New Foundations is consistent

Sky Wilshaw

4th November 2024

Abstract

We give a self-contained account of a version of the proof of Holmes and Wilshaw [4] that Quine's set theory *New Foundations* [5] is consistent relative to the metatheory ZFC. This version of the proof is written in a style that is particularly amenable to the formalisation in Lean [8]; to that end, type-theoretic concerns and dependencies between parts of the proof are explicitly spelled out.

Contents

Chapter 1

Introduction

1.1 Overview

We will begin by discussing the mathematical background for the question of the consistency of NF. We will establish the mathematical context for the proof we will present. In particular, our proof will not directly show the consistency of NF; instead, we will construct a model of a related theory known as *tangled type theory*, or TTT. We will show that there is a structure that satisfies a particular axiomatisation of TTT which we will discuss in section 1.4. The expected conclusion that NF is consistent then follows from the fact that NF and TTT are equiconsistent [3].

We will now outline our general strategy for the construction of a model of tangled type theory. As we will outline in section 1.4, TTT is a many-sorted theory with types indexed by a limit ordinal λ. In order to impose symmetry conditions on our structure, we will add an additional level of objects below type zero. These will not be a part of the final model we construct. This base type will be comprised of objects called *atoms* (although they are not atoms in the traditional model-theoretic sense). Alongside the construction of the types of our model, we will also construct a group of permutations of each type, called the *allowable permutations*. Such permutations will preserve the structure of the model in a strong sense; for instance, they preserve membership.

The construction of a given type can only be done under certain hypotheses on the construction of lower types. The most restrictive condition that we will need to satisfy is a bound on the size of each type. In order to do this, we will need to show that there are a lot of allowable permutations. The main technical lemma establishing this, called the *freedom of action theorem*, roughly states that a partial function that locally behaves like an allowable permutation can be extended to an allowable permutation. It, and its various corollaries, will be outlined in more detail when we are in a position to prove it.

We can then finish the main induction to build the entire model out of the types we have constructed. This step, while invisible to a human reader in set theory, takes substantial effort to formally establish in a dependent type theory. It then remains to show that this is a model of TTT as desired, or alternatively, a model of a particular finite axiomatisation.

Finally, we will set out to formally prove the consistency of NF. This will involve breaking down the model-theoretic arguments from [7, 3] into concrete formalisable lemmas.

Implementation details will be discussed in footnotes.

1.2 The simple theory of types

In 1937, Quine introduced *New Foundations* (NF) [5], a set theory with a very small collection of axioms. To give a proper exposition of the theory that we intend to prove consistent, we will first make a digression to introduce the related theory TST, as explained in [4].

The *simple theory of types* (known as *théorie simple des types* or TST) is a first order set theory with several sorts, indexed by the nonnegative integers. Each sort, called a *type*, is comprised of *sets* of that type; each variable 'x' has a nonnegative integer type('x') which denotes the type it belongs to. For convenience, we may write x^n to denote a variable x with type n.

The primitive predicates of this theory are equality and membership. An equality '$x = y$' is a well-formed formula precisely when type('x') = type('y'), and similarly a membership formula '$x \in y$' is well-formed precisely when type('x') + 1 = type('y').

The axioms of this theory are extensionality

$$\forall x^{n+1}, \forall y^{n+1}, (\forall z^n, z^n \in x^{n+1} \leftrightarrow z^n \in y^{n+1}) \to x^{n+1} = y^{n+1}$$

and comprehension

$$\exists x^{n+1}, \forall y^n, (y^n \in x^{n+1} \leftrightarrow \varphi(y^n))$$

where φ is any well-formed formula, possibly with parameters.

Note that these are both axiom schemes, with instances for all type levels n, and (in the latter case) for all well-formed formulae φ. Because extensionality at level $n + 1$ requires us to talk about sets at level n, the inhabitants of type 0, called *individuals*, cannot be examined using these axioms.

By comprehension, there is a set at each nonzero type that contains all sets of the previous type. However, Russell-style paradoxes are avoided as formulae of the form $x^n \in x^n$ are ill-formed.

1.3 New Foundations

New Foundations is a one-sorted first-order theory based on TST. Its primitive propositions are equality and membership. There are no well-formedness constraints on these primitive propositions.

Its axioms are precisely the axioms of TST with all type annotations erased. That is, it has an axiom of extensionality

$$\forall x, \forall y, (\forall z, z \in x \leftrightarrow z \in y) \to x = y$$

and an axiom scheme of comprehension

$$\exists x, \forall y, (y \in x \leftrightarrow \varphi(y))$$

the latter of which is defined for those formulae φ that can be obtained by erasing the type annotations of a well-formed formula of TST. Such formulae are called *stratified*. To avoid the explicit dependence on TST, we can equivalently characterise the stratified formulae as follows. A formula φ is said to be stratified when there is a function σ from the set of variables to the nonnegative integers, in such a way that for each subformula '$x = y$' of φ we have σ('x') = σ('y'), and for each subformula '$x \in y$' we have σ('x') + 1 = σ('y').

It is important to emphasise that while the axioms come from a many-sorted theory, NF is not one; it is well-formed to ask if any set is a member of, or equal to, any other.

Russell's paradox is avoided because the set $\{x \mid x \notin x\}$ cannot be formed; indeed, $x \notin x$ is an unstratified formula. Note, however, that the set $\{x \mid x = x\}$ is well-formed, and so we have a universal set V. Specker showed in [6] that NF disproves the Axiom of Choice, and Hailperin showed in [2] that NF is finitely axiomatisable.

While our main result is that New Foundations is consistent, we attack the problem by means of an indirection through a third theory.

1.4 Tangled type theory

Introduced by Holmes in [3], *tangled type theory* (TTT) is a multi-sorted first order theory based on TST. This theory is parametrised by a limit ordinal λ, the elements of which will index the sorts; ω works, but we prefer generality. As in TST, each variable 'x' has a type that it belongs to, denoted type('x'). However, in TTT, this is not a positive integer, but an element of λ.

The primitive predicates of this theory are equality and membership. An equality '$x = y$' is a well-formed formula when type('x') = type('y'). A membership formula '$x \in y$' is well-formed when type('x') < type('y').

The axioms of TTT are obtained by taking the axioms of TST and replacing all type indices in a consistent way with elements of λ. More precisely, for any order-embedding $s : \omega \to \lambda$, we can convert a well-formed formula φ of TST into a well-formed formula φ^s of TTT by replacing each type variable α with $s(\alpha)$.

It is important to note that membership across types in TTT behaves in some quite bizarre ways. Let $\alpha \in \lambda$, and let x be a set of type α. For any $\beta < \alpha$, the extensionality axiom implies that x is uniquely determined by its type-β elements. However, it is simultaneously determined by its type-γ elements for any $\gamma < \alpha$. In this way, one extension of a set controls all of the other extensions.

The comprehension axiom allows a set to be built which has a specified extension in a single type. The elements not of this type may be considered 'controlled junk'.

We now present the following striking theorem, which we will prove a version of in detail in chapter 8.

Theorem (Holmes). NF is consistent if and only if TTT is consistent. [3]

Thus, our task of proving NF consistent is reduced to the task of proving TTT consistent. We will do this by exhibiting an explicit model (albeit one that requires a great deal of Choice to construct). As TTT has types indexed by a limit ordinal, and sets can only contain sets of lower type, we can construct a model by recursion over λ. In particular, a model of TTT is a well-founded structure. This was not an option with NF directly, as the universal set $V = \{x \mid x = x\}$ would necessarily be constructed before many of its elements.

1.5 Finitely axiomatising tangled type theory

As mentioned above, Hailperin showed in [2] that the comprehension scheme of NF is equivalent to a finite conjunction of its instances. In fact, the same proof shows that the comprehension scheme of TST (and hence that of TTT) is equivalent to a finite conjunction of its instances, but instantiated at all possible type sequences.

We will exhibit one such collection of instances here, totalling eleven axioms. Our choice is inspired by those used in the Metamath implementation of Hailperin's algorithm in [1]. In the following table, the notation $\langle a, b\rangle$ denotes the Kuratowski pair $\{\{a\},\{a,b\}\}$. The first column is Hailperin's name for the axiom, and the second is (usually) the name from [1].

P1(a)	intersection	$\forall x^1 y^1,\ \exists z^1,\ \forall w^0,\ w \in z \leftrightarrow (w \in x \wedge w \in y)$
P1(b)	complement	$\forall x^1,\ \exists z^1,\ \forall w^0,\ w \in z \leftrightarrow w \notin x$
–	singleton	$\forall x^0,\ \exists y^1,\ \forall z^0,\ z \in y \leftrightarrow z = x$
P2	singleton image	$\forall x^3,\ \exists y^4,\ \forall z^0 w^0,\ \langle\{z\},\{w\}\rangle \in y \leftrightarrow \langle z, w\rangle \in x$
P3	insertion two	$\forall x^3,\ \exists y^5,\ \forall z^0 w^0 t^0,\ \langle\{\{z\}\},\langle w, t\rangle\rangle \in y \leftrightarrow \langle z, t\rangle \in x$
P4	insertion three	$\forall x^3,\ \exists y^5,\ \forall z^0 w^0 t^0,\ \langle\{\{z\}\},\langle w, t\rangle\rangle \in y \leftrightarrow \langle z, w\rangle \in x$
P5	cross product	$\forall x^1,\ \exists y^3,\ \forall z^2,\ z \in y \leftrightarrow \exists w^0 t^0,\ z = \langle w, t\rangle \wedge t \in x$
P6	type lowering	$\forall x^4,\ \exists y^1,\ \forall z^0,\ z \in y \leftrightarrow \forall w^1,\ \langle w, \{z\}\rangle \in x$
P7	converse	$\forall x^3,\ \exists y^3,\ \forall z^0 w^0,\ \langle z, w\rangle \in y \leftrightarrow \langle w, z\rangle \in x$
P8	cardinal one	$\exists x^2,\ \forall y^1,\ y \in x \leftrightarrow \exists z^0,\ \forall w,\ w \in y \leftrightarrow w = z$
P9	subset	$\exists x^4,\ \forall y^1 z^1,\ \langle y, z\rangle \in x \leftrightarrow \forall w^0,\ w \in y \to w \in z$

Axioms P1–P9 except for P6 are *predicative*: they stipulate the existence of a set with type at least that of all of the parameters. It is relatively straightforward to prove the consistency of predicative TTT, and we will see later that the proof of P6 in our model takes a different form to the proofs of the other axioms.

Chapter 2

Setting up the environment

In this chapter, we construct the ambient environment inside which our model will reside. To do this, we will set up various pieces of abstract machinery that will help us later. Some mathematical background not already in mathlib will be included in appendix A.

2.1 Conventions

- We are working in Lean's type theory, so cardinals and ordinals are quotients of a large type. In particular, cardinals are not just specific ordinals, and types cannot be ordinals.
- We write $\#\tau$ for the cardinality of a type τ.
- If τ is a type endowed with a well-order $<$, we write $\mathsf{ot}(\tau)$ for the order type of τ with this well-ordering.
- The initial ordinal corresponding to a cardinal c is denoted $\mathsf{ord}(c)$. The cofinality of an ordinal o is $\mathsf{cof}(o)$, and this is a cardinal.
- The symmetric difference of two sets is denoted $s \triangle t := (s \setminus t) \cup (t \setminus s)$. Note that $(s \triangle t) \cup (s \cap t) = s \cup t$, and the union on the left-hand side is disjoint.
- We use $f[s]$ for the direct image $\{f(x) \mid x \in s\}$. We write $f^{-1}[s]$ for the inverse image $\{x \mid f(x) \in s\}$, and $f^{-1}(x)$ for the fibre $\{y \mid f(y) = x\}$.
- For any type α, we write $\mathsf{Part}\,\alpha$ for the type $\sum_{p : \mathsf{Prop}}(p \to \alpha)$.

2.2 Model parameters

Definition 2.1 (model parameters). A collection of *model parameters* is a tuple $(\lambda, <_\lambda, \kappa, \mu)$ such that

- $< \,=\, <_\lambda$ is a well-order on λ, and under this ordering, λ has no maximal element (so $\mathsf{ot}(\lambda)$ is a limit ordinal);
- $\#\kappa$ is uncountable and regular;

- $\#\mu$ is a strong limit, and satisfies

$$\#\kappa < \#\mu; \quad \#\kappa \leq \mathsf{cof}(\mathsf{ord}(\#\mu)); \quad \mathsf{ot}(\lambda) \leq \mathsf{ord}(\mathsf{cof}(\mathsf{ord}(\#\mu)))$$

so in particular, $\mathsf{ot}(\lambda) \leq \mathsf{ord}(\#\mu)$. Note that the inequalities in κ are inequalities of cardinals; the inequality in λ is an inequality of ordinals.

Given a collection of model parameters, we define

- canonical well-orders on κ and μ such that $\mathsf{ot}(\kappa) = \mathsf{ord}(\#\kappa)$ and $\mathsf{ot}(\mu) = \mathsf{ord}(\#\mu)$; and
- a canonical left-cancellative additive monoid on κ, obtained by passing through the equivalence $\kappa \simeq \{o : \mathsf{Ord} \mid o < \mathsf{ord}(\#\kappa)\}$.

Proposition 2.2. The tuple $(\mathbb{N}, <_{\mathbb{N}}, \aleph_1, \beth_{\omega_1})$ is a collection of model parameters, where the symbols $\aleph_1$ and $\beth_{\omega_1}$ represent particular types of that cardinality.

Proof. Direct. □

Definition 2.3 (type index)**.** The type of *type indices* is $\lambda^\perp \coloneqq \mathtt{WithBot}(\lambda)$: the collection of *proper type indices* λ together with a designated symbol $\perp$ which is smaller than all proper type indices. Note that $\mathsf{ot}(\lambda^\perp) = \mathsf{ot}(\lambda)$, and hence that for each $\alpha : \lambda^\perp$,

$$\#\{\beta : \lambda^\perp \mid \beta < \alpha\} \leq \#\{\beta : \lambda^\perp \mid \beta \leq \alpha\} < \mathsf{cof}(\mathsf{ord}(\#\mu))$$

Definition 2.4 (small)**.** A set $s : \mathsf{Set}(\tau)$ is called *small* if $\#s < \#\kappa$. Smallness is stable under subset, intersection, small-indexed unions, symmetric difference, direct image, injective preimage, and many other operations (each of which should be its own lemma when formalised). Sets $s, t : \mathsf{Set}(\tau)$ are called *near* if $s \mathbin{\triangle} t$ is small; in this case, we write $s \overset{N}{\sim} t$. Nearness is an equivalence relation. If $s \overset{N}{\sim} t$ and u is small, then $s \overset{N}{\sim} (t \diamond u)$, where $\diamond$ is one of $\cup, \cap, \setminus, \triangle$.

Definition 2.5 (litter)**.** A *litter* is a triple $L = (\nu, \beta, \gamma) : \mu \times \lambda^\perp \times \lambda$ where $\beta \neq \gamma$. The type of all litters is denoted $\mathcal{L}$, and $\#\mathcal{L} = \#\mu$.

Definition 2.6 (atom)**.** An *atom* is a pair $a = (L, i) : \mathcal{L} \times \kappa$.[1] The type of all atoms is denoted $\mathcal{A}$, and $\#\mathcal{A} = \#\mu$. We write $(-)^\circ : \mathcal{A} \to \mathcal{L}$ for the operation $(L, i) \mapsto L$.[2] We write $\mathsf{LS}(L) \coloneqq \{a \mid a^\circ = L\}$ for the *litter set* of L.[3]

Definition 2.7 (near-litter)**.** A *near-litter* is a pair $N = (L, s) : \mathcal{L} \times \mathsf{Set}\,\mathcal{A}$ such that $s \overset{N}{\sim} \mathsf{LS}(L)$.[4] We write $(-)^\circ : \mathcal{N} \to \mathcal{L}$ for the operation $(L, s) \mapsto L$. We write $a \in N$ for $a \in s$, where $N = (L, s)$. Near-litters satisfy extensionality: there is at most one choice of L for each s. Each near-litter has size $\#\kappa$ when treated as a set of atoms. The type of all near-litters is denoted $\mathcal{N}$, and $\#\mathcal{N} = \#\mu$ (there are $\#\mu$ litters, and $\#\mu$ small sets of atoms by lemma A.6; the latter observation should be its own lemma).

We have $M \overset{N}{\sim} N$ if and only if $M^\circ = N^\circ$. If $M^\circ = N^\circ$, then $M \mathbin{\triangle} N$ is small and $M \cap N$ has size $\#\kappa$. If $M^\circ \neq N^\circ$, then $M \cap N$ has size $\#\kappa$ and $M \cap N$ is small.

[1] This should be formalised as a structure, not as a definition. We should not use the projections of atoms unless absolutely necessary.

[2] This must be a notation typeclass.

[3] Maybe revise this name?

[4] Like with atoms, this should be a structure. We should create an actual constructor, rather than using the $\langle - \rangle$ syntax.

Definition 2.8 (base permutation)**.** A *base permutation* is a pair $\pi = (\pi^{\mathcal{A}}, \pi^{\mathcal{L}})$, where $\pi^{\mathcal{A}}$ is a permutation $\mathcal{A} \simeq \mathcal{A}$ and $\pi^{\mathcal{L}}$ is a permutation $\mathcal{L} \simeq \mathcal{L}$, such that

$$\pi^{\mathcal{A}}[\mathsf{LS}(L)] \overset{N}{\sim} \mathsf{LS}(\pi^{\mathcal{L}}(L))$$

Base permutations have a natural group structure, they act on atoms by $\pi^{\mathcal{A}}$, they act on litters by $\pi^{\mathcal{L}}$, and they act on near-litters by[5]

$$\pi(N)^\circ = \pi(N^\circ); \quad a \in \pi(N) \leftrightarrow a \in \pi[N]$$

Base permutations are determined by their action on atoms. We should avoid directly referencing $\pi^{\mathcal{A}}$ and $\pi^{\mathcal{L}}$ whenever possible.

2.3 The structural hierarchy

We will now establish the hierarchy of types that our model will be built inside.

Definition 2.9 (path)**.** If α, β are type indices, then a *path* $\alpha \rightsquigarrow \beta$ is given by the constructors

- $\mathsf{nil} : \alpha \rightsquigarrow \alpha$;
- $\mathsf{cons} : (\alpha \rightsquigarrow \beta) \to (\gamma < \beta) \to (\alpha \rightsquigarrow \gamma)$.

We define by recursion a snoc operation on the top of paths. We also prove the induction principle for nil and snoc.

A path $\alpha \rightsquigarrow \bot$ is called an *α-extended index*. We write $\mathsf{nil}(\alpha)$ for the path $\{\alpha\} : \alpha \rightsquigarrow \alpha$. If h is a proof of $\beta < \alpha$, we write $\mathsf{single}(h)$ for the path $\{\alpha, \beta\} : \alpha \rightsquigarrow \beta$.

We have the inequality

$$\begin{aligned}\#(\alpha \rightsquigarrow \beta) &\le (\#\{\gamma : \lambda^\bot \mid \gamma \le \alpha\})^{<\omega} \\ &= \mathsf{max}(\aleph_0, \#\{\gamma : \lambda^\bot \mid \gamma \le \alpha\}) \\ &< \mathsf{cof}(\mathsf{ord}(\#\mu))\end{aligned}$$

Many of the objects in this construction have an associated type level $\alpha : \lambda^\bot$, and by application of a path of the form $\alpha \rightsquigarrow \beta$ or $\beta \rightsquigarrow \alpha$, we can often define a new object of type level β. For this common task, we register the following notation typeclasses.

- $x \Downarrow A$ is the *derivative* of an object of type β along a path $A : \beta \rightsquigarrow \gamma$, giving an object of type γ;
- $x \downarrow h$ abbreviates $x \Downarrow \mathsf{single}(h)$;[6]
- $x \underset{\bot}{\Downarrow} A$ is the *base derivative* of an object of type β along a path $A : \beta \rightsquigarrow \bot$;
- $x \underset{\bot}{\downarrow}$ abbreviates $x \underset{\bot}{\Downarrow} \mathsf{single}(h)$ where h is the canonical proof of $\bot < \beta$ whenever $\beta : \lambda$;
- $x \Uparrow A$ is the *coderivative* of an object of type β along a path $A : \alpha \rightsquigarrow \beta$, giving an object of type α;

[5] We need to emphasise these properties, rather than emphasising the real definition $\pi(N) = (\pi(N^\circ), \pi[N])$.

[6] In practice the typeclasses will probably not formally depend on each other, and this 'abbreviation' may not be a definitional equality.

- $x \uparrow h$ abbreviates $x \Uparrow \mathsf{single}(h)$.

When we say that an object has an associated type level in this context, we mean that the notation typeclass is registered in the following form.

```
class Derivative (X : Type _) (Y : outParam (Type _))
    (β : outParam TypeIndex) (γ : TypeIndex) where
  deriv : X → Path β γ → Y
```

This means that when inferring the type of the expression $x \Downarrow A$, we first compute the type of x, which gives rise to a unique type index β, then the type of A is inferred to give γ, then the output type Y is uniquely determined.

The reason that we distinguish $\underset{\bot}{\Downarrow}$ from $\Downarrow$ is that the associated type Y is allowed to differ between the two forms. We will give a brief example motivated by a definition we are about to make. For each type index β, there is a type of β-structural permutation, comprised of many base permutations. If we have a path $\beta \rightsquigarrow \gamma$, we can convert a β-structural permutation into a γ-structural permutation; this will be the derivative map. We will see that a given $\bot$-structural permutation contains exactly one base permutation, and so the types are in canonical isomorphism. If x is a β-structural permutation and $A : \beta \rightsquigarrow \bot$, then $x \Downarrow A$ is a $\bot$-structural permutation, and $x \underset{\bot}{\Downarrow} A$ is the corresponding base permutation.

Because $\uparrow, \downarrow$ and others are already used by Lean, we use slightly different notation in practice (e.g. $\nearrow, \searrow$). In this writeup, however, we will use subscripts for derivatives and superscripts for coderivatives. We will not distinguish typographically here between the single- and double-struck variants, or between $\Downarrow$ and $\underset{\bot}{\Downarrow}$; in the latter case, the syntax x_A always means $x \underset{\bot}{\Downarrow} A$ whenever A has minimal element $\bot$. We will also use x_γ to denote the derivative x_h where h is some proof of $\gamma < \beta$, and use x^α to denote x^h where $h : \beta < \alpha$.

Definition 2.10 (derivatives of paths)**.** If $A : \alpha \rightsquigarrow \beta$ and $B : \beta \rightsquigarrow \gamma$, the derivative A_B is defined to be the union $A \cup B : \alpha \rightsquigarrow \gamma$, and the coderivative B^A is defined to be A_B.

