $\phi(u, y)$ is true, so $\phi(\sigma(u), \sigma(y))$ is true, so $(\exists x.\phi(x, \sigma(y))$

is true.

Now suppose that $(\exists x.\phi(x, \phi(y))$ holds. Choose a witness u. There is no reason to believe that u is an image under σ. Let u be the referent of a code $f[L]$. Apply an external permutation ρ with no exceptions to u, fixing all elements of the range of σ_0 on which $\sigma(y)$ depends, sending every parent in $P(C_n)$ involved in the structure of L to an element of the range of σ_0. $\rho^* \circ L$ will be an argument list, but not necessarily with the same fine type, as complex fine type information about elements of $P^*(C_n)$ appearing as arguments in L may be suppressed. Let τ^* be the fine type of $\rho^* \circ L$. If $f = (K, \tau)$, then $f^* = (K, \tau^*)$ will provide us with a coder such that $f^*[\rho^* \circ L]$ is well-formed and denotes $\rho(x)$ (this works because all elements of K are applied as coders to elements of L^* not L, and $(\rho \circ L)^*$ has the same fine type as L^*; what happens in the case where element of A are atom coders is simple). Now w coded by $f^*[\rho^* \circ L]$ is an image $\sigma(v)$ by construction. We have $\phi(x, \sigma(y))$ iff $\phi(\rho(x), \rho(\sigma(y)))$ iff $\phi(w, \sigma(y))$ iff $\phi(\sigma(v), \sigma(y)$ iff $\phi(v, y)$, whence we have $(\exists x.\phi(x, y))$.

This is sufficient to see that σ is an elementary embedding as required (no essential complexity is added by handling more arguments, and logical connectives present no difficulties).

At this point, everything asserted in the motivational sections has actually been proved, and the consistency of NF has been proved.

12 Conclusions to be drawn about NF

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

By choosing the parameter λ to be larger (and so to have stronger partition properties) one can show the consistency of a hierarchy of extensions of NF similar to extensions of NFU known to be consistent: one can replicate Jensen's construction of ω- and α-models of NFU to get ω- and α-models of NF (e.g., see how we proved the existence of α-models for the mildly impredicative fragment NFI of NF in [4]). One can show the consistency of NF + Rosser's Axiom of Counting (see [11]), Henson's Axiom of Cantorian Sets (see [3]), or the author's axioms of Small and Large Ordinals (see [5], [6], [13]) in basically the same way as in NFU.

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

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

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

The question of Maurice Boffa as to whether there is an ω-model of TNT (the theory of negative types, that is TST with all integers as types, proposed by Hao Wang ([16])) is settled: an ω-model of NF yields an ω-model of TNT instantly. This work does not answer the question, very interesting to the author, of whether there is a model of TNT in which every set is symmetric under permutations of some lower type.

The question of the possibility of cardinals of infinite Specker rank in ZF is answered, and we see that the existence of such cardinals doesn't require much consistency strength. For those not familiar with this question, the Specker tree of a cardinal is the tree with that cardinal at the top and the children of each node (a cardinal) being its preimages under $\alpha \mapsto 2^\alpha$. It is a theorem of Forster (a corollary of a well known theorem of Sierpinski) that the Specker tree of a cardinal is well-founded (see [2], p. 48), so has an ordinal rank, which we call the Specker rank of the cardinal. NF + Rosser's axiom of counting proves that the Specker rank of the cardinality of the universe is infinite; it was unknown until this point whether the existence of a cardinal of infinite Specker rank was consistent with ZF.

This work does not answer the question as to whether NF proves the existence of infinitely many infinite cardinals (discussed in [2], p. 52). A model with only finitely many infinite cardinals would have to be constructed in a totally different way.

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

13 References and Index

References

[1] Hailperin, T. "A set of axioms for logic". Journal of Symbolic Logic 9 (1944), pp. 1-19.

[2] 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.

[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] Jech, Thomas, *The axiom of choice*, North-Holland 1973, p. 46

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

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

[10] Ramsey, F.P., 1926, "The foundations of mathematics, *Proceedings of the London Mathematical Society*, s225 (1), 338384.

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

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

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

[14] 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.

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

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

[17] Wang, Hao, *Logic, Computers and Sets*, Chelsea 1970, pp. 402-415.

[18] Whitehead, A. N., and Russell B., *Principia Mathematica, second edition*, 3 vols. Cambridge University Press, 1927.

[19] Wiener, N., 1914, "A simplification of the logic of relations, *Proceedings of the Cambridge Philosophical Society*, 17: 387390.

Index

κ, parameter of construction, set officially, 18, 24
κ-amorphous set, 39
κ-amorphous set, defined, 15
λ parameter first mentioned, 11
λ, parameter of construction, set officially, 18, 24

allowable permutation, 19
allowable permutation, officially defined, 34
Ambiguity Scheme, 9
anomalous elements for a near-litter, 32
argument codes, 25
argument list, defined, 26
argument list, first mentioned, 25
atoms, argument codes for, 25
atoms, notation for, defined officially, 31
atoms, official introduction, 31

Choice, axiom of, false in NF, 9
clan, 18
clan index, 24
clan indices, 24
clan, defined officially, 31
coarse type of an argument list, defined, 26
code format, 25
coded parent sets, first mentioned, 18
coder, first mentioned, 25
codes, varieties of, 24

decoding map δ defined, 33
decoding map δ first mentioned, 18
decoding map first mentioned, 18
delta function on argument lists, defined, 27
delta function on argument lists, used, 30
denotation of a parent code, 33
denotation of a set code, 32
denotation of an argument code, 33
double power set lemma, first stated using decoding map, 19
double power set lemma, proved for the main construction, 41

elementarity considerations motivated, 19
exception of an allowable permutation, 34
extension, of an argument code for a near litter, 25

fine type of an argument list, defined, 26
FM construction, main, introduced, 34
FM constructions considered in general, 13
Fraenkel-Mostowski models in general, 13
Frankel-Mostowski interpretation, 13

indiscernibility condition, 27
Infinity, axiom of, a theorem of NF, 9
input fine type of a coder, introduced, 25
input fine types for set coders exhibited, 30
input fine types of parent codes exhibited, 28

large set, 24
large sets, 18
litter, 18
litter, defined officially, 32
litters are κ amorphous, 39

natural model of TST$_n$, 11
near-litter, 18
near-litter defined officially, 32
near-litters, argument codes for, 25
nearness of a litter to a support, 39
NF defined, 8
NFU, consistency of, 9

non-well-foundedness of clan structure, 18
normality condition (in an FM construction), 13, 14
normality condition for main construction, 34

parent (of an atom, litter, near-litter, 19
parent codes and coders introduced, 25
parent codes are equivalence classes, 24
parent set, base case defined, 24
parent set, concrete, 19
permutation, allowable, 19

set codes and coders introduced, 25
set codes, defined, 30
small set, 24
small sets, 18
substitution extension, 37
substitution extension, for external permutations, 37
substitution property, 37
support (set) officially defined, 34
support, support set, 19
symmetry of coded sets, 32

tangled web of cardinals, defined, 11
tangled web of the main construction first mentioned, 19, 41
tangled web, consistency of NF proved from a, 11
TST, 7
TST$_n$, 11
type raising, 8
type theory, simple, 7
type, coarse, of an argument list defined, 26
type, fine, of an argument list, defined, 26