Definition 2.11 (tree)**.** Let τ be any type, and let α be a type index. An *α-tree of τ* is a function t that maps each α-extended index A to an object $t_A : \tau$; this defines its base derivatives. The type of $\bot$-trees of τ is canonically isomorphic to τ. If t is an α-tree and $A : \alpha \rightsquigarrow \beta$, we define the derivative t_A to be the β-tree defined by $(t_A)_B = t_{(A_B)}$. This derivative map is functorial: for any paths $A : \alpha \rightsquigarrow \beta$ and $B : \beta \rightsquigarrow \gamma$, we have $t_{(A_B)} = (t_A)_B$. If τ has a group structure, then so does its type of trees: $(t \cdot u)_A = t_A \cdot u_A$ and $(t^{-1})_A = (t_A)^{-1}$. If τ acts on υ, then α-trees of τ act on α-trees of υ: $(t(u))_A = t_A(u_A)$.

If $\#\tau \leq \#\mu$, there are at most $\#\mu$-many α-trees of τ, since there are less than $\mathsf{cof}(\mathsf{ord}(\#\mu))$-many α-extended indices, allowing us to conclude by lemma A.6 as each such tree is a subset of $(\alpha \rightsquigarrow \bot) \times \tau$ of size less than $\mathsf{cof}(\mathsf{ord}(\#\mu))$. If $\#\tau < \#\mu$, there are less than $\#\mu$-many α-trees of τ, since there are less than $\mathsf{cof}(\mathsf{ord}(\#\mu))$-many α-extended indices and strong limits are closed under exponentials.

Definition 2.12 (structural permutation)**.** Let α be a type index. Then an *α-structural permutation* (or just *α-permutation*) is an α-tree of base permutations. The type of α-permutations is written $\mathsf{StrPerm}_\alpha$.

As an implementation detail, we create a typeclass $\texttt{StrPermClass}_\alpha$ for permutations that 'act like' α-permutations: they have a group structure and a canonical group embedding into $\mathsf{StrPerm}_\alpha$. When we quantify over structural permutations in this paper, it should be formalised using an additional quantification over $\texttt{StrPermClass}_\alpha$.

Definition 2.13 (enumeration)**.** Let τ be a type. An *enumeration* of τ is a pair $E = (i, f)$ where $i : \kappa$ and f is a partial function $\kappa \to \tau$, such that all domain elements of f are strictly less than i.[7] If $x : \tau$, we write $x \in E$ for $x \in \operatorname{ran} f$. The set $\{y \mid y \in E\}$, which we may also loosely call E, is small. We will write $\varnothing$ for the empty enumeration $(0, \varnothing)$.

If $g : \tau \to \sigma$, then g lifts to a direct image function mapping enumerations of τ to enumerations of σ:

$$g(i, f) = (i, f'); \quad f' = \{(j, g(x)) \mid (j, x) \in f\}$$

Thus, $x \in g(E) \leftrightarrow x \in g[E]$. In the same way, groups that act on τ also act on enumerations of τ.[8] If $g : \sigma \to \tau$ is injective, then g lifts to an inverse image function mapping enumerations of τ to enumerations of σ:

$$g^{-1}(i, f) = (i, f'); \quad f' = \{(j, x) \mid (j, g(x)) \in f\}$$

This operation may cause the domain of f to shrink, but we will keep i the same.

If $E = (i, e)$ and $F = (j, f)$ are enumerations of τ, we define their *concatenation* by

$$E + F = (i + j, e \cup f'); \quad f' = \{(i + k, x) \mid (k, x) \in f\}$$

This operation commutes with the others: $x \in E + F \leftrightarrow x \in E \lor x \in F$, $g[E + F] = g[E] + g[F]$, and $g^{-1}[E + F] = g^{-1}[E] + g^{-1}[F]$.

We define a partial order on enumerations by setting $(i, e) \leq (j, f)$ if and only if f extends e as a partial function. We obtain various corollaries, such as $E \leq F \to g(E) \leq g(F)$ and $E \leq E + F$.

Every small subset of τ is the range of some enumeration of τ.

If $\#\tau \leq \#\mu$, then there are at most $\#\mu$-many enumerations of τ: enumerations are determined by an element of κ and a small subset of $\kappa \times \tau$. If $\#\tau < \#\mu$, then there are strictly less than $\#\mu$-many enumerations of τ: use the same reasoning and apply lemma A.6.

Definition 2.14 (base support)**.** A *base support* is a pair $S = (S^{\mathcal{A}}, S^{\mathcal{N}})$ where $S^{\mathcal{A}}$ is an enumeration of atoms and $S^{\mathcal{N}}$ is an enumeration of near-litters. There are precisely $\#\mu$ base supports.

Definition 2.15 (structural support)**.** A β-*structural support* (or just β-*support*) is a pair $S = (S^{\mathcal{A}}, S^{\mathcal{N}})$ where $S^{\mathcal{A}}$ is an enumeration of $(\beta \rightsquigarrow \bot) \times \mathcal{A}$ and $S^{\mathcal{N}}$ is an enumeration of $(\beta \rightsquigarrow \bot) \times \mathcal{N}$. The type of β-supports is written $\mathsf{StrSupp}_\beta$. There are precisely $\#\mu$ structural supports for any type index β.

For a path $A : \beta \rightsquigarrow \bot$, we write S_A for the base support T given by

$$T^{\mathcal{A}} = \{(i, a) \mid (i, (A, a)) \in S^{\mathcal{A}}\}; \quad T^{\mathcal{N}} = \{(i, N) \mid (i, (A, N)) \in S^{\mathcal{N}}\}$$

More generally, for a path $A : \beta \rightsquigarrow \gamma$, we write S_A for the γ-support T given by

$$T^{\mathcal{A}} = \{(i, (B, a)) \mid (i, (A_B, a)) \in S^{\mathcal{A}}\}; \quad T^{\mathcal{N}} = \{(i, (B, N)) \mid (i, (A_B, N)) \in S^{\mathcal{N}}\}$$

For a path $A : \alpha \rightsquigarrow \beta$, we write S^A for the α-support T given by

$$T^{\mathcal{A}} = \{(i, (A_B, a)) \mid (i, (B, a)) \in S^{\mathcal{A}}\}; \quad T^{\mathcal{N}} = \{(i, (A_B, N)) \mid (i, (B, N)) \in S^{\mathcal{N}}\}$$

Clearly, $(S^A)_A = S$.

β-structural permutations act on pairs (A, x) by $\pi(A, x) = (A, \pi_A(x))$, where x is either an atom or a near-litter. Hence, structural permutations act on structural supports.

---

[7] This should be encoded as a coinjective relation $\kappa \to \tau \to \mathsf{Prop}$; see definition A.1.

[8] Actually, we should probably implement this using the inverse image not the direct image for better definitional equalities.

Let τ be a type, and let G be a $\texttt{StrPermClass}_\beta$-group that acts on τ. We say that S *supports* $x : \tau$ under the action of G if whenever $\pi : G$ fixes S, it also fixes x, and moreover, if $\beta = \bot$ then $S_A^{\mathcal{N}}$ is empty for any $A : \bot \rightsquigarrow \bot$ (and of course there is exactly one such path).

Definition 2.16 (structural set)**.** The type of *α-structural sets*, denoted StrSet_α, is defined by well-founded recursion on $\lambda^\bot$.

- $\mathsf{StrSet}_\bot \coloneqq \mathcal{A}$;
- $\mathsf{StrSet}_\alpha \coloneqq \prod_{\beta : \lambda^\bot} \beta < \alpha \to \mathsf{Set}\,\mathsf{StrSet}_\beta$ where $\alpha : \lambda$.

These equalities will in fact only be equivalences in the formalisation. We define the action of α-permutations (more precisely, inhabitants of some type with a $\texttt{StrPermClass}_\alpha$ instance) on α-structural sets by well-founded recursion.

- $\pi(x) = \pi_{\mathsf{nil}(\bot)}(x)$ if $\alpha \equiv \bot$;
- $\pi(x) = (\beta, h \mapsto \pi_h[x(\beta, h)])$ if $\alpha : \lambda$.

2.4 Position functions

Definition 2.17 (position function)**.** Let τ be a type. A *position function* for τ is an injection $\iota : \tau \to \mu$. This is a typeclass.

Proposition 2.18 (injective functions from denied sets)**.** Let τ be a type such that $\#\tau \leq \#\mu$. Let $D : \tau \to \mathsf{Set}(\mu)$ be a function such that for each $x : \tau$, the set $D(x)$, called the *denied set* of x, has size less than $\mathsf{cof}(\mathsf{ord}(\#\mu))$. Then there is an injective function $f : \tau \to \mu$ such that if $\nu \in D(x)$, then $\nu < f(x)$.

Proof. Pick a well-ordering $\prec$ of τ of length at most $\mathsf{ord}(\#\mu)$. Define f by well-founded recursion on $\prec$. Suppose that we have already defined f for all $y \prec x$. Then let

$$X = \{f(y) \mid y \prec x\} \cup \{\nu \mid \exists \nu' \in D(x),\ \nu \leq \nu'\}$$

This set has size strictly less than μ, so there is some $\nu : \mu$ not contained in it. Set $f(x) = \nu$. ☐

Proposition 2.19 (base positions)**.** There are position functions on $\mathcal{A}, \mathcal{N}$ that are jointly injective and satisfy

- $\iota(\mathsf{NL}(a^\circ)) < \iota(a)$ for every atom a;
- $\iota(\mathsf{NL}(N^\circ)) \leq \iota(N)$ for every near-litter N;
- $\iota(a) < \iota(N)$ for every near-litter N and atom $a \in N \triangle \mathsf{LS}(N^\circ)$.[9]

We register these position functions as instances for use in typeclass inference. We also define $\iota(L) = \iota(\mathsf{NL}(L))$ for litters.

Proof. First, establish an equivalence $f_{\mathcal{L}} : \mathcal{L} \simeq \mu$. Use proposition 2.18 to obtain an injective map $f_{\mathcal{A}} : \mathcal{N} \to \mu$ such that $f_{\mathcal{L}}(a^\circ) < f_{\mathcal{A}}(a)$ for each atom a. Now use proposition 2.18 again to obtain an injective map $f_{\mathcal{N}} : \mathcal{N} \to \mu$ such that

$$f_{\mathcal{L}}(N^\circ) < f_{\mathcal{N}}(N); \quad f_{\mathcal{A}}(a) < f_{\mathcal{N}}(N) \text{ for } a \in N \triangle \mathsf{LS}(N^\circ)$$

[9]TODO: Make syntax for $N \triangle \mathsf{LS}(N^\circ)$.

Endow $3 \times \mu$ with its lexicographic order, of order type $\mathsf{ord}(\#\mu)$, giving an order isomorphism $e : 3 \times \mu \simeq \mu$. Finally, we define

$$\iota(a) = e(1, f_{\mathcal{A}}(a)); \quad \iota(N) = \begin{cases} e(0, f_{\mathcal{L}}(N^\circ)) & \text{if } N = \mathsf{NL}(N^\circ) \\ e(2, f_{\mathcal{N}}(N)) & \text{otherwise} \end{cases}$$

□

2.5 Hypotheses of the recursion

Definition 2.20 (model data)**.** Let α be a type index. *Model data* at type α consists of:[10]

- a TSet_α called the type of *tangled sets* or *t-sets*, which will be our type of model elements at level α, with an embedding $U_\alpha : \mathsf{TSet}_\alpha \to \mathsf{StrSet}_\alpha$;
- a group $\mathsf{AllPerm}_\alpha$ called the type of *allowable permutations*, with a $\mathtt{StrPermClass}_\alpha$ instance and a specified action on TSet_α,

such that

- if $\rho : \mathsf{AllPerm}_\alpha$ and $x : \mathsf{TSet}_\alpha$, then $\rho(U_\alpha(x)) = U_\alpha(\rho(x))$;
- every t-set of type α has an α-support (definition 2.15) for its action under the α-allowable permutations.

Definition 2.21 (tangle)**.** Let α be a type index with model data. An *α-tangle* is a pair $t = (x, S)$ where x is a tangled set of type α and S is an α-support for x. We define $\mathsf{set}(t) = x$ and $\mathsf{supp}(t) = S$. The type of α-tangles is denoted Tang_α. Allowable permutations ρ act on tangles by $\rho((x, S)) = (\rho(x), \rho(S))$, and so $\mathsf{supp}(t)$ supports t for its action under the allowable permutations. Therefore, each tangled set x is of the form $\mathsf{set}(t)$ for some tangle t.

Proposition 2.22 (fuzz maps)**.** Let β be a type index with model data, and suppose that Tang_β has a position function. Let γ be any proper type index not equal to β. Then there is an injective *fuzz map* $f_{\beta,\gamma} : \mathsf{Tang}_\beta \to \mathcal{L}$ such that $\iota(t) < \iota(f_{\beta,\gamma}(t))$, and the different $f_{\beta,\gamma}$ all have disjoint ranges. In particular, for any near-litter N with $N^\circ = f_{\beta,\gamma}(t)$, we have $\iota(t) < \iota(N)$, and additionally, for any atom a with $a^\circ = f_{\beta,\gamma}(t)$, we have $\iota(t) < \iota(a)$.[11]

Proof. We define $\mathsf{g} : \mathsf{Tang}_\beta \to \mu$ by proposition 2.18, where the denied sets are given by $D(t) = \{\iota(t)\}$. Then we set $f_{\beta,\gamma}(t) = (\mathsf{g}(t), \beta, \gamma)$. □

Definition 2.23 (inflexible path)**.** Let α be a proper type index. Suppose that we have model data for all type indices $\beta \leq \alpha$ and position functions for Tang_β for all $\beta < \alpha$. For any type index $\beta \leq \alpha$, a *inflexible β-path* is a tuple $I = (\gamma, \delta, \varepsilon, A)$ where $\delta, \varepsilon < \gamma$ are distinct, the index ε is proper, and $A : \beta \rightsquigarrow \gamma$. We will write $\gamma_I, \delta_I, \varepsilon_I, A_I$ for its fields. Inflexible paths have a coderivative operation, given by $(\gamma, \delta, \varepsilon, A)^B = (\gamma, \delta, \varepsilon, A^B)$.

[10] A type theory problem with `exporting` this data is that under different assumptions, things like different spellings of TSet_α might require case splitting on α before they become defeq (e.g. see the old version of `Model/FOA.lean`). There doesn't seem to be an easy way around this.

[11] We might want to encapsulate atoms and near-litters somehow. We could make a typeclass, or write theorems in terms of the coproduct $\mathcal{A} \oplus \mathcal{N}$.

Definition 2.24 (typed near-litter)**.** Let α be a proper type index with model data, and suppose that Tang_α has a position function. We say that α has *typed near-litters* if there is an embedding $\mathsf{typed}_\alpha : \mathcal{N} \to \mathsf{TSet}_\alpha$ such that

- if $\rho : \mathsf{AllPerm}_\alpha$ and $N : \mathcal{N}$, then $\rho(\mathsf{typed}_\alpha(N)) = \mathsf{typed}_\alpha(\rho_\perp(N))$; and
- if N is a near-litter and t is an α-tangle such that $\mathsf{set}(t) = \mathsf{typed}_\alpha(N)$, we have $\iota(N) \leq \iota(t)$.

Objects of the form typed_α are called *typed near-litters*.

Definition 2.25 (coherent data)**.** Let α be a proper type index. Suppose that we have model data for all type indices $\beta \leq \alpha$, position functions for Tang_β for all $\beta < \alpha$, and typed near-litters for all $\beta < \alpha$. We say that the model data is *coherent* below level α if the following hold.

- For $\gamma < \beta \leq \alpha$, there is a map $\mathsf{AllPerm}_\beta \to \mathsf{AllPerm}_\gamma$ that commutes with the coercions from $\mathsf{AllPerm}_{(-)}$ to $\mathsf{StrPerm}_{(-)}$. We can use this map to define arbitrary derivatives of allowable permutations by recursion on paths.
- If (x, S) is a β-tangle for $\beta < \alpha$, and y is an atom or near-litter that occurs in the range of S_A, then $\iota(y) < \iota(x, S)$.
- Let $\beta \leq \alpha$, and let $\gamma, \delta < \beta$ be distinct with δ proper. Let $t : \mathsf{Tang}_\gamma$ and $\rho : \mathsf{AllPerm}_\beta$. Then
$$(\rho_\delta)_\perp(f_{\gamma,\delta}(t)) = f_{\gamma,\delta}(\rho_\gamma(t))$$
In particular, for every $\beta \leq \alpha$, β-allowable permutation ρ, and β-inflexible path I, we obtain
$$((\rho_A)_\varepsilon)_\perp(f_{\delta,\varepsilon}(t)) = f_{\delta,\varepsilon}((\rho_A)_\delta(t))$$
where subscripts of I are suppressed.
- Suppose that $\beta \leq \alpha$ and $(\rho(\gamma))_{\gamma<\beta}$ is a collection of allowable permutations such that whenever $\gamma, \delta < \beta$ are distinct, δ is proper, and $t : \mathsf{Tang}_\delta$, we have
$$(\rho(\delta))_\perp(f_{\gamma,\delta}(t)) = f_{\gamma,\delta}(\rho(\gamma)(t))$$
Then there is a β-allowable permutation ρ with $\rho_\gamma = \rho(\gamma)$ for each $\gamma < \beta$.
- If $\beta \leq \alpha$ is a proper type index and $x : \mathsf{TSet}_\beta$, then for any $\gamma < \beta$,
$$U_\beta(x)(\gamma) \subseteq \mathsf{ran}\, U_\gamma$$
- (extensionality) If $\beta, \gamma \leq \alpha$ are proper type indices where $\gamma < \beta$, the map $U_\beta(-)(\gamma) : \mathsf{TSet}_\beta \to \mathsf{Set}\,\mathsf{StrSet}_\gamma$ is injective.
- If $\beta, \gamma : \lambda$ where $\gamma < \beta$, for every $x : \mathsf{TSet}_\gamma$ there is some $y : \mathsf{TSet}_\beta$ such that $U_\beta(y)(\gamma) = \{x\}$. We write $\mathsf{singleton}_\beta(x)$ for this object y, which is uniquely defined by extensionality.

Note that this structure contains data (the derivative maps for allowable permutations and the singleton operations), but the type is a subsingleton: any two inhabitants are propositionally equal. We will not use this fact directly, but the idea will have relevance in chapter 6.

The strategy of our construction is as follows.

- In chapter 3, we assume model data, position functions, and typed near-litters for all types $\beta < \alpha$, and construct model data at level α.

- In chapters 4 and 5, we assume coherent data below level α (along with the assumptions required for this definition to make sense) and prove that $\#\mathsf{TSet}_\alpha = \#\mu$.
- In chapter 6, we use the results of chapters 3 to 5 to show that we can provide position functions and typed near-litters at level α. We then show that these constructions may be iterated so that we may define all of the above structures at every proper type level.

Chapter 3

Constructing the types

In this section, we are trying to construct the type of tangled sets at level α. We assume model data, position functions for Tang_β, and typed near-litters for all types $\beta < \alpha$.

3.1 Codes and clouds

Definition 3.1 (code). A *code* is a pair $c = (\beta, s)$ where $\beta < \alpha$ is a type index and s is a nonempty set of TSet_β.

Definition 3.2 (cloud). The *cloud relation* $\prec$ on codes is given by the constructor

$$(\beta, s) \prec (\gamma, \{\mathsf{typed}_\gamma(N) \mid \exists t : \mathsf{Tang}_\beta,\ \mathsf{set}(t) \in s \land N^\circ = f_{\beta,\gamma}(t)\})$$

where $\beta, \gamma < \alpha$ are distinct and γ is proper.

Proposition 3.3. If $c \prec (\gamma, s_1)$ and $c \prec (\gamma, s_2)$, then $s_1 = s_2$.

Proof. Let $c = (\beta, s)$. We obtain

$$s_1 = \{\mathsf{typed}_\gamma(N) \mid \exists t : \mathsf{Tang}_\beta,\ \mathsf{set}(t) \in s \land N^\circ = f_{\beta,\gamma}(t)\} = s_2$$

as required. □

Proposition 3.4. The cloud relation is injective (definition A.1). That is, if $c_1, c_2 \prec d$, then $c_1 = c_2$.

Proof. Let $c_i = (\beta_i, s_i)$ for $i = 1, 2$, and let $d = (\gamma, s')$.

We first show that $\beta_1 = \beta_2$. Choose some $t_1 : \mathsf{Tang}_{\beta_1}$ such that $\mathsf{set}(t_1) \in s_1$. We can show directly that $\mathsf{typed}_\gamma(\mathsf{NL}(f_{\beta_1,\gamma}(t_1))) \in s'$. So there is some $t_2 : \mathsf{Tang}_{\beta_2}$ such that

$$\mathsf{typed}_\gamma(\mathsf{NL}(f_{\beta_1,\gamma}(t_1))) = \mathsf{typed}_\gamma(N); \quad N^\circ = f_{\beta_2,\gamma}(t_2)$$

Then since the typed near-litter map is injective (definition 2.24), the fact that the equations $N^\circ = L_1$ and $N = \mathsf{NL}(L_2)$ imply $L_1 = L_2$, and that the f-maps have disjoint ranges (proposition 2.22), we obtain $\beta_1 = \beta_2$.

We now show that if $(\beta, s_1), (\beta, s_2) \prec d$, then $s_1 \subseteq s_2$. Let $d = (\gamma, s')$ as above. Let $x \in s_1$, and choose $t_1 : \mathsf{Tang}_\beta$ such that $x = \mathsf{set}(t_1)$. Then as $(\beta, s_1) \prec d$, we have $\mathsf{typed}_\gamma(\mathsf{NL}(f_{\beta,\gamma}(t_1))) \in s'$. So there is some $t_2 : \mathsf{Tang}_\beta$ with $\mathsf{set}(t_2) \in s_2$ such that

$$\mathsf{typed}_\gamma(\mathsf{NL}(f_{\beta,\gamma}(t_1))) = \mathsf{typed}_\gamma(N); \quad N^\circ = f_{\beta,\gamma}(t_2)$$

For the same reasons as above, together with injectivity of $f_{\beta,\gamma}$, we have $t_1 = t_2$. In particular, $x \in s_2$ as required.

This gives the required result by antisymmetry. □

Proposition 3.5. The cloud relation is well-founded.

Proof. Define a function F that maps a code $c = (\beta, s)$ to the set

$$\{\iota(t) \mid \mathsf{set}(t) \in s\} : \mathsf{Set}\,\mu$$

We first show that $c_1 \prec c_2$ implies that

$$\forall \nu_2 \in F(c_2), \exists \nu_1 \in F(c_1), \nu_1 < \nu_2$$

Let $c_i = (\beta_i, s_i)$ for $i = 1, 2$, and suppose $\nu_2 \in F(c_2)$. Then $\nu_2 = \iota(t_2)$ with $\mathsf{set}(t_2) \in s_2$. By definition, $\mathsf{set}(t_2) = \mathsf{typed}_{\beta_2}(N)$ where $N^\circ = f_{\beta_1,\beta_2}(t_1)$ and $\mathsf{set}(t_1) \in s_1$. Then $\iota(t_1) \in F(c_1)$, and $\iota(t_1) < \iota(N) \leq \iota(t_2)$ by proposition 2.22 and definition 2.24, as required.

Now, we define a function f that maps a code c to $\min F(c)$, which is always well-defined as $F(c)$ is nonempty. The above result shows that $c_1 \prec c_2$ implies $f(c_1) < f(c_2)$. Thus $\prec$ is a subrelation of the well-founded relation given by the inverse image of f, so is well-founded. □

Proposition 3.6. Let $\prec$ be a relation on a type τ. We say that an object $x : \tau$ is $\prec$-*even* if all of its $\prec$-predecessors are odd; we say that x is $\prec$-*odd* if it has a $\prec$-predecessor that is even. Then:

1. Minimal objects are even.
2. If $\prec$ is well-founded, then every object $x : \tau$ is either even or odd, but not both.

Proof. Part 1. Direct from the definition.

Part 2. We show this by induction along $\prec$. Suppose that all $\prec$-predecessors of x are either even or odd but not both. If all of these $\prec$-predecessors are odd, then x is even, and it is clearly not odd, because no $y \prec x$ is even. Otherwise, there is $y \prec x$ that is even, so x is odd, and it is not even because this y is not odd. □

Definition 3.7. We define the relation $\looparrowright$ between codes by the following two constructors.

- If c is a $\prec$-even code, then $c \looparrowright c$.
- If c is a $\prec$-even code and $c \prec d$, then $c \looparrowright d$.

This relation is cofunctional (definition A.1): if d is a code, there is exactly one c such that $c \looparrowright d$. Moreover, if $c \looparrowright (\beta, s_1), (\beta, s_2)$, then $s_1 = s_2$.

Proof of claim. If d is even, then $d \looparrowright d$. If c is any other even code, $c \not\prec d$.

If d is odd, then there is an even code c with $c \prec d$, so $c \looparrowright d$. If c' is any other even code with $c' \looparrowright d$, we must have $c' \prec d$ as c' and d have different parities so cannot be equal, so $c, c' \prec d$, so $c = c'$ by proposition 3.4.

Finally, suppose $c \looparrowright (\beta, s_1), (\beta, s_2)$. Suppose that $c = (\beta, s_1)$. Then $(\beta, s_1) \looparrowright (\beta, s_2)$ implies $s_1 = s_2$ because in the other constructor we may conclude $\beta \neq \beta$. The same holds for $c = (\beta, s_2)$ by symmetry. Now suppose that $c \prec (\beta, s_1), (\beta, s_2)$. In this case, we immediately obtain $s_1 = s_2$ by proposition 3.3. □

Proposition 3.8 (extensionality). Let $x : \mathsf{TSet}_\beta$ for some type index $\beta < \alpha$, and let c be a code. We say that x is a *type-β member* of c if there is a set $s : \mathsf{Set}\,\mathsf{TSet}_\beta$ such that $x \in s$ and $c \looparrowright (\beta, s)$, and hence for all sets $s : \mathsf{Set}\,\mathsf{TSet}_\beta$ such that $c \looparrowright (\beta, s)$, we have $x \in s$ by definition 3.7. We write $x \in_\beta c$. Note that this definition is only useful when c is even.

Let c_1, c_2 be even codes. If $\beta < \alpha$ is a proper type index such that

$$\forall x : \mathsf{TSet}_\beta,\ x \in_\beta c_1 \leftrightarrow x \in_\beta c_2$$

then $c_1 = c_2$.

Proof. Suppose that there is no $x : \mathsf{TSet}_\beta$ such that $x \in_\beta c_1$. Then it is easy to check that $c_1 = (\gamma, \varnothing)$ for some γ, which is a contradiction as all codes are assumed to have nonempty second projections.

So there is some $x : \mathsf{TSet}_\beta$ such that $x \in_\beta c_1$. Then there are sets s_1, s_2 where $c_i \looparrowright (\beta, s_i)$ for $i = 1, 2$. Then, as $x \in_\beta c_i$ holds if and only if $x \in s_i$, we conclude $s_1 = s_2$. Hence $c_1, c_2 \looparrowright (\beta, s_1)$, so as $\looparrowright$ is injective, we conclude $c_1 = c_2$. □

3.2 Model data defined

Definition 3.9 (new allowable permutation). A *new allowable permutation* is a dependent function ρ of type $\prod_{\beta<\alpha} \mathsf{AllPerm}_\beta$, subject to the condition

$$(\rho_\gamma)_\perp(f_{\beta,\gamma}(t)) = f_{\beta,\gamma}(\rho(\beta)(t))$$

for every $t : \mathtt{Tang}_\beta$. These form a group and have a $\mathtt{StrPermClass}_\alpha$ instance.

Proposition 3.10. Define an action of allowable permutations on codes by

$$\rho(\beta, s) = (\beta, \rho(\beta)[s])$$

Then

1. $c \prec d$ implies $\rho(c) \prec \rho(d)$;
2. $c \looparrowright d$ implies $\rho(c) \looparrowright \rho(d)$;
3. c is even if and only if $\rho(c)$ is even;
4. $x \in_\beta c$ if and only if $\rho(\beta)(x) \in_\beta \rho(c)$.

Proof. Part 1. Suppose that $c \prec d$. Then, writing $c = (\beta, s)$ and $d = (\gamma, s')$, we obtain

$$s' = \{\mathsf{typed}_\gamma(N) \mid \exists t : \mathsf{Tang}_\beta,\ \mathsf{set}(t) \in s \wedge N^\circ = f_{\beta,\gamma}(t)\}$$

Now, note that

$$\begin{aligned}
\rho(\gamma)[s'] &= \rho(\gamma)[\{\mathsf{typed}_\gamma(N) \mid \exists t : \mathsf{Tang}_\beta,\ \mathsf{set}(t) \in s \wedge N^\circ = f_{\beta,\gamma}(t)\}] \\
&= \{\rho(\gamma)(\mathsf{typed}_\gamma(N)) \mid \exists t : \mathsf{Tang}_\beta,\ \mathsf{set}(t) \in s \wedge N^\circ = f_{\beta,\gamma}(t)\} \\
&= \{\mathsf{typed}_\gamma(\rho(\gamma)_\perp(N)) \mid \exists t : \mathsf{Tang}_\beta,\ \mathsf{set}(t) \in s \wedge N^\circ = f_{\beta,\gamma}(t)\} \\
&= \{\mathsf{typed}_\gamma(N) \mid \exists t : \mathsf{Tang}_\beta,\ \mathsf{set}(t) \in s \wedge \rho(\gamma)_\perp^{-1}(N)^\circ = f_{\beta,\gamma}(t)\} \\
&= \{\mathsf{typed}_\gamma(N) \mid \exists t : \mathsf{Tang}_\beta,\ \mathsf{set}(t) \in s \wedge N^\circ = \rho(\gamma)_\perp(f_{\beta,\gamma}(t))\} \\
&= \{\mathsf{typed}_\gamma(N) \mid \exists t : \mathsf{Tang}_\beta,\ \mathsf{set}(t) \in s \wedge N^\circ = f_{\beta,\gamma}(\rho(\beta)(t))\}
\end{aligned}$$

So $\rho(c) \prec \rho(d)$ as required.

Part 2. Direct.

Part 3. Follows from the general fact that if $f : \tau \to \sigma$ is a bijection and we have $x \prec_\tau y$ if and only if $f(x) \prec_\sigma f(y)$, then the $\prec_\tau$-parity of x is the same as the $\prec_\sigma$-parity of $f(x)$.

Part 4. We only need to show one direction, because we can apply the one-directional result to ρ^{-1} to obtain the converse. Suppose that $x \in_\beta c$, so $c \looparrowright (\beta, s)$ and $x \in s$. Then $\rho(c) \looparrowright (\beta, \rho(\beta)[s])$, so $\rho(\beta)(x) \in_\beta \rho(c)$ as required, □

Definition 3.11 (new t-set). A *new t-set* is an even code c such that there is an α-support that supports c under the action of new allowable permutations, or a designated object called the empty set. New allowable permutations act on new t-sets in the same way that they act on codes, and map the empty set to itself. We define the map U_α from new t-sets to StrSet_α by cases from the top of the path in the obvious way (using recursion on paths and the membership relation from proposition 3.8). It is easy to check that $\rho(U_\alpha(x)) = U_\alpha(\rho(x))$ by proposition 3.10.

Definition 3.12 (new model data). Given model data, position functions, and typed near-litters for all types $\beta < \alpha$, *new model data* is the model data at level α consisting of new t-sets (definition 3.11) and new allowable permutations (definition 3.9). The embedding from new t-sets to StrSet_α is defined by

$$U_\alpha(c)(\beta) = \{x \mid x \in_\beta c\}$$

3.3 Typed near-litters, singletons, and positions

Definition 3.13 (typed near-litters). We define a function typed_α from the type of near-litters to the type of new t-sets by mapping a near-litter N to the code $(\perp, N)$. This code is even as all codes of the form $(\perp, s)$ are even. This function is clearly injective, and satisfies

$$\rho(\mathsf{typed}_\alpha(N)) = \mathsf{typed}_\alpha(\rho(\perp)(N))$$

by definition.

Definition 3.14 (singletons). We define a function $\mathsf{singleton}_\alpha$ for each lower type index β from TSet_β to the type of new t-sets by $x \mapsto (\beta, \{x\})$. The given code is even as all singleton codes are even. This satisfies $U_\alpha(\mathsf{singleton}_\alpha(x))(\beta) = \{x\}$.

Proposition 3.15 (position function)**.** Using the model data from definition 3.12, if $\#\mathsf{Tang}_\alpha \leq \#\mu$, then there is a position function on the type of α-tangles,[1] such that

- if N is a near-litter and t is an α-tangle such that $\mathsf{set}(t) = \mathsf{typed}_\alpha(N)$, we have $\iota(N) \leq \iota(t)$; and
- if t is an α-tangle and x is an atom or near-litter that occurs in the range of $\mathsf{supp}(t)_A$, then $\iota(x) < \iota(t)$.

Proof. We use proposition 2.18 to construct the position function, using denied set

$$D(t) = \{\iota(N) \mid \mathsf{set}(t) = \mathsf{typed}_\alpha(N)\} \cup \{\iota(a) \mid a \in \operatorname{im} \mathsf{supp}(t)_A^{\mathcal{A}}\} \cup \{\iota(N) \mid N \in \operatorname{im} \mathsf{supp}(t)_A^{\mathcal{N}}\}$$

The first set is a subsingleton and the second two sets are small, so the denied set has size less than $\mathsf{cof}(\mathsf{ord}(\#\mu))$ as required. □

[1] It may be easier in practice to construct a position function on the product of the type of new t-sets and the type of α-supports, and then get the required position function on tangles from this.

Chapter 4

Freedom of action

4.1 Base approximations

Definition 4.1 (base approximation)**.** A *base approximation* is a pair $\psi = (\psi^{E\mathcal{A}}, \psi^{\mathcal{L}})$ such that $\psi^{E\mathcal{A}}$ and $\psi^{\mathcal{L}}$ are permutative relations of atoms and litters respectively (definition A.1), and for each litter L, the set

$$\mathsf{LS}(L) \cap \operatorname{coim} \psi^{E\mathcal{A}}$$

is small. The relation $\psi^{E\mathcal{A}}$ is called the *exceptional atom graph*, and $\psi^{\mathcal{L}}$ is called the *litter graph*. We make the following definitions.

- The *inverse* of a base approximation is $\psi^{-1} = ((\psi^{E\mathcal{A}})^{-1}, (\psi^{\mathcal{L}})^{-1})$.
- If ψ and χ are base approximations where $\operatorname{coim} \psi^{E\mathcal{A}} = \operatorname{coim} \chi^{E\mathcal{A}}$ and $\operatorname{coim} \psi^{\mathcal{L}} = \operatorname{coim} \chi^{\mathcal{L}}$, then their *composition* $\psi \circ \chi$ is the base approximation $(\psi^{E\mathcal{A}} \circ \chi^{E\mathcal{A}}, \psi^{\mathcal{L}} \circ \chi^{\mathcal{L}})$.
- The *ψ-sublitter* of a litter L, written L_ψ, is the near-litter $(L, \mathsf{LS}(L) \setminus \operatorname{coim} \psi^{E\mathcal{A}})$.

Definition 4.2 (atom graph of an approximation)**.** The *typical atom graph* of ψ is the relation $\psi^{T\mathcal{A}}$ given by the following constructor. If $(L_1, L_2) \in \psi^{\mathcal{L}}$, then

$$(h_{(L_1)_\psi}(i), h_{(L_2)_\psi}(i)) \in \psi^{T\mathcal{A}}$$

for some $i : \kappa$, where for any near-litter N, h_N is an equivalence $\kappa \simeq N$ chosen in advance.

The *atom graph* of ψ is the relation $\psi^{\mathcal{A}} = \psi^{E\mathcal{A}} \sqcup \psi^{T\mathcal{A}}$: the join of the exceptional and typical atom graphs.

Proposition 4.3. $(\psi^{T\mathcal{A}})^{-1} = (\psi^{-1})^{T\mathcal{A}}$ and hence $(\psi^{\mathcal{A}})^{-1} = (\psi^{-1})^{\mathcal{A}}$.

Proof. This follows directly from the fact that $L_\psi = L_{\psi^{-1}}$ for any litter L. □

Proposition 4.4. The graphs $\psi^{T\mathcal{A}}$ and $\psi^{\mathcal{A}}$ are permutative.

Proof. The typical atom graph is injective, because the equation $h_{L_\psi}(i)^\circ = L$ can be used to establish the the parameters of the relevant h maps coincide. Furthermore, we can use the fact that $\psi^{\mathcal{L}}$ has equal image and coimage to produce images of any image element of this relation. We then appeal to symmetry using proposition 4.3 to conclude that $\psi^{T\mathcal{A}}$ is permutative.

The (co)image of $\psi^{T\mathcal{A}}$ is

$$\bigcup_{L \in \operatorname{coim} \psi^{\mathcal{L}}} L_\psi = \bigcup_{L \in \operatorname{coim} \psi^{\mathcal{L}}} (\mathsf{LS}(L) \setminus \operatorname{coim} \psi^{E\mathcal{A}})$$

which is clearly disjoint from the coimage of $\psi^{E\mathcal{A}}$.[1] So $\psi^{\mathcal{A}}$ is permutative by one of the results of proposition A.2. □

Proposition 4.5. If ψ, χ have equal exceptional atom and litter coimages, then $(\psi \circ \chi)^{T\mathcal{A}} = \psi^{T\mathcal{A}} \circ \chi^{T\mathcal{A}}$.

Proof. Suppose that $(a_1, a_3) \in (\psi \circ \chi)^{T\mathcal{A}}$, so

$$a_1 = h_{(L_1)_{\psi\circ\chi}}(i); \quad a_3 = h_{(L_3)_{\psi\circ\chi}}(i); \quad (L_1, L_3) \in (\psi \circ \chi)^{\mathcal{L}}$$

Since $(\psi \circ \chi)^{\mathcal{L}} = \psi^{\mathcal{L}} \circ \chi^{\mathcal{L}}$, there is L_2 such that $(L_1, L_2) \in \chi^{\mathcal{L}}$ and $(L_2, L_3) \in \psi^{\mathcal{L}}$. Hence

$$(h_{(L_1)_\chi}(i), h_{(L_2)_\chi}(i)) \in \chi^{T\mathcal{A}}; \quad (h_{(L_2)_\psi}(i), h_{(L_3)_\psi}(i)) \in \psi^{T\mathcal{A}}$$

But $L_{\psi\circ\chi} = L_\psi = L_\chi$, so we obtain

$$(a_1, h_{(L_2)_\chi}(i)) \in \chi^{T\mathcal{A}}; \quad (h_{(L_2)_\chi}(i), a_3) \in \psi^{T\mathcal{A}}$$

For the converse, suppose that $(a_1, a_2) \in \chi^{T\mathcal{A}}$ and $(a_2, a_3) \in \psi^{T\mathcal{A}}$. Then

$$a_1 = h_{(L_1)_\chi}(i); \quad a_2 = h_{(L_2)_\chi}(i); \quad a_2 = h_{(L_2')_\psi}(j); \quad a_3 = h_{(L_3)_\psi}(j)$$

We obtain $L_2 = L_2'$, and $(L_2)_\chi = (L_2)_\psi$ so we also conclude $i = j$. Since $(L_1, L_2) \in \chi^{\mathcal{L}}$ and $(L_2, L_3) \in \psi^{\mathcal{L}}$, we conclude $(L_1, L_3) \in (\psi \circ \chi)^{\mathcal{L}}$, as required. □

Definition 4.6 (near-litter graph of an approximation)**.** The *near-litter graph* of ψ is the relation $\psi^{\mathcal{N}}$ given by setting $(N_1, N_2) \in \psi^{\mathcal{N}}$ if and only if $(N_1^\circ, N_2^\circ) \in \psi^{\mathcal{L}}$, N_1 and N_2 are subsets of $\operatorname{coim} \psi^{\mathcal{A}}$, and the image of $\psi^{\mathcal{A}}$ on N_1 is N_2 (or equivalently, by proposition A.2, the coimage of $\psi^{\mathcal{A}}$ on N_2 is N_1).

Proposition 4.7. Let s be a set of atoms near $\mathsf{LS}(L)$ for some litter L. If $(L, L') \in \psi^{\mathcal{L}}$, then the image of $\psi^{\mathcal{A}}$ on s is near $\mathsf{LS}(L')$.

Proof. We calculate

$$\begin{aligned}
\operatorname{im} \psi^{\mathcal{A}}|_s &= \operatorname{im} \psi^{\mathcal{A}}|_{\mathsf{LS}(L)} \mathbin{\triangle} \operatorname{im} \psi^{\mathcal{A}}|_{s \triangle \mathsf{LS}(L)} \\
&\overset{N}{\sim} \operatorname{im} \psi^{\mathcal{A}}|_{\mathsf{LS}(L)} \\
&= \operatorname{im} \psi^{\mathcal{A}}|_{\mathsf{LS}(L) \setminus \operatorname{coim} \psi^{E\mathcal{A}}} \cup \operatorname{im} \psi^{\mathcal{A}}|_{\mathsf{LS}(L) \cap \operatorname{coim} \psi^{E\mathcal{A}}} \\
&\overset{N}{\sim} \operatorname{im} \psi^{\mathcal{A}}|_{\mathsf{LS}(L) \setminus \operatorname{coim} \psi^{E\mathcal{A}}} \\
&= \operatorname{im} \psi^{T\mathcal{A}}|_{L_\psi} \\
&= L'_\psi \\
&\overset{N}{\sim} \mathsf{LS}(L')
\end{aligned}$$

□

[1] This result should of course be its own lemma.

Proposition 4.8. $(\psi^{-1})^{\mathcal{N}} = (\psi^{\mathcal{N}})^{-1}$, and $\psi^{\mathcal{N}}$ is permutative.

Proof. The first part follows from proposition 4.3. To show $\psi^{\mathcal{N}}$ is permutative, it suffices to show that it is injective and that its image is contained in its coimage; then, by taking inverses, the converses will also hold. Suppose that $(N_1, N_3), (N_2, N_3) \in \psi^{\mathcal{N}}$. Then the coimage of $\psi^{\mathcal{A}}$ on N_3 is equal to both N_1 and N_2, so $N_1 = N_2$, giving injectivity.

Now suppose that $(N_1, N_2) \in \psi^{\mathcal{N}}$. As $(N_1^\circ, N_2^\circ) \in \psi^{\mathcal{L}}$, we must have $(N_2^\circ, L) \in \psi^{\mathcal{L}}$ for some L. By proposition 4.7, the image s of $\psi^{\mathcal{A}}$ on N_2 is near $\mathsf{LS}(L)$, so (L, s) is a near-litter, and $(N_2, (L, s)) \in \psi^{\mathcal{N}}$ as required. □

Definition 4.9. Base approximations act on base supports in the following way. If $S^{\mathcal{A}} = (i, f)$, then $\psi(S)^{\mathcal{A}} = (i, f')$ where
$$f' = \{(j, a_2) \mid (j, a_1) \in f \wedge (a_1, a_2) \in \psi^{\mathcal{A}}\}$$
The same definition is used for near-litters.

4.2 Extensions of approximations

Definition 4.10. We define a partial order on base approximations by setting $\psi \leq \chi$ when $\psi^{E\mathcal{A}} = \chi^{E\mathcal{A}}$ and $\psi^{\mathcal{L}} \leq \chi^{\mathcal{L}}$.

Proposition 4.11 (adding orbits)**.** Let ψ be a base approximation, and let $L : \mathbb{N} \to \mathcal{L}$ be a function such that
$$L(m) = L(n) \to L(m+k) = L(n+k)$$
for all integers $m, n, k : \mathbb{Z}$. Suppose that for all n, $L(n) \notin \operatorname{coim} \psi^{\mathcal{L}}$. Then there is an extension $\chi \geq \psi$ such that $\chi^{\mathcal{L}}(L(n)) = L(n+1)$ and $\operatorname{coim} \chi^{\mathcal{L}} = \operatorname{coim} \psi^{\mathcal{L}} \cup \operatorname{ran} L$.

Proof. Define the relation
$$R = \{(L(n), L(n+1)) \mid n : \mathbb{Z}\}$$
This clearly has equal image and coimage. It is injective: if $(L_1, L_3), (L_2, L_3) \in R$, then there are $m, n : \mathbb{Z}$ such that
$$L_1 = L(m); \quad L_3 = L(m+1); \quad L_2 = L(n); \quad L_3 = L(n+1)$$
So $L(m+1) = L(n+1)$, giving $L_1 = L(m) = L(n) = L_2$ by substituting $k = -1$ in the hypothesis. It is also coinjective by substituting $k = 1$ in the hypothesis. So R is permutative. Therefore, $\psi^{\mathcal{L}} \sqcup R$ is a permutative relation, so $(\psi^{E\mathcal{A}}, \psi^{\mathcal{L}} \sqcup R)$ is an extension of ψ, and it clearly satisfies the result. □

4.3 Structural approximations

Definition 4.12. For a type index β, a β-*approximation* is a β-tree of base approximations. We define the partial order on β-approximations branchwise. We define an action of β-approximations ψ on β-supports S by $(\psi(S))_A = \psi_A(S_A)$.

Definition 4.13. Let A be a β-extended type index. A litter L is A-*inflexible* if there is an inflexible β-path I such that $A = ((A_I)_{\varepsilon_I})_\perp$ and $L = f_{\delta_I, \varepsilon_I}(t)$ for some $t : \mathsf{Tang}_{\delta_I}$. The coderivative operation works in the obvious way. A litter can be A-inflexible in at most one way.[2]

[2] We should make A-inflexibility into a subsingleton structure.

We say that a L is *A-flexible* if it is not A-inflexible.[3] If L is B_A-flexible, then L is A-flexible.

Definition 4.14. A β-approximation ψ is *coherent* at (A, L_1, L_2) if:

- If L_1 is A-inflexible with inflexible β-path $I = (\gamma, \delta, \varepsilon, B)$ and tangle $t : \mathsf{Tang}_\delta$, then there is some δ-allowable permutation ρ such that
$$(\psi_B)_\delta(\mathsf{supp}(t)) = \rho(\mathsf{supp}(t))$$
and
$$L_2 = f_{\delta,\varepsilon}(\rho(t))$$
(and hence all δ-allowable permutations ρ such that $(\psi_B)_\delta(\mathsf{supp}(t)) = \rho(\mathsf{supp}(t))$ satisfy $L_2 = f_{\delta,\varepsilon}(\rho(t))$).
- If L_1 is A-flexible, then L_2 is A-flexible.

We say that ψ is *coherent* if whenever $(L_1, L_2) \in \psi_A^{\mathcal{L}}$, ψ is coherent at (A, L_1, L_2).

Proposition 4.15 (adding orbits coherently). Suppose that ψ is an approximation and $L : \mathbb{Z} \to \mathcal{L}$ is a function satisfying the hypotheses of proposition 4.11. Let χ be the extension produced by the structural version of this result.[4] If ψ is coherent, and is additionally coherent at $(L(n), L(n+1))$ for each integer n, then χ is coherent.

Proof. This proof just relies on the fact that if $(\psi_B)_\delta(\mathsf{supp}(t)) = \rho(\mathsf{supp}(t))$, then the same holds for every extension χ of ψ.[5] □

Proposition 4.16. If ψ is coherent, then ψ^{-1} is coherent.

Proof. Suppose that $(L_1, L_2) \in (\psi_A^{-1})^{\mathcal{L}}$, so $(L_2, L_1) \in \psi_A^{\mathcal{L}}$. Suppose first that L_1 is A-inflexible with inflexible β-path $I = (\gamma, \delta, \varepsilon, B)$ and tangle $t : \mathsf{Tang}_\delta$. If L_2 were A-flexible, then L_1 would also be A-flexible by coherence, which is a contradiction. So L_2 is A-inflexible with path $(\gamma', \delta', \varepsilon', B')$ and tangle $t' : \mathsf{Tang}_{\delta'}$, and there is $\rho : \mathsf{AllPerm}_{\delta'}$ such that
$$(\psi_{B'})_{\delta'}(\mathsf{supp}(t')) = \rho(\mathsf{supp}(t'))$$
and
$$A = (B_{\varepsilon'})_\perp; \quad L_2 = f_{\delta',\varepsilon'}(t'); \quad L_1 = f_{\delta',\varepsilon'}(\rho(t'))$$
We thus deduce $\varepsilon = \varepsilon'$ and $\gamma = \gamma'$ by the equations for A. By the equation $L_1 = f_{\delta,\varepsilon}(t)$, we also obtain $\delta = \delta'$ and $t = \rho(t')$. Then we find
$$\begin{aligned}
(\psi_B)_\delta(\mathsf{supp}(\rho^{-1}(t))) &= \rho(\mathsf{supp}(\rho^{-1}(t))) \\
(\psi_B)_\delta(\rho^{-1}(\mathsf{supp}(t))) &= \rho(\rho^{-1}(\mathsf{supp}(t))) \\
(\psi_B)_\delta(\rho^{-1}(\mathsf{supp}(t))) &= \mathsf{supp}(t) \\
\rho^{-1}(\mathsf{supp}(t)) &= (\psi_B^{-1})_\delta(\mathsf{supp}(t))
\end{aligned}$$
where the last equation uses the fact that $(\psi_B)_\delta$ is defined on all of $\mathsf{supp}(t')$. Finally, the equation $L_2 = f_{\delta,\varepsilon}(\rho^{-1}(t))$ gives coherence at (A, L_1, L_2) as required.

Now suppose that L_1 is A-flexible. If L_2 were A-inflexible, then so would be L_1 by coherence. So L_2 is A-flexible, as required. □

[3] This is not data, but a proposition.

[4] We need the extension exactly as produced (as data), not an arbitrary extension satisfying the conclusion of the proposition.

[5] Maybe there's a better lemma to abstract out this idea for this and proposition 4.23?

Proposition 4.17. If ψ and χ are coherent and have equal coimages along all paths, then $\psi \circ \chi$ is coherent.

Proof. Suppose that $(L_1, L_3) \in ((\psi \circ \chi)_A)^{\mathcal{L}}$, so $(L_1, L_2) \in \psi_A^{\mathcal{L}}$ and $(L_2, L_3) \in \chi_A^{\mathcal{L}}$. Suppose that L_1 is A-inflexible with inflexible β-path $I = (\gamma, \delta, \varepsilon, B)$ and tangle $t : \mathsf{Tang}_\delta$. Then by coherence of ψ, we have ρ such that

$$(\psi_B)_\delta(\mathsf{supp}(t)) = \rho(\mathsf{supp}(t))$$

and

$$L_2 = f_{\delta,\varepsilon}(\rho(t))$$

Then L_2 is A-inflexible with path I and tangle $\rho(t)$. So by coherence of χ, we have ρ' such that

$$(\psi_B)_\delta(\mathsf{supp}(\rho(t))) = \rho'(\mathsf{supp}(\rho(t)))$$

and

$$L_3 = f_{\delta,\varepsilon}(\rho'(\rho(t)))$$

As $\rho'(\mathsf{supp}(\rho(t))) = \rho'(\rho(\mathsf{supp}(t)))$, we obtain the desired coherence result.

Instead, if L_1 is A-flexible, then so is L_2 by coherence of ψ, and so is L_3 by coherence of χ. □

Proposition 4.18. If ψ is a coherent β-approximation and A is a path $\beta \rightsquigarrow \beta'$, then ψ_A is a coherent β'-approximation.

Proof. Let $(L_1, L_2) \in (\psi_A)_B^{\mathcal{L}}$. Suppose that L_1 is B-inflexible with path $(\gamma, \delta, \varepsilon, C)$ and $t : \mathsf{Tang}_\delta$. Then L_1 is A_B-inflexible with path $(\gamma, \delta, \varepsilon, A_C)$ and the same tangle t. So by coherence of ψ, we obtain a δ-allowable ρ such that

$$(\psi_{(A_C)})_\delta(\mathsf{supp}(t)) = \rho(\mathsf{supp}(t))$$

and

$$L_2 = f_{\delta,\varepsilon}(\rho(t))$$

This same ρ can thus be used to establish coherence of ψ_A at (B, L_1, L_2).

Thus, by proposition 4.16, whenever L_2 is B-inflexible with path I and tangle t, L_1 is also B-inflexible with path I. So if L_1 is B-flexible, so is L_2, as required. □

4.4 Proving freedom of action

Definition 4.19 (approximates)**.** We say that a β-approximation ψ *approximates* a β-allowable permutation ρ if $\psi_A^{\mathcal{L}} \leq \rho_A^{\mathcal{L}}$ and $\psi_A^{\mathcal{A}} \leq \rho_A^{\mathcal{A}}$ for each path $A : \beta \rightsquigarrow \bot$. If ψ approximates ρ then ψ^n approximates ρ^n for each $n : \mathbb{Z}$.[6] A β-approximation ψ *exactly approximates* a β-allowable permutation ρ if ψ approximates ρ, and in addition, if a is an atom and $A : \beta \rightsquigarrow \bot$, then $a \notin \mathsf{coim}\,\psi_A^{\mathcal{A}}$ implies $\rho(a)^\circ = \rho(a^\circ)$ and $\rho^{-1}(a)^\circ = \rho^{-1}(a^\circ)$.

Definition 4.20 (freedom of action)**.** We say that *freedom of action* holds at a type index δ if every coherent δ-approximation exactly approximates some δ-allowable permutation.

Proposition 4.21 (adding flexible litters)**.** Let ψ be a coherent β-approximation, and let L be A-flexible. Then there is a coherent extension $\chi \geq \psi$ with $L \in \mathsf{coim}\,\chi_A^{\mathcal{L}}$.

[6] We should define what it means for a base approximation to approximate a near-litter permutation, and define this in terms of that.

Proof. Define $L' : \mathbb{Z} \to \mathcal{L}$ by $L'(n) = L$, then appeal to proposition 4.15 to obtain $\chi \geq \psi$. All we must do is check that ψ is coherent at (L, L), which is trivial. □

Proposition 4.22 (adding inflexible litters)**.** Let ψ be a coherent β-approximation, and let L be A-inflexible with path $(\gamma, \delta, \varepsilon, B)$ and tangle $t : \mathsf{Tang}_\delta$. Suppose that $(\psi_B)_\delta$ is defined on all of $\mathsf{supp}(t)$.[7] Suppose that freedom of action holds at level δ. Then there is a coherent extension $\chi \geq \psi$ with $L \in \mathsf{coim}\, \chi_A^{\mathcal{L}}$.

Proof. Let ρ be a δ-allowable permutation that $(\psi_B)_\delta$ approximates. Then for each $n : \mathbb{Z}$, as $(\psi_B^n)_\delta$ approximates ρ^n, we obtain $(\psi_B^n)_\delta(\mathsf{supp}(t)) = \rho^n(\mathsf{supp}(t))$ as $(\psi_B^n)_\delta$ is defined on all of $\mathsf{supp}(t)$.[8] Define $L : \mathbb{Z} \to \mathcal{L}$ by $L(n) = f_{\delta,\varepsilon}(\rho^n(t))$.

Suppose that there is some n such that $L(n) \in \mathsf{coim}\, \psi^{\mathcal{L}}$. Note that

$$\begin{aligned}
(\psi_B^n)_\delta(\mathsf{supp}(t)) &= \rho^n(\mathsf{supp}(t)) \\
\mathsf{supp}(t) &= (\psi_B^{-n})_\delta(\rho^n(\mathsf{supp}(t))) \\
\rho^{-n}(\rho^n(\mathsf{supp}(t))) &= (\psi_B^{-n})_\delta(\mathsf{supp}(\rho^n(t)))
\end{aligned}$$

So as ψ^{-n} is coherent, we obtain $(L(n), f_{\delta,\varepsilon}(t)) \in (\psi_A^{-n})^{\mathcal{L}}$. In particular, $f_{\delta,\varepsilon}(t) \in \mathsf{coim}\, \psi_A^{\mathcal{L}}$ already, and no work needs to be done.

We first check the hypothesis of proposition 4.11 for adding orbits. If $f_{\delta,\varepsilon}(\rho^m(t)) = f_{\delta,\varepsilon}(\rho^n(t))$, then $\rho^m(t) = \rho^n(t)$, so $\rho^{m+k}(t) = \rho^{n+k}(t)$, giving $f_{\delta,\varepsilon}(\rho^{m+k}(t)) = f_{\delta,\varepsilon}(\rho^{n+k}(t))$ as required.

We now check the criterion of proposition 4.15 for adding orbits coherently. It suffices to show that ψ is coherent at $(L(n), L(n+1))$ for each $n : \mathbb{Z}$. This is witnessed by ρ, which satisfies

$$(\psi_B)_\delta(\mathsf{supp}(\rho^n(t))) = \rho(\mathsf{supp}(\rho^n(t)))$$

and

$$L(n) = f_{\delta,\varepsilon}(\rho^n(t)); \quad L(n+1) = f_{\delta,\varepsilon}(\rho(\rho^n(t)))$$

as required.[9] □

Proposition 4.23. If $(\psi_i)_{i:I}$ is a chain of coherent approximations where I is a linear order, then the supremum ψ is coherent.

Proof. Direct, using the same idea as the proof of proposition 4.15. □

Theorem 4.24 (freedom of action)**.** Freedom of action holds at all type indices $\beta \leq \alpha$.

Proof. By induction, we may assume freedom of action holds at all $\delta < \beta$. Let ψ be a coherent β-approximation, and let χ be a maximal coherent extension, which exists by Zorn's lemma and proposition 4.23.

Suppose that there is a litter L such that there exists a path A where $L \notin \mathsf{coim}\, \chi_A^{\mathcal{L}}$. Let L have minimal position with this property, and let A be such a path.

---

[7]This is a nontrivial definition to make.

[8]This should of course be its own lemma.

[9]It might be helpful to abstract away the lemma $(\psi_B^m)_\delta(\mathsf{supp}(\rho^n(t))) = \mathsf{supp}(\rho^{n+m}(t))$ for the two places in the proof where this idea is used.

Suppose that L is A-flexible. Then by proposition 4.21, there is an extension φ of χ such that $L \in \operatorname{coim} \varphi_A^{\mathcal{L}}$, contradicting maximality of χ.

Suppose that L is A-inflexible, with path $(\gamma, \delta, \varepsilon, B)$ and tangle t. Then $(\psi_B)_\delta$ is defined on all of $\operatorname{supp}(t)$. Indeed, by definition 2.25 (coherent data) and proposition 2.22 (fuzz maps), for each atom or near-litter y that appears in the range of $\operatorname{supp}(t)_C$, we have $\iota(y) < \iota(t) < \iota(L)$, giving the desired conclusion by minimality of the position of L and the criteria of proposition 2.19. Thus, we obtain the same contradiction by proposition 4.22.

So $\operatorname{coim} \chi_A^{\mathcal{L}}$ is the set of all litters for each path A. We then use the fact that our model data is coherent to recursively compute the allowable permutation ρ with the same action as χ. Then χ exactly approximates ρ, so ψ also exactly approximates ρ.[10] □

4.5 Base actions

Definition 4.25. The *interference* of near-litters N_1, N_2 is

$$\operatorname{interf}(N_1, N_2) = \begin{cases} N_1 \mathbin{\triangle} N_2 & \text{if } N_1^\circ = N_2^\circ \\ N_1 \cap N_2 & \text{if } N_1^\circ \neq N_2^\circ \end{cases}$$

which is a small set of atoms.

Definition 4.26. A *base action* is a pair $\xi = (\xi^{\mathcal{A}}, \xi^{\mathcal{N}})$ such that $\xi^{\mathcal{A}}$ and $\xi^{\mathcal{N}}$ are relations of atoms and near-litters respectively (definition A.1), such that

- $\xi^{\mathcal{A}}$ and $\xi^{\mathcal{N}}$ are defined on small sets;
- $\xi^{\mathcal{A}}$ is one-to-one;
- if $(a_1, a_2) \in \xi^{\mathcal{A}}$ and $(N_1, N_2) \in \xi^{\mathcal{N}}$, then $a_1 \in N_1$ if and only if $a_2 \in N_2$;
- if $(N_1, N_3), (N_2, N_4) \in \xi^{\mathcal{N}}$, then $N_1^\circ = N_2^\circ$ if and only if $N_3^\circ = N_4^\circ$;
- for each $(N_1, N_3), (N_2, N_4) \in \xi^{\mathcal{N}}$,

$$\operatorname{interf}(N_1, N_2) \subseteq \operatorname{coim} \xi^{\mathcal{A}}; \quad \operatorname{interf}(N_3, N_4) \subseteq \operatorname{im} \xi^{\mathcal{A}}$$

Note that these conditions imply that $\xi^{\mathcal{N}}$ is one-to-one. We define the one-to-one relation $\xi^{\mathcal{L}}$ by the constructor

$$(N_1, N_2) \in \xi^{\mathcal{N}} \to (N_1^\circ, N_2^\circ) \in \xi^{\mathcal{L}}$$

The partial order on base actions is defined by $\xi \leq \zeta$ if and only if $\xi^{\mathcal{A}} \leq \zeta^{\mathcal{A}}$ and $\xi^{\mathcal{N}} = \zeta^{\mathcal{N}}$.[11] The inverse of ξ is $((\xi^{\mathcal{A}})^{-1}, (\xi^{\mathcal{N}})^{-1})$. They act on base supports in the natural way.

Definition 4.27. A base action ξ is *nice* if whenever $(N_1, N_2) \in \xi^{\mathcal{N}}$,

$$N_1 \mathbin{\triangle} \operatorname{LS}(N_1^\circ) \subseteq \operatorname{coim} \xi^{\mathcal{A}}; \quad N_2 \mathbin{\triangle} \operatorname{LS}(N_2^\circ) \subseteq \operatorname{im} \xi^{\mathcal{A}}$$

Proposition 4.28 (extending orbits inside near-litters)**.** Every base action ξ admits an extension ζ satisfying

$$\forall N \in \operatorname{coim} \xi^{\mathcal{N}}, N \setminus \operatorname{LS}(N^\circ) \subseteq \operatorname{coim} \xi^{\mathcal{A}}$$

[10] In general, if $\psi \leq \chi$ and χ (exactly) approximates ρ then ψ (exactly) approximates ρ.

[11] We should make utilities for constructing extensions of base actions, reducing the proof obligations of showing that these are base actions (e.g. removing the last two bullet points and not needing to prove results we already know about ξ).

Proof. For each litter L, there is an injection

$$i_L : \bigcup_{N \in \mathsf{coim}\, \xi^{\mathcal{N}}} (N \setminus \mathsf{LS}(N^\circ)) \to \{a : \mathcal{A} \mid a^\circ = L \wedge \forall N \in \mathsf{im}\, \xi^{\mathcal{N}}, N^\circ = L \to a \in N\} \setminus \mathsf{im}\, \xi^{\mathcal{A}}$$

by a cardinality argument. Define the relation R on atoms by the constructor

$$\forall (N_1, N_2) \in \xi^{\mathcal{N}}, \forall a \in N_1 \setminus \mathsf{LS}(N_1^\circ) \setminus \mathsf{coim}\, \xi^{\mathcal{A}}, (a, i_{N_2^\circ}(a)) \in R$$

This is one-to-one and has disjoint image and coimage from $\xi^{\mathcal{A}}$.

We now show that if $(a_1, a_2) \in R$ and $(N_1, N_2) \in \xi^{\mathcal{N}}$, then $a_1 \in N_1$ if and only if $a_2 \in N_2$. Let $(N_1, N_2), (N_1', N_2') \in \xi^{\mathcal{N}}$, and let $a \in N_1 \setminus \mathsf{LS}(N_1^\circ) \setminus \mathsf{coim}\, \xi^{\mathcal{A}}$. Suppose that $a \in N_1'$; we must show $i_{N_2^\circ}(a) \in N_2'$. If $N_1^\circ \neq {N_1'}^\circ$, then $\mathsf{interf}(N_1, N_1') = N_1 \cap N_1'$, so $a \in \mathsf{interf}(N_1, N_1') \subseteq \mathsf{coim}\, \xi^{\mathcal{A}}$, a contradiction. So $N_1^\circ = {N_1'}^\circ$, giving $N_2^\circ = {N_2'}^\circ$, so $i_{N_2^\circ}(a) = i_{{N_2'}^\circ}(a) \in N_2'$ by definition.

Conversely, suppose that $i_{N_2^\circ}(a) \in N_2'$; we must show $a \in N_1'$. Note that by definition, $i_{N_2^\circ}(a) \in N_2$. So if $N_2^\circ \neq {N_2'}^\circ$, we would have $i_{N_2^\circ}(a) \in N_2 \cap N_2' = \mathsf{interf}(N_2, N_2') \subseteq \mathsf{im}\, \xi^{\mathcal{A}}$, a contradiction. Hence $N_2^\circ = {N_2'}^\circ$ and $N_1^\circ = {N_1'}^\circ$. Thus, if $a \notin N_1'$, we would have $a \in N_1 \mathbin{\triangle} N_1' = \mathsf{interf}(N_1, N_1') \subseteq \mathsf{coim}\, \xi^{\mathcal{A}}$, again a contradiction.

Hence $\zeta = (\xi^{\mathcal{A}} \sqcup R, \xi^{\mathcal{N}})$ is a base action and satisfies the conclusion. □

Proposition 4.29 (extending orbits outside near-litters)**.** Every base action ξ admits an extension ζ satisfying

$$\forall N \in \mathsf{coim}\, \xi^{\mathcal{N}}, \mathsf{LS}(N) \setminus N \subseteq \mathsf{coim}\, \xi^{\mathcal{A}}$$

Proof. Without loss of generality (as extensions are transitive), let ξ satisfy the conclusion of proposition 4.28.

Let L be an arbitrary litter that whose litter set does not contain an atom in $\mathsf{im}\, \xi^{\mathcal{A}}$ or $\bigcup \mathsf{im}\, \xi^{\mathcal{N}}$. Define an injection

$$i : \bigcup_{N \in \mathsf{coim}\, \xi^{\mathcal{N}}} (\mathsf{LS}(N^\circ) \setminus N \setminus \mathsf{coim}\, \xi^{\mathcal{A}}) \to \mathsf{LS}(L)$$

by a cardinality argument. Note that i has domain disjoint from $\mathsf{coim}\, \xi^{\mathcal{A}}$ and image disjoint from $\mathsf{im}\, \xi^{\mathcal{A}}$.

We show that $(\xi^{\mathcal{A}} \sqcup \mathsf{graph}\, i, \xi^{\mathcal{N}})$ is a base action. It suffices to check that if $(a_1, a_2) \in \mathsf{graph}\, i$ and $(N_1, N_2) \in \xi^{\mathcal{N}}$, then $a_1 \in N_1$ if and only if $a_2 \in N_2$. As $a_2 \in \mathsf{LS}(L)$, we have $a_2 \notin N_2$. Suppose that $a_1 \in N_1$. We know that there is a near-litter $N \in \mathsf{coim}\, \xi^{\mathcal{N}}$ such that $a_1 \in \mathsf{LS}(N^\circ) \setminus N \setminus \mathsf{coim}\, \xi^{\mathcal{A}}$. If $N^\circ = N_1^\circ$, then $a_1 \in N \mathbin{\triangle} N_1 = \mathsf{interf}(N, N_1) \subseteq \mathsf{coim}\, \xi^{\mathcal{A}}$, a contradiction, hence $a^\circ = N^\circ \neq N_1^\circ$. But then as ξ satisfies the conclusion of proposition 4.28, we have $N_1 \setminus \mathsf{LS}(N_1^\circ) \subseteq \mathsf{coim}\, \xi^{\mathcal{A}}$, which again is a contradiction. □

Proposition 4.30. Every base action has a nice extension.

Proof. Apply proposition 4.28 to ξ to obtain ξ_1; apply proposition 4.28 again to ξ_1^{-1} to obtain ξ_2; apply proposition 4.29 to ξ_2 to obtain ξ_3, and finally apply proposition 4.29 again to ξ_3^{-1} to obtain ξ_4, our target. □

4.6 Structural actions

Definition 4.31. For a type index β, a *β-action* is a β-tree of base actions. We define an action of β-actions ξ on β-supports S by $(\xi(S))_A = \xi_A(S_A)$.

Definition 4.32. A β-action ξ is *coherent* at (A, L_1, L_2) if:

- If L_1 is A-inflexible with inflexible β-path $I = (\gamma, \delta, \varepsilon, B)$ and tangle $t : \mathsf{Tang}_\delta$, then there is some δ-allowable permutation ρ such that
$$(\xi_B)_\delta(\mathsf{supp}(t)) = \rho(\mathsf{supp}(t))$$
and
$$L_2 = f_{\delta,\varepsilon}(\rho(t))$$
(and hence again every δ-allowable ρ satisfying the hypothesis also satisfies the conclusion).
- If L_1 is A-flexible, then L_2 is A-flexible.

We say that ξ is *coherent* if whenever $(L_1, L_2) \in \xi_A^{\mathcal{L}}$, ξ is coherent at (A, L_1, L_2).

Definition 4.33. Let $A : \beta \rightsquigarrow \bot$. An *$A$-flexible approximation* of a base action ξ is a base approximation ψ such that

1. $\xi^{\mathcal{A}} \leq \psi^{E\mathcal{A}}$;
2. if $L \in \mathsf{coim}\,\psi^{\mathcal{L}}$, then L is A-flexible;
3. if $(N_1, N_2) \in \xi^{\mathcal{N}}$ and N_1° is A-flexible, then $(N_1^\circ, N_2^\circ) \in \psi^{\mathcal{L}}$;
4. if $(N_1, N_2) \in \xi^{\mathcal{N}}$, then $N_1 \bigtriangleup \mathsf{LS}(N_1^\circ) \subseteq \mathsf{coim}\,\psi^{\mathcal{A}}$ and $N_2 \bigtriangleup \mathsf{LS}(N_2^\circ) \subseteq \mathsf{coim}\,\psi^{\mathcal{A}}$;
5. if $(N_1, N_2) \in \xi^{\mathcal{N}}$, then for each atom a_2,
$$a_2 \in N_2 \leftrightarrow (\exists a_1 \in N_1,\ (a_1, a_2) \in \psi^{E\mathcal{A}}) \vee (a_2 \notin \mathsf{coim}\,\psi^{E\mathcal{A}} \wedge a_2^\circ = N_2^\circ)$$

A *flexible approximation* of a β-action ξ is a β-approximation ψ such that for each $A : \beta \rightsquigarrow \bot$, the base approximation ψ_A is an A-flexible approximation of ξ_A. Flexible approximations are coherent.

Proposition 4.34. Every base action has an A-flexible approximation. Hence, every β-action has a flexible approximation, which can be computed branchwise.

Proof. If $\xi \leq \zeta$ and ψ is an A-flexible approximation for ζ, then ψ is an A-flexible approximation for ξ. So it suffices to prove the result for nice base actions ξ by proposition 4.30.

Define the permutative relation $R : \mathcal{L} \to \mathcal{L} \to \mathsf{Prop}$ to be a permutative extension of $\xi^{\mathcal{L}}$, which exists by proposition A.5. Let π be the permutation of litters defined by R, or the identity on any litter not in $\mathsf{coim}\,R$.

Define an orbit restriction (t, f, π) (definition A.3) for field $\xi^{\mathcal{A}}$ by
$$u = \{a : \mathcal{A} \mid \forall N \in \mathsf{field}\,\xi^{\mathcal{N}},\ N^\circ = a^\circ \to a \in N\}; \quad t = u \setminus \mathsf{field}\,\xi^{\mathcal{A}}$$
with function $f : \mathcal{A} \to \mathcal{L}$ defined by $f(a) = a^\circ$, and litter permutation π. We must check that for each litter L, the set $t \cap \mathsf{LS}(L)$ has cardinality at least $\max(\aleph_0, \#\,\mathsf{field}\,\xi^{\mathcal{A}})$. But we can write
$$t \cap \mathsf{LS}(L) = \mathsf{LS}(L) \setminus \left(\mathsf{field}\,\xi^{\mathcal{A}} \cup \bigcup_{N \in \mathsf{field}\,\xi^{\mathcal{N}},\, N^\circ = L} (\mathsf{LS}(L) \setminus N) \right)$$

where the set being removed from $\mathsf{LS}(L)$ is small, so $t \cap \mathsf{LS}(L)$ is a large set, and $\aleph_0$ and $\#\,\mathsf{field}\,\xi^{\mathcal{A}}$ are less than $\#\kappa$, as required. Then by proposition A.4, there is a permutative relation $S \geq \xi^{\mathcal{A}}$ defined on a small set and contained in $\mathsf{field}\,\xi^{\mathcal{A}} \cup t = \mathsf{field}\,\xi^{\mathcal{A}} \cup u$, such that if

$$(a_1, a_2) \in S \to (a_1, a_2) \in \xi^{\mathcal{A}} \vee \pi(a_1^\circ) = a_2^\circ$$

Let T be a permutative extension of the restriction of $\xi^{\mathcal{L}}$ to the A-flexible litters, with coimage contained entirely in the set of A-flexible litters, given by proposition A.5. From this, we define a base approximation $\psi = (S, T)$.

It remains to check that ψ is an A-flexible approximation of ξ. Conditions 1–3 are trivial, and condition 4 follows from the fact that we assumed ξ was nice.

We first show an auxiliary result. Let $(N_1, N_2) \in \xi^{\mathcal{N}}$, and let $(a_1, a_2) \in S$; we will show that $a_1 \in N_1 \leftrightarrow a_2 \in N_2$. Suppose first that $(a_1, a_2) \in \xi^{\mathcal{A}}$, in which case we are done as ξ is a base action. Instead, we have $a_1 \notin \mathsf{coim}\,\xi^{\mathcal{A}}, a_2 \notin \mathsf{im}\,\xi^{\mathcal{A}}$ and $\pi(a_1^\circ) = a_2^\circ$. As ξ is nice, we must have $a_1 \in N_1 \leftrightarrow a_1^\circ = N_1^\circ$. Similarly, $a_2 \in N_2 \leftrightarrow a_2^\circ = N_2^\circ$. So if $a_1 \in N_1$, we conclude that $a_2^\circ = \pi(a_1^\circ) = \pi(N_1^\circ) = N_2^\circ$ giving $a_2 \in N_2$, and if $a_2 \in N_2$, we find $\pi(a_1^\circ) = a_2^\circ = N_2^\circ$ so $a_1^\circ = N_1^\circ$, giving $a_1 \in N_1$.

We now prove condition 5, which is the equation

$$a_2 \in N_2 \leftrightarrow (\exists a_1 \in N_1,\ (a_1, a_2) \in S) \vee (a_2 \notin \mathsf{coim}\,S \wedge a_2^\circ = N_2^\circ)$$

where $(N_1, N_2) \in \xi^{\mathcal{N}}$. Consider the first the case where $(a_1, a_2) \in S$ and $a_1 \in N_1$. The auxiliary lemma shows that $a_2 \in N_2$ as required. Now consider the case where $a_2 \notin \mathsf{coim}\,S$ and $a_2^\circ = N_2^\circ$. If $a_2 \notin N_2$, then $a_2 \in N_2 \mathbin{\triangle} \mathsf{LS}(N_2^\circ) \subseteq \mathsf{im}\,\xi^{\mathcal{A}}$, a contradiction. Finally suppose that neither holds, so

$$(\forall a_1,\ (a_1, a_2) \in S \to a_1 \notin N_1) \wedge (a_2 \in \mathsf{coim}\,S \vee a_2^\circ \neq N_2^\circ)$$

If $a_2 \in \mathsf{coim}\,S$, then there is a_1 such that $(a_1, a_2) \in S$, and we have $a_1 \notin N_1$, giving $a_2 \notin N_2$ by the auxiliary lemma. Finally, if $a_2 \notin \mathsf{coim}\,S$ and $a_2^\circ \neq N_2^\circ$, then $a_2 \notin N_2$, since $a_2 \in N_2$ would imply $a_2 \in N_2 \mathbin{\triangle} \mathsf{LS}(N_2^\circ)$, contradicting the fact that ξ is nice. □

Definition 4.35 (approximates)**.** We say that a β-action ξ *approximates* a β-allowable permutation ρ if $\xi_A^{\mathcal{N}} \leq \rho_A^{\mathcal{N}}$ and $\xi_A^{\mathcal{A}} \leq \rho_A^{\mathcal{A}}$ for each path $A : \beta \rightsquigarrow \bot$.[12]

Proposition 4.36. Let ξ be a base action, and let ψ be an A-flexible approximation of it. Let π be a base permutation that ψ exactly approximates. If $(N_1, N_2) \in \xi^{\mathcal{N}}$ and $\pi(N_1^\circ) = N_2^\circ$, then $\pi(N_1) = N_2$.

Proof. First, note that

$$\begin{aligned}\pi[N_1] &= \pi[\mathsf{LS}(N_1^\circ)] \mathbin{\triangle} \pi[N_1 \mathbin{\triangle} \mathsf{LS}(N_1^\circ)] \\ &= (\pi[\mathsf{LS}(N_1^\circ) \cap \mathsf{coim}\,\psi^{E\mathcal{A}}] \cup \pi[\mathsf{LS}(N_1^\circ) \setminus \mathsf{coim}\,\psi^{E\mathcal{A}}]) \mathbin{\triangle} \pi[N_1 \mathbin{\triangle} \mathsf{LS}(N_1^\circ)]\end{aligned}$$

As ψ exactly approximates π and $\pi(N_1^\circ) = N_2^\circ$, we have the equation

$$\pi[\mathsf{LS}(N_1^\circ) \setminus \mathsf{coim}\,\psi^{E\mathcal{A}}] = \mathsf{LS}(N_2^\circ) \setminus \mathsf{coim}\,\psi^{E\mathcal{A}}$$

[12] Again, we should define what it means for a base action to approximate a near-litter permutation, and define this in terms of that.

Combining this with the fact that $\psi^{E\mathcal{A}} \leq \pi^{\mathcal{A}}$, and that $N_1 \triangle \mathsf{LS}(N_1^\circ) \subseteq \mathsf{coim}\,\psi^{E\mathcal{A}}$, we obtain

$$\begin{aligned}\pi[N_1] &= (\mathsf{im}\,\psi^{E\mathcal{A}}|_{\mathsf{LS}(N_1^\circ)\cap\mathsf{coim}\,\psi^{E\mathcal{A}}} \cup (\mathsf{LS}(N_2^\circ) \setminus \mathsf{coim}\,\psi^{E\mathcal{A}})) \triangle \mathsf{im}\,\psi^{E\mathcal{A}}|_{N_1 \triangle \mathsf{LS}(N_1^\circ)} \\ &= (\mathsf{im}\,\psi^{E\mathcal{A}}|_{\mathsf{LS}(N_1^\circ)\cap\mathsf{coim}\,\psi^{E\mathcal{A}}} \triangle \mathsf{im}\,\psi^{E\mathcal{A}}|_{N_1 \triangle \mathsf{LS}(N_1^\circ)}) \cup (\mathsf{LS}(N_2^\circ) \setminus \mathsf{coim}\,\psi^{E\mathcal{A}}) \\ &= \mathsf{im}\,\psi^{E\mathcal{A}}|_{(\mathsf{LS}(N_1^\circ)\cap\mathsf{coim}\,\psi^{E\mathcal{A}})\triangle(N_1 \triangle \mathsf{LS}(N_1^\circ))} \cup (\mathsf{LS}(N_2^\circ) \setminus \mathsf{coim}\,\psi^{E\mathcal{A}}) \\ &= \mathsf{im}\,\psi^{E\mathcal{A}}|_{N_1\cap\mathsf{coim}\,\psi^{E\mathcal{A}}} \cup (\mathsf{LS}(N_2^\circ) \setminus \mathsf{coim}\,\psi^{E\mathcal{A}})\end{aligned}$$

which is equal to N_2 by part of definition 4.33. □

Proposition 4.37. Let ξ be a coherent β-action, and let ψ be a flexible approximation for it. If ψ exactly approximates some allowable permutation ρ, then ξ approximates ρ.

Proof. First, note that $\xi_A^{\mathcal{A}} \leq \psi_A^{E\mathcal{A}}$ and $\psi_A^{\mathcal{A}} \leq \rho_A^{\mathcal{A}}$ give the required result for atoms. Now suppose that $(N_1, N_2) \in \xi_A^{\mathcal{N}}$; we must show that $\rho_A(N_1) = N_2$. We prove this by induction on $\iota(N_1)$, generalising over all A.

By proposition 4.36, it suffices to show that $\rho_A(N_1^\circ) = N_2^\circ$. Suppose that N_1° is A-flexible. Then by definition 4.33, $(N_1^\circ, N_2^\circ) \in \psi_A^{\mathcal{L}}$. Hence $\rho_A(N_1^\circ) = N_2^\circ$ as required.

Suppose not, so N_1° is A-inflexible with path $(\gamma, \delta, \varepsilon, B)$ and tangle $t : \mathsf{Tang}_\delta$. By coherence of ξ, we know that $(\xi_B)_\delta$ is defined on $\mathsf{supp}(t)$, and it suffices to show that

$$(\xi_B)_\delta(\mathsf{supp}(t)) = (\rho_B)_\delta(\mathsf{supp}(t))$$

Let $C : \delta \rightsquigarrow \bot$ and a be an atom such that $(i, a) \in \mathsf{supp}(t)_C^{\mathcal{A}}$ for some i. Then $(a, ((\rho_B)_\delta)_C(a)) \in ((\xi_B)_\delta)_C^{\mathcal{A}}$ by the result for atoms. Now suppose N is a near-litter such that $(i, N) \in \mathsf{supp}(t)_C^{\mathcal{N}}$. Then

$$\iota(N) < \iota(t) < \iota(f_{\delta,\varepsilon}(t)) = \iota(N_1^\circ)$$

So we may apply the inductive hypothesis, giving $(N, ((\rho_B)_\delta)_C(N)) \in ((\xi_B)_\delta)_C^{\mathcal{N}}$ as required. □

Theorem 4.38 (freedom of action for actions)**.** Every coherent action approximates some allowable permutation.

Proof. Let ξ be a coherent β-action, and let ψ be a flexible approximation for it, which exists by proposition 4.34. Then apply theorem 4.24 (freedom of action) to ψ to obtain a β-allowable permutation ρ that ψ exactly approximates. Finally, appeal to proposition 4.37 to conclude that ξ approximates ρ. □

Chapter 5

The counting argument

5.1 Strong supports

Definition 5.1. We define a preorder $\preceq$ on base supports by $S \preceq T$ if and only if $\operatorname{im} S^{\mathcal{A}} \subseteq \operatorname{im} T^{\mathcal{A}}$ and $\operatorname{im} S^{\mathcal{N}} \subseteq \operatorname{im} T^{\mathcal{N}}$. For β-supports, we define $S \preceq T$ if and only if $S_A \preceq T_A$ for each A.

Definition 5.2. A β-support S is *strong* if:

- for every pair of near-litters $N_1, N_2 \in \operatorname{im} S_A^{\mathcal{N}}$, we have $\operatorname{interf}(N_1, N_2) \subseteq \operatorname{im} S_A^{\mathcal{A}}$; and
- for every inflexible path $I = (\gamma, \delta, \varepsilon, A)$ and $t : \mathsf{Tang}_\delta$, if there is a near-litter $N \in \operatorname{im} S_{A_{\varepsilon\bot}}^{\mathcal{N}}$ with $N^\circ = f_{\delta,\varepsilon}(t)$, then $\operatorname{supp}(t) \preceq S_{A_\delta}$.

Proposition 5.3. If S is a strong β-support and ρ is β-allowable, then $\rho(S)$ is strong.

Proof. Interference is stable under application of allowable permutations, and the required supports are also preserved under action of allowable permutations. □

Proposition 5.4. For every support S, there is a strong support $T \succeq S$.

Proof. We define a relation R on pairs (A, N) where $A : \beta \rightsquigarrow \bot$ and N is a near-litter by the following constructor. If $I = (\gamma, \delta, \varepsilon, A)$ and $t : \mathsf{Tang}_\delta$, then for any near-litter N_1 such that $N_1^\circ = f_{\delta,\varepsilon}(t)$ and any path $B : \delta \rightsquigarrow \bot$ and near-litter $N_2 \in \operatorname{supp}(t)_B^{\mathcal{N}}$, we define $((A_\delta)_B, N_2)\ R\ ((A_\varepsilon)_\bot, N_1)$. This is well-founded, because if $(B, N_2)\ R\ (A, N_1)$ then $\iota(N_2) < \iota(N_1)$. For any small set s of such pairs, the transitive closure of s under R is small.

Let S be a support, and let s be the transitive closure of the set of pairs (A, N) such that $N \in \operatorname{im} S_A^{\mathcal{N}}$. Generate a support T from S and s using the fact that every small set is the range of some enumeration. This satisfies the second condition of being a strong support.

Now, for any base support T, we define its *interference support* to be a base support U consisting of just the atoms in the interference of all near-litters that appear in T. We may extend this definition to structural supports.

Finally, if U is the interference support of the T defined above, the support $T + U$ is strong, and since $S \succeq T$, we conclude $S \preceq T + U$. □

5.2 Coding functions

Definition 5.5. For a type index $\beta \leq \alpha$, a *β-support orbit* is the quotient of $\mathsf{StrSupp}_\beta$ under the relation of being in the same orbit under β-allowable permutations.[1]

Definition 5.6. For any type index $\beta \leq \alpha$, a *β-coding function* is a relation $\chi : \mathsf{StrSupp}_\beta \to \mathsf{TSet}_\beta \to \mathsf{Prop}$ such that:

- χ is coinjective;
- χ is nonempty;
- if $(S, x) \in \chi$, then S is a support for x;
- if $S, T \in \mathsf{coim}\,\chi$ then S and T are in the same support orbit;
- if $(S, x) \in \chi$ and ρ is β-allowable, then $(\rho(S), \rho(x)) \in \chi$.

Proposition 5.7 (extensionality for coding functions)**.** Let χ_1, χ_2 be β-coding functions. If $(S, x) \in \chi_1, \chi_2$, then $\chi_1 = \chi_2$.

Proof. We show $\chi_1 \subseteq \chi_2$; the result then follows by antisymmetry. Suppose $(T, y) \in \chi_1$. Then $T = \rho(S)$ for some β-allowable ρ. As $(\rho(S), \rho(x)) \in \chi_1$ and χ_1 is coinjective, we obtain $y = \rho(x)$. Hence $(T, y) \in \chi_2$ as required. □

Definition 5.8. Let $t : \mathsf{Tang}_\beta$. Then we define the coding function χ_t by the constructor

$$\forall \rho : \mathsf{AllPerm}_\beta,\ (\rho(\mathsf{supp}(t)), \rho(\mathsf{set}(t))) \in \chi$$

This is clearly a coding function, and satisfies $(\mathsf{supp}(t), \mathsf{set}(t)) \in \chi_t$.

Proposition 5.9. Let $t, u : \mathsf{Tang}_\beta$. Then $\chi_t = \chi_u$ if and only if there is a β-allowable ρ with $\rho(t) = u$.

Proof. If $\rho(t) = u$, then $(\mathsf{supp}(t), \mathsf{set}(t)) \in \chi_t$ implies $(\mathsf{supp}(u), \mathsf{set}(u)) \in \chi_t$, giving $\chi_t = \chi_u$ by proposition 5.7. Conversely if $\chi_t = \chi_u$, then $(\mathsf{supp}(u), \mathsf{set}(u)) \in \chi_t$, so there is ρ such that $\rho(\mathsf{supp}(t)) = \mathsf{supp}(u)$, and $(\rho(\mathsf{supp}(t)), \rho(\mathsf{set}(t))) \in \chi_t$, so by coinjectivity we obtain $\rho(set(t)) = \mathsf{set}(u)$ as required. □

5.3 Specifications

Definition 5.10. An *atom condition* is a pair (s, t) where $s, t : \mathsf{Set}\,\kappa$. A *$\beta$-near-litter condition* is either

- a *flexible condition*, consisting of a set $s : \mathsf{Set}\,\kappa$; or
- an *inflexible condition*, consisting of an inflexible β-path $I = (\gamma, \delta, \varepsilon, A)$, a δ-coding function χ, and two δ-trees $R^{\mathcal{A}}, R^{\mathcal{N}}$ of relations on κ.

A *β-specification* is a pair $(\sigma^{\mathcal{A}}, \sigma^{\mathcal{N}})$ where

- $\sigma^{\mathcal{A}}$ is a β-tree of enumerations of atom conditions; and
- $\sigma^{\mathcal{N}}$ is a β-tree of enumerations of β-near-litter conditions.

[1]This can be implemented using `MulAction.orbitRel.Quotient`. We need to make sure there's plenty of API for support orbits to avoid code duplication.

Definition 5.11. Let S be a β-support. Then *its specification* is the β-specification $\sigma = \mathsf{spec}(S)$ given by the following constructors.

- Whenever $(i, a) \in S_A^{\mathcal{A}}$, we have $(i, (s, t)) \in \sigma_A^{\mathcal{A}}$ where
$$s = \{j : \kappa \mid (j, a) \in S_A^{\mathcal{A}}\}; \quad t = \{j : \kappa \mid \exists N, (j, N) \in S_A^{\mathcal{N}} \wedge a \in N\}$$
- Whenever $(i, N) \in S_A^{\mathcal{N}}$ and N° is A-flexible, we have $(i, c) \in \sigma_A^{\mathcal{N}}$ where c is the flexible condition given by
$$s = \{j : \kappa \mid \exists N', (j, N') \in S_A^{\mathcal{N}} \wedge N^\circ = N'^\circ\}$$
- Whenever $I = (\gamma, \delta, \varepsilon, A)$ is an inflexible β-path, $t : \mathsf{Tang}_\delta$, and $(i, N) \in S_{A_{\varepsilon\bot}}^{\mathcal{N}}$ is such that $N^\circ = f_{\delta,\varepsilon}(t)$, we have $(i, c) \in \sigma_{A_{\varepsilon\bot}}^{\mathcal{N}}$ where c is the inflexible condition given by path I and coding function χ_t, and $R^{\mathcal{A}}$ and $R^{\mathcal{N}}$ are the δ-trees of relations given by the constructors
$$\forall i, \forall j, \forall a, (i, a) \in S_{A_{\delta B}}^{\mathcal{A}} \to (j, a) \in \mathsf{supp}(t)_B^{\mathcal{A}} \to (i, j) \in R_B^{\mathcal{A}}$$
$$\forall i, \forall j, \forall N, (i, N) \in S_{A_{\delta B}}^{\mathcal{N}} \to (j, N) \in \mathsf{supp}(t)_B^{\mathcal{N}} \to (i, j) \in R_B^{\mathcal{N}}$$

Proposition 5.12. Let S, T be β-supports. Then $\mathsf{spec}(S) = \mathsf{spec}(T)$ if and only if[2]

- $\mathsf{coim}\, S_A^{\mathcal{A}} = \mathsf{coim}\, T_A^{\mathcal{A}}$ and $\mathsf{coim}\, S_A^{\mathcal{N}} = \mathsf{coim}\, T_A^{\mathcal{N}}$.
- (atom condition) Whenever $(i, a_1) \in S_A^{\mathcal{A}}$ and $(i, a_2) \in T_A^{\mathcal{A}}$, we have
$$\forall j, (j, a_1) \in S_A^{\mathcal{A}} \leftrightarrow (j, a_2) \in T_A^{\mathcal{A}}$$
and
$$\forall j, (\exists N, (j, N) \in S_A^{\mathcal{N}} \wedge a_1 \in N) \leftrightarrow (\exists N, (j, N) \in T_A^{\mathcal{N}} \wedge a_2 \in N)$$
- (flexible condition) Whenever $(i, N_1) \in S_A^{\mathcal{N}}$ and $(i, N_2) \in T_A^{\mathcal{N}}$, if N_1° is A-flexible, then so is N_2°, and
$$\forall j, (\exists N', (j, N') \in S_A^{\mathcal{N}} \wedge N_1^\circ = N'^\circ) \leftrightarrow (\exists N', (j, N') \in T_A^{\mathcal{N}} \wedge N_2^\circ = N'^\circ)$$
- (inflexible condition) Let $I = (\gamma, \delta, \varepsilon)$ be an inflexible β-path and let $t : \mathsf{Tang}_\delta$. Then whenever $(i, N_1) \in S_{A_{\varepsilon\bot}}^{\mathcal{N}}$ and $(i, N_2) \in T_{A_{\varepsilon\bot}}^{\mathcal{N}}$ are such that $N_1^\circ = f_{\delta,\varepsilon}(t)$, there is some δ-allowable permutation ρ such that $N_2^\circ = f_{\delta,\varepsilon}(\rho(t))$, and
$$\forall j, \forall a, (j, a) \in \mathsf{supp}(t)_B^{\mathcal{A}} \to \forall i, (i, a) \in S_{A_{\delta B}}^{\mathcal{A}} \leftrightarrow (i, \rho_B(a)) \in T_{A_{\delta B}}^{\mathcal{A}}$$
$$\forall j, \forall N, (j, N) \in \mathsf{supp}(t)_B^{\mathcal{N}} \to \forall i, (i, N) \in S_{A_{\delta B}}^{\mathcal{N}} \leftrightarrow (i, \rho_B(N)) \in T_{A_{\delta B}}^{\mathcal{N}}$$

Proof. We will only sketch the fourth bullet point of this proof; the remainder is direct (but quite long to write down on paper). Moreover, we will show this for atoms; the result for near-litters is identical. The specifications $\mathsf{spec}(S)$ and $\mathsf{spec}(T)$ give rise to the same trees $R^{\mathcal{A}}$ precisely when
$$\forall i, \forall j, (\exists a, (i, a) \in S_{A_{\delta B}}^{\mathcal{A}} \wedge (j, a) \in \mathsf{supp}(t)_B^{\mathcal{A}}) \leftrightarrow (\exists a, (i, a) \in T_{A_{\delta B}}^{\mathcal{A}} \wedge (j, a) \in \mathsf{supp}(\rho(t))_B^{\mathcal{A}})$$
We must show that this holds if and only if
$$\forall j, \forall a, (j, a) \in \mathsf{supp}(t)_B^{\mathcal{A}} \to \forall i, (i, a) \in S_{A_{\delta B}}^{\mathcal{A}} \leftrightarrow (i, \rho_B(a)) \in T_{A_{\delta B}}^{\mathcal{A}}$$
This can be concluded by appealing to the basic behaviour of ρ and noting the coinjectivity of $\mathsf{supp}(t)_B^{\mathcal{A}}$. □

[2]The following bullet points should comprise a proposition type relating S and T.

Proposition 5.13. Let ρ be β-allowable, and let S be a β-support. Then $\mathsf{spec}(\rho(S)) = \mathsf{spec}(S)$.

Proof. We appeal to proposition 5.12. Clearly the coimage condition holds.

For the atom condition, we must check that if $(i, a) \in S_A^{\mathcal{A}}$, we have

$$\forall j,\ (j, a) \in S_A^{\mathcal{A}} \leftrightarrow (j, \rho_A(a)) \in \rho(S)_A^{\mathcal{A}}$$

and

$$\forall j,\ (\exists N,\ (j, N) \in S_A^{\mathcal{N}} \wedge a \in N) \leftrightarrow (\exists N,\ (j, N) \in \rho(S)_A^{\mathcal{N}} \wedge \rho_A(a) \in N)$$

both of which are trivial.

For the flexible condition, we must check that if $(i, N) \in S_A^{\mathcal{N}}$ and N° is A-flexible, then $\rho_A(N)^\circ$ is also A-flexible (which is direct, and should already be its own lemma), and that

$$\forall j,\ (\exists N',\ (j, N') \in S_A^{\mathcal{N}} \wedge N^\circ = N'^\circ) \leftrightarrow (\exists N',\ (j, N') \in \rho(S)_A^{\mathcal{N}} \wedge \rho(N)^\circ = N'^\circ)$$

which is similarly trivial.

Finally, for the inflexible condition, suppose that $I = (\gamma, \delta, \varepsilon)$ is an inflexible β-path and $t : \mathsf{Tang}_\delta$. Let $(i, N) \in S_{A_{\varepsilon\perp}}^{\mathcal{N}}$ be such that $N^\circ = f_{\delta,\varepsilon}(t)$. Then by the coherence condition,

$$\rho_{A_{\varepsilon\perp}}(N)^\circ = \rho_{A_{\varepsilon\perp}}(N^\circ) = \rho_{A_{\varepsilon\perp}}(f_{\delta,\varepsilon}(t)) = f_{\delta,\varepsilon}(\rho_{A_\delta}(t))$$

We must check that

$$\begin{gathered}\forall j,\ \forall a,\ (j, a) \in \mathsf{supp}(t)_B^{\mathcal{A}} \to \forall i,\ (i, a) \in S_{A_{\delta B}}^{\mathcal{A}} \leftrightarrow (i, (\rho_{A_\delta})_B(a)) \in \rho(S)_{A_{\delta B}}^{\mathcal{A}} \\ \forall j,\ \forall N,\ (j, N) \in \mathsf{supp}(t)_B^{\mathcal{N}} \to \forall i,\ (i, N) \in S_{A_{\delta B}}^{\mathcal{N}} \leftrightarrow (i, (\rho_{A_\delta})_B(N)) \in \rho(S)_{A_{\delta B}}^{\mathcal{N}}\end{gathered}$$

which again is trivial. $\square$

Definition 5.14. Let S and T be base supports. We define the relations $\mathsf{conv}_{S,T}^{\mathcal{A}}, \mathsf{conv}_{S,T}^{\mathcal{N}}$ by the constructors[3]

$$\begin{gathered}(i, a_1) \in S^{\mathcal{A}} \to (i, a_2) \in T^{\mathcal{A}} \to (a_1, a_2) \in \mathsf{conv}_{S,T}^{\mathcal{A}} \\ (i, N_1) \in S^{\mathcal{N}} \to (i, N_2) \in T^{\mathcal{N}} \to (N_1, N_2) \in \mathsf{conv}_{S,T}^{\mathcal{N}}\end{gathered}$$

Note that ${\mathsf{conv}_{S,T}^{\mathcal{A}}}^{-1} = \mathsf{conv}_{T,S}^{\mathcal{A}}$ and ${\mathsf{conv}_{S,T}^{\mathcal{N}}}^{-1} = \mathsf{conv}_{T,S}^{\mathcal{N}}$.

Proposition 5.15. Let S, T be supports such that $\mathsf{spec}(S) = \mathsf{spec}(T)$. Then $\mathsf{conv}_{S_A,T_A}^{\mathcal{A}}$ is one-to-one.

Proof. If $(a_1, a_2), (a_1, a_3) \in \mathsf{conv}_{S_A,T_A}^{\mathcal{A}}$, then there are i, j such that $(i, a_1), (j, a_1) \in S_A^{\mathcal{A}}$ and $(i, a_2), (j, a_3) \in T_A^{\mathcal{A}}$. By proposition 5.12, we deduce $(j, a_1) \in S_A^{\mathcal{A}} \leftrightarrow (j, a_2) \in T_A^{\mathcal{A}}$, so by coinjectivity of $T_A^{\mathcal{A}}$, we deduce $a_2 = a_3$. Hence $\mathsf{conv}_{S_A,T_A}^{\mathcal{A}}$ is coinjective. By symmetry, $\mathsf{conv}_{S_A,T_A}^{\mathcal{A}}$ is one-to-one. $\square$

Proposition 5.16. Let S, T be supports such that $\mathsf{spec}(S) = \mathsf{spec}(T)$. If $(a_1, a_2) \in \mathsf{conv}_{S_A,T_A}^{\mathcal{A}}$ and $(N_1, N_2) \in \mathsf{conv}_{S_A,T_A}^{\mathcal{N}}$, then $a_1 \in N_1$ if and only if $a_2 \in N_2$.

[3] This should be abstracted out even further; this can be defined for any pair of relations with common domain.

Proof. As $(a_1, a_2) \in \mathsf{conv}^{\mathcal{A}}_{S_A,T_A}$, there is i such that $(i, a_1) \in S^{\mathcal{A}}_A$ and $(i, a_2) \in T^{\mathcal{A}}_A$, and as $(N_1, N_2) \in \mathsf{conv}^{\mathcal{N}}_{S_A,T_A}$, there is j such that $(j, N_1) \in S^{\mathcal{N}}_A$ and $(j, N_2) \in T^{\mathcal{N}}_A$. By proposition 5.12, we deduce that

$$\forall j,\ (\exists N,\ (j, N) \in S^{\mathcal{N}}_A \wedge a_1 \in N) \leftrightarrow (\exists N,\ (j, N) \in T^{\mathcal{N}}_A \wedge a_2 \in N)$$

If $a_1 \in N_1$, then as $(j, N_1) \in S^{\mathcal{N}}_A$, we deduce that there is a near-litter N' such that $(j, N') \in T^{\mathcal{N}}_A$ and $a \in N'$. But $T^{\mathcal{N}}_A$ is coinjective, so $N' = N_2$, giving $a \in N_2$. The converse holds by symmetry. □

Proposition 5.17. Let S, T be supports such that T is strong and $\mathsf{spec}(S) = \mathsf{spec}(T)$. If $(N_1, N_3), (N_2, N_4) \in \mathsf{conv}^{\mathcal{N}}_{S_A,T_A}$, then $N_1^\circ = N_2^\circ$ if and only if $N_3^\circ = N_4^\circ$.

Proof. There are i, j such that $(i, N_1), (j, N_2) \in S^{\mathcal{N}}_A$ and $(i, N_3), (j, N_4) \in T^{\mathcal{N}}_A$. Suppose that $N_1^\circ = N_2^\circ$; we show that $N_3^\circ = N_4^\circ$.

First, suppose that N_1° is A-flexible. Then by proposition 5.12, we have

$$\forall j,\ (\exists N',\ (j, N') \in S^{\mathcal{N}}_A \wedge N_1^\circ = N'^\circ) \leftrightarrow (\exists N',\ (j, N') \in T^{\mathcal{N}}_A \wedge N_3^\circ = N'^\circ)$$

So as $(j, N_2) \in S^{\mathcal{N}}_A$ and $N_1^\circ = N_2^\circ$, there is N' with $(j, N') \in T^{\mathcal{N}}_A$ and $N_3^\circ = N'^\circ$, but clearly $N' = N_4$, giving the result.

Now suppose that N_1° is A-inflexible, so there is an inflexible β-path $(\gamma, \delta, \varepsilon, B)$ and tangle $t : \mathsf{Tang}_\delta$ such that

$$A = B_{\varepsilon\bot};\quad N_1^\circ = f_{\delta,\varepsilon}(t)$$

Then by proposition 5.12, there is some δ-allowable ρ such that $N_3^\circ = f_{\delta,\varepsilon}(\rho(t))$ and

$$\forall j,\ \forall a,\ (j, a) \in \mathsf{supp}(t)^{\mathcal{A}}_B \to \forall i,\ (i, a) \in S^{\mathcal{A}}_{A_{\delta B}} \leftrightarrow (i, \rho_B(a)) \in T^{\mathcal{A}}_{A_{\delta B}}$$
$$\forall j,\ \forall N,\ (j, N) \in \mathsf{supp}(t)^{\mathcal{N}}_B \to \forall i,\ (i, N) \in S^{\mathcal{N}}_{A_{\delta B}} \leftrightarrow (i, \rho_B(N)) \in T^{\mathcal{N}}_{A_{\delta B}}$$

But as $N_1^\circ = N_2^\circ$, we draw the same conclusion about N_2 and N_4, giving a δ-allowable permutation ρ' such that $N_4^\circ = f_{\delta,\varepsilon}(\rho'(t))$; note that the inflexible path and tangle in question will be the same for both pairs. We also have

$$\forall j,\ \forall a,\ (j, a) \in \mathsf{supp}(t)^{\mathcal{A}}_B \to \forall i,\ (i, a) \in S^{\mathcal{A}}_{A_{\delta B}} \leftrightarrow (i, \rho'_B(a)) \in T^{\mathcal{A}}_{A_{\delta B}}$$
$$\forall j,\ \forall N,\ (j, N) \in \mathsf{supp}(t)^{\mathcal{N}}_B \to \forall i,\ (i, N) \in S^{\mathcal{N}}_{A_{\delta B}} \leftrightarrow (i, \rho'_B(N)) \in T^{\mathcal{N}}_{A_{\delta B}}$$

Combining these, we obtain

$$\forall j,\ \forall a,\ (j, a) \in \mathsf{supp}(t)^{\mathcal{A}}_B \to \forall i,\ (i, \rho_B(a)) \in T^{\mathcal{A}}_{A_{\delta B}} \leftrightarrow (i, \rho'_B(a)) \in T^{\mathcal{A}}_{A_{\delta B}}$$
$$\forall j,\ \forall N,\ (j, N) \in \mathsf{supp}(t)^{\mathcal{N}}_B \to \forall i,\ (i, \rho_B(N)) \in T^{\mathcal{N}}_{A_{\delta B}} \leftrightarrow (i, \rho'_B(N)) \in T^{\mathcal{N}}_{A_{\delta B}}$$

We claim that $\rho(\mathsf{supp}(t)) = \rho'(\mathsf{supp}(t))$. As T is strong, for each atom a such that $(j, a) \in \mathsf{supp}(t)^{\mathcal{A}}_B$, there is some k such that $(i, \rho_B(a)) \in T^{\mathcal{A}}_{A_{\delta B}}$. Thus $\rho_B(a) = \rho'_B(a)$. The same conclusion may be drawn for near-litters. Thus $\rho(\mathsf{supp}(t)) = \rho'(\mathsf{supp}(t))$, giving $\rho(t) = \rho'(t)$, and hence $N_3^\circ = N_4^\circ$. □

Proposition 5.18. Let S, T be strong supports such that $\mathsf{spec}(S) = \mathsf{spec}(T)$. Then for each $(N_1, N_3), (N_2, N_4) \in \mathsf{conv}^{\mathcal{N}}_{S_A,T_A}$,

$$\mathsf{interf}(N_1, N_2) \subseteq \mathsf{coim}\,\mathsf{conv}^{\mathcal{A}}_{S_A,T_A};\quad \mathsf{interf}(N_3, N_4) \subseteq \mathsf{im}\,\mathsf{conv}^{\mathcal{A}}_{S_A,T_A}$$

Proof. As S is strong, we have $\mathrm{interf}(N_1, N_2) \subseteq \operatorname{im} S_A^{\mathcal{A}}$. But $\operatorname{coim} \mathrm{conv}_{S_A,T_A}^{\mathcal{A}} = \operatorname{im} S_A^{\mathcal{A}}$, as required.[4] The result for $\mathrm{interf}(N_3, N_4)$ then follows by symmetry. □

Definition 5.19. Let S, T be strong β-supports such that $\mathrm{spec}(S) = \mathrm{spec}(T)$. Then for each A, we define the base action conv_{S_A,T_A} to be $(\mathrm{conv}_{S_A,T_A}^{\mathcal{A}}, \mathrm{conv}_{S_A,T_A}^{\mathcal{N}})$; this is a base action by propositions 5.15 to 5.18. We now define the β-action $\mathrm{conv}_{S,T}$ by $(\mathrm{conv}_{S,T})_A = \mathrm{conv}_{S_A,T_A}$.

Proposition 5.20. Let S, T be strong supports such that $\mathrm{spec}(S) = \mathrm{spec}(T)$. Then $\mathrm{conv}_{S,T}$ is coherent.

Proof. Suppose that $(N_1, N_2) \in \mathrm{conv}_{S_A,T_A}^{\mathcal{N}}$, so there is i such that $(i, N_1) \in S_A^{\mathcal{N}}$ and $(i, N_2) \in T_A^{\mathcal{N}}$.

Suppose that N_1° is A-flexible. By proposition 5.12, we immediately conclude that N_2° is A-flexible as required.

Now suppose that N_1° is A-inflexible with inflexible β-path $I = (\gamma, \delta, \varepsilon, B)$ and tangle $t : \mathsf{Tang}_\delta$. By proposition 5.12, there is some δ-allowable permutation ρ such that $N_2^\circ = f_{\delta,\varepsilon}(\rho(t))$ and

$$\forall j,\ \forall a,\ (j, a) \in \mathrm{supp}(t)_C^{\mathcal{A}} \to \forall i,\ (i, a) \in S_{B_{\delta C}}^{\mathcal{A}} \leftrightarrow (i, \rho_C(a)) \in T_{B_{\delta C}}^{\mathcal{A}}$$
$$\forall j,\ \forall N,\ (j, N) \in \mathrm{supp}(t)_C^{\mathcal{N}} \to \forall i,\ (i, N) \in S_{B_{\delta C}}^{\mathcal{N}} \leftrightarrow (i, \rho_C(N)) \in T_{B_{\delta C}}^{\mathcal{N}}$$

We must show that $((\mathrm{conv}_{S,T})_B)_\delta(\mathrm{supp}(t)) = \rho(\mathrm{supp}(t))$. We will show the result for atoms; the result for near-litters is identical. Let $(j, a) \in \mathrm{supp}(t)_C^{\mathcal{A}}$. Then as S is strong, there is k such that $(k, a) \in S_{B_{\delta C}}^{\mathcal{A}}$. Then by the equation above, $(k, \rho_C(a)) \in T_{B_{\delta C}}^{\mathcal{A}}$. Hence $(a, \rho_C(a)) \in (((\mathrm{conv}_{S,T})_B)_\delta)_C$ as required. □

Proposition 5.21. Let S, T be strong supports such that $\mathrm{spec}(S) = \mathrm{spec}(T)$. Then there is an allowable permutation ρ such that $\rho(S) = T$.

Proof. By proposition 5.20, we may apply theorem 4.38 to $\mathrm{conv}_{S,T}$ to obtain an allowable permutation ρ that $\mathrm{conv}_{S,T}$ approximates, which directly gives $\rho(S) = T$ as required. □

5.4 Recoding

Definition 5.22. Let $\gamma < \beta$ be proper type indices at most α. An object $x : \mathsf{TSet}_\beta$ is called a γ-*combination* of a set of β-coding functions s with respect to a β-support S if

$$U_\beta(x)(\gamma) = \bigcup_{(V,v)\in\chi\in s,\ V\geq S} U_\beta(v)(\gamma)$$

By extensionality, a set of coding functions s has at most one γ-combination with respect to a given support S.

Proposition 5.23. If x is a γ-combination of s with respect to S then $\rho(x)$ is a γ-combination of s with respect to $\rho(S)$.

[4] Make this a lemma.

Proof. We can calculate

$$\begin{aligned}
U_\beta(\rho(x))(\gamma) &= \rho_\gamma[U_\beta(x)(\gamma)] \\
&= \rho_\gamma\left[\bigcup_{(V,v)\in\chi\in s,\, V\geq S} U_\beta(v)(\gamma)\right] \\
&= \bigcup_{(V,v)\in\chi\in s,\, V\geq S} \rho_\gamma\left[U_\beta(v)(\gamma)\right] \\
&= \bigcup_{(V,v)\in\chi\in s,\, V\geq S} U_\beta(\rho(v))(\gamma) \\
&= \bigcup_{(V,v)\in\chi\in s,\, V\geq \rho(S)} U_\beta(v)(\gamma)
\end{aligned}$$

where the last inequality uses the fact that coding functions are defined on a support orbit. ☐

Definition 5.24. Let s be a set of β-coding functions, and let o be a β-support orbit such that for each $S \in o$, s has a γ-combination x with respect to S where S supports x. Then the (γ, β)-*raised coding function* for (s, o) is the relation $\chi : \mathsf{StrSupp}_\beta \to \mathsf{TSet}_\beta \to \mathsf{Prop}$ defined by the constructor

$$\forall S \in o,\ \forall x \text{ combinations of } (s, S),\ (S, x) \in \chi$$

Proposition 5.25. The (γ, β)-raised coding function for (s, o) is a coding function.

Proof. Coinjectivity follows from uniqueness of combinations. The nonemptiness and support orbit conditions follow from the definition, as does the condition that $(S, x) \in \chi$ implies that S is a support for x. It remains to show that if $(S, x) \in \chi$ and ρ is β-allowable, then $(\rho(S), \rho(x)) \in \chi$, and this follows directly from proposition 5.23. ☐

Definition 5.26 (designated support)**.** For a type index $\beta \leq \alpha$, a β-*set orbit* is the quotient of TSet_β under the relation of being in the same orbit under β-allowable permutations. We write $[x]$ for the set orbit of x. For each set orbit o, we choose a representative $\mathsf{repr}(o) : \mathsf{TSet}_\beta$ with $[\mathsf{repr}(o)] = o$, and define a support S_o for $\mathsf{repr}(o)$. For each set, we choose a β-allowable permutation twist_t with the property that $\mathsf{twist}_t(\mathsf{repr}([t])) = t$, and we define the *designated support* of t to be $\mathsf{dsupp}(t) = \mathsf{twist}_t(S_{[t]})$. This is a support for t.

Definition 5.27. Let $\gamma < \beta$ be proper type indices. Let S be a β-support and let $u : \mathsf{TSet}_\gamma$. Then the (γ, β)-*raised singleton* coding function is

$$\mathsf{raise}(S, u) = \chi_{(\mathsf{singleton}_\beta(u), S + \mathsf{dsupp}(u)^\beta)}$$

A coding function is called a (γ, β)-*raised singleton* if it is of the form $\mathsf{raise}(S, u)$.

Proposition 5.28. Let $\gamma < \beta$ be proper type indices, and let $x : \mathsf{TSet}_\beta$. Then for any support S of x, x is a combination of the set

$$\{\mathsf{raise}(S, u) \mid u \in U_\beta(x)(\gamma)\}$$

Proof. We must show that

$$U_\beta(x)(\gamma) = \bigcup_{u\in U_\beta(x)(\gamma),\, (V,v)\in \mathsf{raise}(S,u),\, V\geq S} U_\beta(v)(\gamma)$$

First, suppose $u \in U_\beta(x)(\gamma)$. Then we have

$$(S + \mathsf{dsupp}(u)^\beta, \mathsf{singleton}_\beta(u)) \in \mathsf{raise}(S, u); \quad u \in U_\beta(\mathsf{singleton}_\beta(u))(\gamma)$$

so the left-hand side is contained in the right-hand side.

For the converse, suppose that $u \in U_\beta(x)(\gamma)$ and $(V, v) \in \mathsf{raise}(S, u)$ with $V \geq S$. As $(V, v) \in \mathsf{raise}(S, u)$, there is some β-allowable ρ such that

$$\rho(S + \mathsf{dsupp}(u)^\beta) = V; \quad \rho(\mathsf{singleton}_\beta(u)) = v$$

As $V \geq S$, we obtain $\rho(S) = S$,[5] so $\rho(x) = x$. Then

$$\begin{aligned} U_\beta(v)(\gamma) &= U_\beta(\rho(\mathsf{singleton}_\beta(u)))(\gamma) \\ &= U_\beta(\mathsf{singleton}_\beta(\rho_\gamma(u)))(\gamma) \\ &= \{\rho_\gamma(u)\} \\ &\subseteq U_\beta(\rho(x))(\gamma) \\ &= U_\beta(x)(\gamma) \end{aligned}$$

as required. □

Proposition 5.29. Let $\gamma < \beta$ be proper type indices, and let χ be a β-coding function. Then there is a set of (γ, β)-raised singletons s and support orbit o such that χ is the (γ, β)-raised coding function for (s, o).

Proof. Let χ be a β-coding function, and let $(S, x) \in \chi$. Let

$$s = \{\mathsf{raise}(S, u) \mid u \in U_\beta(x)(\gamma)\}$$

Let o be the support orbit such that $T \in o$ if and only if $T \in \mathsf{coim}\,\chi$. We claim that χ is the raised coding function for (s, o). It suffices to show that (S, x) is in this raised coding function. That is, we must show that $S \in o$, which is trivial, and that x is a combination of s, which is the content of proposition 5.28. □

5.5 Coding the base type

Proposition 5.30 (the swap permutation)**.** Let S be a base support that is closed under interference of near-litters. Let a_1, a_2 be atoms not in $\mathsf{im}\, S^{\mathcal{A}}$ such that

$$\forall N \in \mathsf{im}\, S^{\mathcal{N}},\ a_1 \in N \leftrightarrow a_2 \in N$$

Then there is a base permutation π that fixes S and maps a_1 to a_2.

Proof. Let i be an index that does not occur in $\mathsf{coim}\, S^{\mathcal{A}}$, and define T_1, T_2 by

$$U_n^{\mathcal{A}} = S^{\mathcal{A}} \sqcup \{(i, a_n)\}; \quad U_n^{\mathcal{N}} = S^{\mathcal{N}}$$

for $n = 1, 2$. We claim that T_1 and T_2 are have the same specification, treated as $\bot$-supports. By appealing to proposition 5.12 and noting that every litter is flexible for the unique path $\bot \rightsquigarrow \bot$, it suffices to check:

[5] This is a good lemma.

- (atom condition) For all i, a_1, a_2, if $(i, a_1) \in T_1^{\mathcal{A}}$ and $(i, a_2) \in T_2^{\mathcal{A}}$, we have

$$\forall j,\ (j, a_1) \in T_1^{\mathcal{A}} \leftrightarrow (j, a_2) \in T_2^{\mathcal{A}}$$

  and

$$\forall j,\ (\exists N,\ (j, N) \in T_1^{\mathcal{N}} \wedge a_1 \in N) \leftrightarrow (\exists N,\ (j, N) \in T_2^{\mathcal{N}} \wedge a_2 \in N)$$

- (litter condition) For all i, N_1, N_2, if $(i, N_1) \in T_1^{\mathcal{N}}$ and $(i, N_2) \in T_2^{\mathcal{N}}$, we have

$$\forall j,\ (\exists N',\ (j, N') \in T_1^{\mathcal{N}} \wedge N_1^\circ = N'^\circ) \leftrightarrow (\exists N',\ (j, N') \in T_2^{\mathcal{N}} \wedge N_2^\circ = N'^\circ)$$

The atom condition follows directly from the two assumptions, and the litter condition is vacuously true as $T_1^{\mathcal{N}} = T_2^{\mathcal{N}}$.

Note also that T_1, T_2 are strong, treated as $\perp$-supports. Then, by proposition 5.21, there is a $\perp$-allowable permutation π such that $\pi(T_1) = T_2$. Thus, $\pi(S) = S$ and $\pi(a_1) = a_2$. □

Proposition 5.31. Let S be a base support that is closed under interference of near-litters. Suppose that S supports a set $s : \mathsf{Set}\,\mathcal{A}$ under the action of base permutations. Then for every pair of atoms a_1, a_2, the statements

$$a_1, a_2 \notin \operatorname{im} S^{\mathcal{A}}$$

and

$$\forall N \in \operatorname{im} S^{\mathcal{N}},\ a_1 \in N \leftrightarrow a_2 \in N$$

imply that $a_1 \in s \leftrightarrow a_2 \in s$.

Proof. Let a_1, a_2 be atoms that satisfy the two statements. By proposition 5.30, there is a base permutation π that fixes S and maps a_1 to a_2. As S supports s, we obtain $\pi(s) = s$. So $a_1 \in s$ if and only if $a_2 \in s$, as required. □

Proposition 5.32. Let S be a base support. Then S supports at most $2^{\#\kappa}$-many sets $s : \mathsf{Set}\,\mathcal{A}$ under the action of base permutations.

Proof. Without loss of generality, we may assume S is closed under interference of near-litters, since extensions will support any object that the original support supports.[6] The *information* of a set $s : \mathsf{Set}\,\mathcal{A}$ for a base support S is a triple $(i^{\mathcal{A}}, i^{\mathcal{N}}, p)$ where

- $i^{\mathcal{A}}$ is the set of indices i such that $(i, a) \in S^{\mathcal{A}}$ for some $a \in s$;
- $i^{\mathcal{N}}$ is the set of indices i such that $(i, N) \in S^{\mathcal{N}}$ for some near-litter N with $N \cap s \setminus \operatorname{im} S^{\mathcal{A}} \neq \varnothing$;
- p is the proposition that every atom $a \notin \operatorname{im} S^{\mathcal{A}} \cup \bigcup \operatorname{im} S^{\mathcal{N}}$ lies in s.

Suppose that s, t are sets of atoms that S supports. We claim that if s and t have the same information $(i^{\mathcal{A}}, i^{\mathcal{N}}, p)$, they are equal. By antisymmetry it suffices to show that $s \subseteq t$.

Let $a \in s$. Suppose that $(i, a) \in S^{\mathcal{A}}$ for some i. Then $s \in t$ as s, t have the same $i^{\mathcal{A}}$.

Now suppose that $a \notin \operatorname{im} S^{\mathcal{A}}$, but that there is some near-litter N with $a \in N$ and $(i, N) \in S^{\mathcal{N}}$. Then there is some atom $a' \in N \cap t \setminus \operatorname{im} S^{\mathcal{A}}$. It suffices by proposition 5.31 to show that

$$\forall N' \in \operatorname{im} S^{\mathcal{N}},\ a \in N' \leftrightarrow a' \in N'$$

[6] This closure operation should have been developed for proposition 5.4.

because then $a \in t$ if and only if $a' \in t$. Suppose that $a \in N'$ for some $N' \in \operatorname{im} S^{\mathcal{N}}$. If $N^\circ \neq N'^\circ$, then $a \in N \cap N' \setminus \operatorname{im} S^{\mathcal{A}}$ would contradict the assumption that S is closed under interference. So $N^\circ = N'^\circ$. If $a' \notin N'$, then $a \in (N \mathbin{\triangle} N') \setminus \operatorname{im} S^{\mathcal{A}}$ would also contradict the assumption that S is closed under interference. Hence $a' \in N'$ as required.

Finally, suppose that $a \notin \operatorname{im} S^{\mathcal{A}}$ and for all near-litters $N \in \operatorname{im} S^{\mathcal{N}}$, we have $a \notin N$. If $a \notin t$, then p is false, so there is an atom a' with the same properties that does not lie in s. Again, it suffices by proposition 5.31 to show that

$$\forall N' \in \operatorname{im} S^{\mathcal{N}},\ a \in N' \leftrightarrow a' \in N'$$

because then $a \in s$ if and only if $a' \in s$. But the left-hand side and the right-hand side are both false, giving the result.

This result shows that the map that sends a set s that S supports to its information is injective, and so as there are $2^{\#\kappa}$-many possible information tuples, there are at most $2^{\#\kappa}$-many sets that S supports under the action of base permutations. □

5.6 Counting

Proposition 5.33. Suppose that for all type indices $\delta < \beta$, there are strictly less than $\#\mu$-many δ-coding functions. Then there are less than $\#\mu$ β-specifications.

Proof. There are less than $\#\mu$ atom conditions as $\#\kappa < \#\mu$ and $\#\mu$ is a strong limit. There are less than $\#\mu$ inflexible β-paths, as each is determined by three type indices less than β and a path with maximum less than β, of which there are $\#\mu$-many. There are less than $\#\mu$-many coding functions of any type $\delta < \beta$, because König's theorem gives

$$\sum_{\delta<\beta} \#\{\delta\text{-coding functions}\} < \prod_{\delta<\beta} \#\mu = \#\mu^{\#\{\delta<\beta\}} = \#\mu$$

where the last equality follows from the facts that $\#\{\delta < \beta\}$ has cardinality less than $\operatorname{cof}(\operatorname{ord}(\#\mu))$ and that $\#\mu$ is a strong limit. There are less than $\#\mu$ δ-trees of relations on κ for each $\delta \leq \alpha$, because there are less than $\#\mu$-many relations on κ as $\#\mu$ is a strong limit, allowing us to conclude by one of the remarks in definition 2.11. Hence there are less than $\#\mu$ β-near-litter conditions. We can again apply the result about cardinalities of types of trees to deduce that there are less than $\#\mu$ β-trees of enumerations of atom conditions and of β-near-litter conditions, as required. □

Definition 5.34. A *weak β-specification* is a triple $W = (R^{\mathcal{A}}, R^{\mathcal{N}}, \sigma)$ where $R^{\mathcal{A}}, R^{\mathcal{N}}$ are β-trees of relations on κ, and σ is a β-specification. We say that a weak specification *specifies* a support S if there is a strong support T such that $\sigma = \operatorname{spec}(T)$, $S \preceq T$, and

$$(i, j) \in R^{\mathcal{A}}_A \leftrightarrow \exists a,\ (i, a) \in S^{\mathcal{A}}_A \wedge (j, a) \in T^{\mathcal{A}}_A$$
$$(i, j) \in R^{\mathcal{N}}_A \leftrightarrow \exists N,\ (i, N) \in S^{\mathcal{N}}_A \wedge (j, N) \in T^{\mathcal{N}}_A$$

Proposition 5.35. Every support has a weak specification that specifies it.

Proof. Let S be a support, and let T be a strong support such that $S \preceq T$, which exists by proposition 5.4. Then simply define $R^{\mathcal{A}}$ and $R^{\mathcal{N}}$ to be the required relations. □

Proposition 5.36. If W is a weak specification that specifies supports S and T, then there is an allowable permutation ρ such that $\rho(S) = T$.

Proof. Let $W = (R^{\mathcal{A}}, R^{\mathcal{N}}, \sigma)$, and let U, V be strong supports such that $\operatorname{spec}(U) = \sigma = \operatorname{spec}(V)$, $S \preceq U, T \preceq V$, and

$$(\exists a,\ (i, a) \in S_A^{\mathcal{A}} \wedge (j, a) \in U_A^{\mathcal{A}}) \leftrightarrow (\exists a,\ (i, a) \in T_A^{\mathcal{A}} \wedge (j, a) \in V_A^{\mathcal{A}})$$
$$(\exists N,\ (i, N) \in S_A^{\mathcal{N}} \wedge (j, N) \in U_A^{\mathcal{N}}) \leftrightarrow (\exists N,\ (i, N) \in T_A^{\mathcal{N}} \wedge (j, N) \in V_A^{\mathcal{N}})$$

By proposition 5.21, there is an allowable permutation ρ such that $\rho(U) = V$. We claim that $\rho(S) = T$. Suppose $(i, a) \in S_A^{\mathcal{A}}$. Then as $S \preceq U$, there is j such that $(j, a) \in U_A^{\mathcal{A}}$. So there is a' such that $(i, a') \in T_A^{\mathcal{A}}$ and $(j, a') \in V_A^{\mathcal{A}}$. Then $a' = \rho_A(a)$ as $\rho(U) = V$, as required. The same simple calculation gives the required result for near-litters. □

Proposition 5.37. Suppose that for all type indices $\delta < \beta$, there are strictly less than $\#\mu$-many δ-coding functions. Then there are less than $\#\mu$ weak β-specifications.

Proof. Follows directly from proposition 5.33 and the remark that there are less than $\#\mu$ β-trees of relations on κ. □

Proposition 5.38. Suppose that for all type indices $\delta < \beta$, there are strictly less than $\#\mu$-many δ-coding functions. Then there are less than $\#\mu$ β-support orbits.

Proof. Define a function from the type of β-support orbits into the type of weak β-specifications by mapping a representative to a weak specification that specifies it; one will always exist by proposition 5.35. This is an injection: if o_1, o_2 are orbits with representatives S_1, S_2 and S_1, S_2 have the same assigned weak specification, then by proposition 5.36 there is an allowable permutation ρ such that $\rho(S_1) = S_2$, and so $o_1 = o_2$. So we are done as there are less than $\#\mu$ weak β-specifications by proposition 5.37. □

Proposition 5.39. Let β be a type index (which in practice will be $\bot$ or the lowest proper type index). Suppose that there are less than $\#\mu$ β-support orbits. Let ν be a cardinal less than $\#\mu$ such that for each β-support S, there are at most ν-many objects $x : \mathsf{TSet}_\beta$ that S supports. Then there are less than $\#\mu$ β-coding functions.

Proof. Every β-coding function is of the form $\chi_{(x,S)}$ where S is a representative chosen in advance for a support orbit, and x is an object that S supports under the action of β-allowable permutations. So there are at most

$$\sum_{S \text{ representatives}} \nu = \#\{\text{support orbits}\} \cdot \nu$$

coding functions, which is less than $\#\mu$. □

Proposition 5.40. There are less than $\#\mu$ $\bot$-coding functions.

Proof. By proposition 5.39, it suffices to show that there are less than $\#\mu$ $\bot$-support orbits and that there is a bound less than $\#\mu$ on the amount of objects that a given $\bot$-support supports. The first result follows from proposition 5.38, which has vacuous assumptions in this case. The second result follows from applying proposition 5.32 to singletons of atoms, after applying the bijections between $\bot$-supports and base supports, and between $\bot$-allowable permutations and base permutations. □

Proposition 5.41. There are less than $\#\mu$ β-coding functions if β is the minimal inhabitant of λ.

Proof. Again, we apply proposition 5.39. The first claim follows from proposition 5.38, where this time we use the fact that there are less than $\#\mu$ $\bot$-coding functions (proposition 5.40). The second claim follows from proposition 5.32 to $\bot$-extensions of type β objects, noting that β-supports correspond to base supports and that β-allowable permutations correspond to base permutations.[7] The fact that β-allowable permutations correspond to base permutations relies on the fact that we can construct a β-permutation from a base permutation using coherent data (definition 2.25). □

Proposition 5.42. Suppose that for all type indices $\delta < \beta$, there are strictly less than $\#\mu$-many δ-coding functions. Then if $\gamma < \beta$ is a proper type index, there are strictly less than $\#\mu$ (γ, β)-raised singletons.

Proof. A (γ, β)-raised singleton $\mathsf{raise}(S, u) = \chi_{(\mathsf{singleton}_\beta(u), S + \mathsf{dsupp}(u)^\beta)}$ is determined by a triple (R, o, χ) where

- R is the β-tree given by $R_A = i$ when $S_A = (i, f)$;
- o is the support orbit of $S + \mathsf{dsupp}(u)^\beta$; and
- χ is the coding function $\chi_{(u, \mathsf{dsupp}(u))}$.

Indeed, suppose $\mathsf{raise}(S, u)$ and $\mathsf{raise}(T, v)$ have the same triple (R, o, χ). We must show that

$$\chi_{(\mathsf{singleton}_\beta(u), S + \mathsf{dsupp}(u)^\beta)} = \chi_{(\mathsf{singleton}_\beta(v), S + \mathsf{dsupp}(v)^\beta)}$$

Then there is a β-allowable ρ such that $\rho(S + \mathsf{dsupp}(u)^\beta) = T + \mathsf{dsupp}(v)^\beta$, and as they have the same tree R, we can decompose this into $\rho(S) = T$ and $\rho(\mathsf{dsupp}(u)^\beta) = \mathsf{dsupp}(v)^\beta$. In particular, $\rho_\beta(\mathsf{dsupp}(u)) = \mathsf{dsupp}(v)$. As $\chi_{(u, \mathsf{dsupp}(u))} = \chi_{(v, \mathsf{dsupp}(v))}$, there is a γ-allowable permutation ρ' such that $\rho'(u) = v$ and $\rho'(\mathsf{dsupp}(u)) = \mathsf{dsupp}(v)$. Hence $\rho_\beta(u) = v$. This gives

$$\rho(\mathsf{singleton}_\beta(u)) = \mathsf{singleton}_\beta(\rho_\beta(u)) = \mathsf{singleton}_\beta(v)$$

as required.

Now, it remains to show that there are less than $\#\mu$ such triples (R, o, χ). But this follows directly from proposition 5.38 and the assumption on the cardinalities of the types of coding functions. □

Proposition 5.43. There are less than $\#\mu$-many β-coding functions for all type indices $\beta \le \alpha$.

Proof. By induction we may assume that for all type indices $\delta < \beta$, there are strictly less than $\#\mu$-many δ-coding functions. By propositions 5.40 and 5.41, we may assume that β is a proper type index, and that it is not minimal in λ, so there is some proper type index $\gamma < \beta$. By proposition 5.29, every β-coding function is determined by a set of (γ, β)-raised singletons s and support orbit o. The conclusion then follows from propositions 5.38 and 5.42 and the fact that $\#\mu$ is a strong limit. □

Proposition 5.44. For each type index $\beta \le \alpha$, $\#\mathsf{TSet}_\beta = \#\mu$.

[7] One useful claim to prove for this is that there is a unique path $\beta \rightsquigarrow \bot$, and so type β objects satisfy $\bot$-extensionality by injectivity of U_β.

Proof. If β is $\bot$, we already know $\#\mathsf{TSet}_\bot = \#\mathcal{A} = \#\mu$, so suppose $\#\beta$ is a proper type index. Each object $x : \mathsf{TSet}_\beta$ is determined by a β-coding function and a β-support. Since there are less than $\#\mu$-many β-coding functions (proposition 5.43) and there are exactly $\#\mu$ β-supports, we obtain $\#\mathsf{TSet}_\beta \leq \#\mu$. Since the typed near-litter map $\mathcal{N} \to \mathsf{TSet}_\beta$ is injective, there are at least $\#\mathcal{N} = \#\mu$ inhabitants of TSet_β, giving the result by antisymmetry. □

Proposition 5.45. For each type index $\beta \leq \alpha$, $\#\mathsf{Tang}_\beta = \#\mu$.

Proof. Use proposition 5.44 and the fact that there are precisely $\#\mu$-many β-supports. □

Chapter 6

Wrapping up the main induction

6.1 Induction, in abstract

In this section, we prove a theorem on inductive constructions using a proof-irrelevant Prop.

Definition 6.1. Let $I : \mathsf{Type}_u$ be a type with a well-founded transitive relation $\prec$. Let $A : I \to \mathsf{Type}_v$ be a family of types indexed by I, and let

$$B : \prod_{i:I} A_i \to \left(\prod_{j:I} j \prec i \to A_j \right) \to \mathsf{Sort}_w$$

An *inductive construction* for (I, A, B) is a pair of functions

$$\begin{aligned} F_A &: \prod_{i:I} \prod_{d:\prod_{j:I} j \prec i \to A_j} \\ &\left(\prod_{j:I} \prod_{h:j \prec i} B\ j\ (d\ j\ h)\ (k\ h' \mapsto d\ k\ (\mathsf{trans}(h', h))) \right) \to A_i \\ F_B &: \prod_{i:I} \prod_{d:\prod_{j:I} j \prec i \to A_j} \\ &\prod_{h:\left(\prod_{j:I} \prod_{h:j \prec i} B\ j\ (d\ j\ h)\ (k\ h' \mapsto d\ k\ (\mathsf{trans}(h', h)))\right)} B\ i\ (F_A\ i\ d\ h)\ d \end{aligned}$$

Proposition 6.2 (inductive construction theorem for propositions)**.** Let (F_A, F_B) be an inductive construction for (I, A, B), where w is 0. Then there are computable functions

$$G : \prod_{i:I} A_i; \quad H : \prod_{i:I} B\ i\ G_i\ (j\ \_ \mapsto G_j)$$

such that for each $i : I$,

$$G_i = F_A\ i\ (j\ \_ \mapsto G_j)\ (F_B\ i\ (j\ \_ \mapsto H_j))$$

Proof. Recall that $\mathsf{Part}\,\alpha$ denotes the type $\sum_{p:\mathsf{Prop}}(p \to \alpha)$. For $i : I$, we define the *hypothesis* on data $t : \prod_{j:I} j \prec i \to \mathsf{Part}\, A_j$ to be the proposition

$$\mathsf{IH}(i,t) = \sum_{D:\prod_{j:I}\prod_{h:j\prec i}\mathsf{pr}_1(t\ j\ h)} \prod_{j:I}\prod_{h:j\prec i} B\ j\ (\mathsf{pr}_2(t\ j\ h)\ (D\ j\ h))$$
$$(k\ h' \mapsto (\mathsf{pr}_2(t\ k\ (\mathsf{trans}(h',h)))\ (D\ k\ (\mathsf{trans}(h',h)))))$$

Now we define $K : \prod_{i:I} \mathsf{Part}\, A_i$ by

$$K = \mathsf{fix}_{\mathsf{Part}\,A_{(-)}}\left(it \mapsto \left\langle \mathsf{IH}(i,t), h \mapsto F_A\ i\ (j\ h' \mapsto (\mathsf{pr}_2(t\ j\ h')\ (\mathsf{pr}_1(h)\ j\ h')))\ \mathsf{pr}_2(h)\right\rangle\right)$$

where fix_C is the fixed point function for $\prec$ and induction motive $C : I \to \mathsf{Type}_v$, with type

$$\mathsf{fix}_C : \left(\prod_{i:I}\left(\prod_{j:I} j \prec i \to C_j\right) \to C_i\right) \to \prod_{i:I} C_i$$

By definition of fix, we obtain the equation

$$K_i = \left\langle \mathsf{IH}(i, j\_ \mapsto K_j), h \mapsto F_A\ i\ (j\ h' \mapsto (\mathsf{pr}_2(K_j)\ (\mathsf{pr}_1(h)\ j\ h')))\ \mathsf{pr}_2(h)\right\rangle$$

Further, if $h_1 : \mathsf{pr}_1(K_i)$, we have the equation

$$\mathsf{pr}_2(K_i)\ h_1 = F_A\ i\ (j\ h' \mapsto (\mathsf{pr}_2(K_j)\ (\mathsf{pr}_1(h_2)\ j\ h')))\ \mathsf{pr}_2(h_2)$$

where $h_2 : \mathsf{IH}(i, j\_ \mapsto K_j)$ is obtained by casting from h_1 using the previous equation; this equation is derived from the extensionality principle of Part, which states that

$$\prod_{x,y:\mathsf{Part}\,\alpha}\left(\prod_{a:\alpha}(\exists h,\ a = \mathsf{pr}_2(x)\ h) \leftrightarrow (\exists h,\ a = \mathsf{pr}_2(y)\ h)\right) x = y$$

Using these two equations, we may now show directly by induction on i that $D' : \prod_{i:I} \mathsf{pr}_1(K_i)$.[1] From this, we may define $G : \prod_{i:I} A_i$ by $G_i = \mathsf{pr}_2(K_i)\, D'_i$. We may then easily obtain $H : \prod_{i:I} B\, i\, G_i\, (j\_ \mapsto G_j)$ by appealing to F_B and the two equations above. The required equality also easily follows from the two given equations. □

Theorem 6.3 (inductive construction theorem)**.** Let (F_A, F_B) be an inductive construction for (I, A, B). Then there are *noncomputable* functions

$$G : \prod_{i:I} A_i; \quad H : \prod_{i:I} B\, i\, G_i\, (j\,\_ \mapsto G_j)$$

such that for each $i : I$,

$$G_i = F_A\ i\ (j\,\_ \mapsto G_j)\ (F_B\ i\ (j\,\_ \mapsto H_j))$$

Proof. Define

$$C : \prod_{i:I} A_i \to \left(\prod_{j:I} j \prec i \to A_j\right) \to \mathsf{Prop}$$

by $C\ i\ x\ d = \mathsf{Nonempty}(B\ i\ x\ d)$. We then define the inductive construction (F'_A, F'_B) for (I, A, C) by

$$F'_A\ i\ d\ h = F_A\ i\ d\ (jh' \mapsto \mathsf{some}(h\ j\ h')); \quad F'_B\ i\ d\ h = \langle F_B\ i\ d\ (jh' \mapsto \mathsf{some}(h\ j\ h'))\rangle$$

where $\langle - \rangle$ is the constructor and some is the noncomputable destructor of Nonempty. The result is then direct from proposition 6.2. □

[1] TODO: More details?

6.2 Building the tower

Definition 6.4. For a proper type index α, the *main motive* at α is the type Motive_α consisting of model data at α, a position function for Tang_α, and typed near-litters at α, such that if (x, S) is an α-tangle and y is an atom or near-litter that occurs in the range of S_A, then $\iota(y) < \iota(x, S)$.

Definition 6.5. We define the *main hypothesis*

$$\mathsf{Hypothesis} : \prod_{\alpha : \lambda} \mathsf{Motive}_\alpha \to \left(\prod_{\beta < \alpha} \mathsf{Motive}_\beta \right) \to \mathsf{Type}_{u+1}$$

at α, given Motive_α and $\prod_{\beta<\alpha} \mathsf{Motive}_\beta$, to be the type consisting of the following data.

- For $\gamma < \beta \leq \alpha$, there is a map $\mathsf{AllPerm}_\beta \to \mathsf{AllPerm}_\gamma$ that commutes with the coercions from $\mathsf{AllPerm}_{(-)}$ to $\mathsf{StrPerm}_{(-)}$.
- Let $\beta, \gamma < \alpha$ be distinct with γ proper. Let $t : \mathsf{Tang}_\beta$ and $\rho : \mathsf{AllPerm}_\alpha$. Then
$$(\rho_\gamma)_\perp(f_{\beta,\gamma}(t)) = f_{\beta,\gamma}(\rho_\beta(t))$$
- Suppose that $(\rho(\beta))_{\beta<\alpha}$ is a collection of allowable permutations such that whenever $\beta, \gamma < \alpha$ are distinct, γ is proper, and $t : \mathsf{Tang}_\gamma$, we have
$$(\rho(\gamma))_\perp(f_{\beta,\gamma}(t)) = f_{\beta,\gamma}(\rho(\beta)(t))$$
Then there is an α-allowable permutation ρ with $\rho_\beta = \rho(\beta)$ for each $\beta < \alpha$.
- For any $\beta < \alpha$,
$$U_\alpha(x)(\beta) \subseteq \mathsf{ran}\, U_\beta$$
- (extensionality) If $\beta : \lambda$ is such that $\beta < \alpha$, the map $U_\alpha(-)(\beta) : \mathsf{TSet}_\beta \to \mathsf{Set}\, \mathsf{StrSet}_\beta$ is injective.
- If $\beta : \lambda$ is such that $\beta < \alpha$, for every $x : \mathsf{TSet}_\beta$ there is some $y : \mathsf{TSet}_\alpha$ such that $U_\alpha(y)(\beta) = \{x\}$, and we write $\mathsf{singleton}_\alpha(x)$ for this object y.

Definition 6.6. The *inductive step for the main motive* is the function

$$\mathsf{Step}_M : \prod_{\alpha : \lambda} \prod_{M : \prod_{\beta<\alpha} \mathsf{Motive}_\beta} \left(\prod_{\beta : \lambda} \prod_{h : \beta < \alpha} \mathsf{Hypothesis}\ \beta\ (M\ \beta\ h)\ (\gamma\ h' \mapsto M\ \gamma\ (\mathsf{trans}(h', h))) \right) \to \mathsf{Motive}_\alpha$$

given as follows. Given the hypotheses α, M, H, we use definition 3.12 to construct model data at level α. Then, combine this with H to create an instance of coherent data below level α.[2] By proposition 5.45, we conclude that $\#\mathsf{Tang}_\alpha = \#\mu$, and so by proposition 3.15 there is a position function on Tang_α that respects the typed near-litters defined in definition 3.13. These data comprise the main motive at level α.

[2] This requires using the definition of new allowable permutations. There may be some difficulties here in converting between the types of model data at level α together with model data at all $\beta < \alpha$, and model data at all levels $\beta \leq \alpha$.

Definition 6.7. The *inductive step for the main hypothesis* is the function

$$\begin{aligned}\mathsf{Step}_H \,:\, &\prod_{\alpha : \lambda} \prod_{M : \prod_{\beta < \alpha} \mathsf{Motive}_\beta} \\ &\prod_{H : \prod_{\beta : \lambda} \prod_{h : \beta < \alpha} \mathsf{Hypothesis}\ \beta\ (M\ \beta\ h)\ (\gamma\ h' \mapsto M\ \gamma\ (\mathsf{trans}(h', h)))} \\ &\mathsf{Hypothesis}\ \alpha\ (\mathsf{Step}_M\ \ \alpha\, M\, H)\, M\end{aligned}$$

given as follows. Given the hypotheses α, M, H, we use the definitions from definition 6.6 to obtain model data at level α and coherent data below level α. The remaining proof obligations are handled by definition 3.14 and proposition 3.15.

Theorem 6.8 (model construction)**.** There are noncomputable functions

$$\mathsf{ComputeMotive} \,:\, \prod_{\alpha : \lambda} \mathsf{Motive}_\alpha$$

and

$$\mathsf{ComputeHypothesis} \,:\, \prod_{\alpha : \lambda} \mathsf{Hypothesis}\ \alpha\ \ \mathsf{ComputeMotive}_\alpha\ (\beta\ \_ \mapsto \mathsf{ComputeMotive}_\beta)$$

such that for each $\alpha : \lambda$,

$$\begin{aligned}\mathsf{ComputeMotive}_\alpha = \mathsf{Step}_M\ \ &\alpha\,(\beta\ \_ \mapsto \mathsf{ComputeMotive}_\beta) \\ &(\mathsf{Step}_H\ \ \alpha\,(\beta\ \_ \mapsto \mathsf{ComputeHypothesis}_\beta))\end{aligned}$$

Proof. Direct from theorem 6.3. □

We can then reconstruct, for each level α, the type of coherent data below α.

Chapter 7

Verifying Con(TTT)

7.1 Raising strong supports

In this section, let $\gamma < \beta$ be proper type indices strictly below the current level α.

Proposition 7.1. Let T be a γ-support, and let U be the strong support generated by T^β. If A is a path such that $\mathsf{im}\, U_A^{\mathcal{N}}$ is nonempty, then A has length at least 2.

Proof. Let $N \in \mathsf{im}\, U_A^{\mathcal{N}}$. Either A is of the form B^β for $B : \gamma \rightsquigarrow \bot$ and $N \in \mathsf{im}\, T_B^{\mathcal{N}}$, in which case B clearly has length at least 2 as γ is a proper type index, or $N \in \mathsf{im}\, V_B^{\mathcal{N}}$ where $A = B^\beta$ and $B : \delta \rightsquigarrow \bot$, and V supports some δ-set. But we cannot have $\delta = \bot$, because $\bot$-supports that support some δ-set cannot contain near-litters, so in this other case A also has length at least 2. □

Proposition 7.2. Let T be a γ-support, and let U be the strong support generated by T^β. Then U^α is strong.

Proof. The interference condition is clear.

First we show that proofs that A^α-inflexibility for litters that appear in U^α correspond to proofs of A-inflexibility. Clearly if L is A^α-flexible then L is A-flexible. Instead, let $I = (\delta, \varepsilon, \zeta, A)$ be an inflexible α-path and $t : \mathsf{Tang}_\varepsilon$ such that there is a near-litter $N \in \mathsf{im}(U^\alpha)^{\mathcal{N}}_{A_{\zeta_\bot}}$ with $N^\circ = f_{\varepsilon,\zeta}(t)$. Suppose that A is the empty path, so $\delta = \alpha$. Then $(U^\alpha)^{\mathcal{N}}_{A_{\zeta_\bot}} = (U^\alpha)^{\mathcal{N}}_{\zeta_\bot}$ is nonempty, so $\zeta = \beta$. This shows that $\mathsf{im}(U^\alpha)^{\mathcal{N}}_{\beta_\bot} = \mathsf{im}\, U^{\mathcal{N}}_\bot$ is nonempty, contradicting proposition 7.1. So A is nonempty, and is of the form B^α for $B : \beta \rightsquigarrow \delta$. So $(\delta, \varepsilon, \zeta, B)$ is an inflexible β-path for N°.

The expected conclusion then follows directly from the fact that U is strong. □

Proposition 7.3. Let S be a strong α-support and let T be a γ-support. Let U be the strong support generated by T^β, and let V be the support whose image is precisely those atoms in the interference of S_β and U. Let ρ be a β-allowable permutation that fixes S_β. Then $S + (\rho(U + V))^\alpha$ is strong.

Proof. Follows directly from propositions 5.3 and 7.2. □

Proposition 7.4. Let S be a strong α-support. Let U be a strong β-support with the property that if A is a path such that $\operatorname{im} U_A^{\mathcal{N}}$ is nonempty, then A has length at least 2. Then for every β-allowable ρ that fixes S_β, we have $\operatorname{spec}(S + U^\alpha) = \operatorname{spec}(S + (\rho(U))^\alpha)$.

Proof. Appeal to proposition 5.12. First, note that $(i, a) \in (S + (\rho(U))^\alpha)_A^{\mathcal{A}}$ if and only if $(i, a) \in S$ or $A = B^\alpha$ and $(i, a) \in \rho(U)_B^{\mathcal{A}}$. Thus,

$$\begin{aligned}(i, a_1) \in (S + U^\alpha)_A^{\mathcal{A}} \to \exists a_2,\ &(i, a_2) \in (S + (\rho(U))^\alpha)_A^{\mathcal{A}} \\ &\wedge [(\exists B,\ A = B^\alpha \wedge \rho_B(a_1) = a_2) \vee ((\forall B,\ A \neq B^\alpha) \wedge a_1 = a_2)]\end{aligned}$$

and the same holds for the other direction[1] and for near-litters.

The coimage condition is clear. Suppose that $(i, a_1) \in (S + U^\alpha)_A^{\mathcal{A}}$ and $(i, a_2) \in (S + (\rho(U))^\alpha)_A^{\mathcal{A}}$. Then by coinjectivity we can apply the above result, giving that either there is B such that $A = B^\alpha$ and $\rho_B(a_1) = a_2$, or there is no such B, and $a_1 = a_2$. In either case, the result is easy to show.

The result for litters follows from the fact that the condition on U implies that proofs of inflexiblity of litters in U correspond bijectively to proofs of inflexibility of litters in U^α.[2] □

Proposition 7.5. Let S be a strong α-support and let T be a γ-support. Let ρ be a β-allowable permutation that fixes S_β. Then there is an α-allowable permutation ρ' such that

$$\rho'(S) = S; \quad (\rho'_\beta)_\gamma(T) = \rho_\gamma(T)$$

Proof. Let U be the strong support generated by T^β, and let V be the support whose image is precisely those atoms in the interference of S_β and U. Then apply propositions 5.21, 7.1, 7.3 and 7.4. □

7.2 Tangled type theory

Definition 7.6 (tangled membership). We define the membership relation $\in_\beta^\alpha \colon \mathsf{TSet}_\beta \to \mathsf{TSet}_\alpha \to \mathsf{Prop}$ by

$$x \in_\beta^\alpha y \leftrightarrow U_\beta(x) \in U_\alpha(y)(\beta)$$

Extensionality holds at all proper type indices by proposition 3.8. Also, for every α-allowable ρ, we have

$$\rho_\beta(x) \in_\beta^\alpha \rho(y) \leftrightarrow x \in_\beta^\alpha y$$

Definition 7.7 (symmetric). Let $\beta < \alpha$ be proper type indices. A set $s : \mathsf{TSet}_\beta$ is called *α-symmetric* if there is some α-support S such that if ρ is an α-allowable permutation that fixes S, then ρ fixes s setwise. Note that it suffices to prove that for all ρ that fix S, $s \subseteq \rho[s]$ (or alternatively, $\rho[s] \subseteq s$). Every small set is symmetric, since a support can be obtained by raising the types of chosen supports for elements of the set and then collating the results.

Proposition 7.8. Let $\beta < \alpha$ be proper type indices. Let $s : \mathsf{TSet}_\beta$ be α-symmetric. Then there is $x : \mathsf{TSet}_\alpha$ such that

$$\forall y : \mathsf{TSet}_\beta,\ y \in_\beta^\alpha x \leftrightarrow y \in s$$

Moreover, every $x : \mathsf{TSet}_\alpha$ arises in this way; this is an induction principle for TSet_α.

[1]We can prove the other direction easily from this by substituting $\rho(U)$ and ρ^{-1}.
[2]TODO: Probably want more details when we get here.

Proof. If s is empty, the result follows directly from the definition of TSet_α. Otherwise, consider the code (β, s). By definition 3.7, there is exactly one code d such that $c \nrightarrow (\beta, s)$. Then c is a new t-set at level α, so is naturally an inhabitant of TSet_α. Using the membership relation from proposition 3.8, the type-β members of c are precisely s, as required.

For the induction principle, we apply the above construction to the set

$$s = \{y \mid y \in^\alpha_\beta x\}$$

and use extensionality to deduce that the object constructed is exactly x. □

Proposition 7.9 (unions of singletons)**.** Let $\gamma < \beta < \alpha$ be proper type indices. Let $s : \mathsf{Set}\,\mathsf{TSet}_\gamma$ be such that $\mathsf{singleton}_\beta[s]$ is α-symmetric. Then s is β-symmetric.

Proof. Let S be an α-support for $\mathsf{singleton}_\beta[s]$, which without loss of generality is strong. We claim that the α-symmetry of s is witnessed by S_β. Let ρ be a β-allowable permutation such that $\rho(S_\beta) = S_\beta$. Let $x \in s$; it suffices by substituting ρ^{-1} to prove that $\rho_\gamma(x) \in s$. As $x : \mathsf{TSet}_\gamma$, there is a γ-support T for x. By proposition 7.5, there is an α-allowable permutation ρ' such that

$$\rho'(S) = S; \quad (\rho'_\beta)_\gamma(T) = \rho_\gamma(T)$$

Thus,

$$\rho'_\beta(\mathsf{singleton}_\beta[s]) = \mathsf{singleton}_\beta[s]; \quad (\rho'_\beta)_\gamma(x) = \rho_\gamma(x)$$

We thus have

$$\begin{gathered} x \in s \\ \mathsf{singleton}_\beta(x) \in \mathsf{singleton}_\beta[s] \\ \rho'_\beta(\mathsf{singleton}_\beta(x)) \in \rho'_\beta(\mathsf{singleton}_\beta[s]) \\ \rho'_\beta(\mathsf{singleton}_\beta(x)) \in \mathsf{singleton}_\beta[s] \\ \mathsf{singleton}_\beta((\rho'_\beta)_\gamma(x)) \in \mathsf{singleton}_\beta[s] \\ \mathsf{singleton}_\beta(\rho_\gamma(x)) \in \mathsf{singleton}_\beta[s] \\ \rho_\gamma(x) \in s \end{gathered}$$

□

Theorem 7.10 (consistency of tangled type theory)**.** Let $\{x\}_\beta$ be an abbreviation for $\mathsf{singleton}_\beta(x)$, and let $\langle x, y\rangle_{\gamma,\beta}$ be an abbreviation for $\{\{x\}_\gamma, \{x, y\}_\gamma\}_\beta$. Then, the following axioms hold for our model TSet at all sequences of proper type indices $\zeta < \varepsilon < \delta < \gamma < \beta < \alpha$.

–	extensionality	$\forall x^\alpha, \forall y^\alpha, (\forall z^\beta, z \in x \leftrightarrow z \in y) \to x = y$
P1(a)	intersection	$\forall x^\alpha y^\alpha, \exists z^\alpha, \forall w^\beta, w \in z \leftrightarrow (w \in x \wedge w \in y)$
P1(b)	complement	$\forall x^\alpha, \exists z^\alpha, \forall w^\beta, w \in z \leftrightarrow w \notin x$
–	singleton	$\forall x^\beta, \exists y^\alpha, \forall z^\beta, z \in y \leftrightarrow z = x$
P2	singleton image	$\forall x^\beta, \exists y^\alpha, \forall z^\varepsilon w^\varepsilon, \langle \{z\}_\delta, \{w\}_\delta \rangle_{\gamma,\beta} \in y \leftrightarrow \langle z, w \rangle_{\delta,\gamma} \in x$
P3	insertion two	$\forall x^\gamma, \exists y^\alpha, \forall z^\zeta w^\zeta t^\zeta, \langle \{\{z\}_\varepsilon\}_\delta, \langle w, t \rangle_{\varepsilon,\delta} \rangle_{\gamma,\beta} \in y \leftrightarrow \langle z, t \rangle_{\varepsilon,\delta} \in x$
P4	insertion three	$\forall x^\gamma, \exists y^\alpha, \forall z^\zeta w^\zeta t^\zeta, \langle \{\{z\}_\varepsilon\}_\delta, \langle w, t \rangle_{\varepsilon,\delta} \rangle_{\gamma,\beta} \in y \leftrightarrow \langle z, w \rangle_{\varepsilon,\delta} \in x$
P5	cross product	$\forall x^\gamma, \exists y^\alpha, \forall z^\beta, z \in y \leftrightarrow \exists w^\delta t^\delta, z = \langle w, t \rangle_{\gamma,\beta} \wedge t \in x$
P6	type lowering	$\forall x^\alpha, \exists y^\delta, \forall z^\varepsilon, z \in y \leftrightarrow \forall w^\delta, \langle w, \{z\}_\delta \rangle_{\gamma,\beta} \in x$
P7	converse	$\forall x^\alpha, \exists y^\alpha, \forall z^\delta w^\delta, \langle z, w \rangle_{\gamma,\beta} \in y \leftrightarrow \langle w, z \rangle_{\gamma,\beta} \in x$
P8	cardinal one	$\exists x^\alpha, \forall y^\beta, y \in x \leftrightarrow \exists z^\gamma, \forall w, w \in y \leftrightarrow w = z$
P9	subset	$\exists x^\alpha, \forall y^\delta z^\delta, \langle y, z \rangle_{\gamma,\beta} \in x \leftrightarrow \forall w^\varepsilon, w \in y \to w \in z$

Proof. The axiom of extensionality was proven in proposition 3.8. All axioms except for the type lowering axiom can be easily proven using proposition 7.8. For type lowering, consider the set

$$y' = \{\{\{\{z^\varepsilon\}_\delta\}_\gamma\}_\beta \mid \forall w^\delta, \langle w, \{z\}_\delta \rangle_{\gamma,\beta} \in x\}$$

which exists by proposition 7.8, and then apply proposition 7.9 three times. $\square$

Chapter 8

Model theory and verifying $\mathsf{Con}(\mathsf{NF})$

In this chapter, we establish the model theory that is used to derive the consistency of NF from that of TTT. This requires much of mathlib's model theory library to be completely rewritten for the many-sorted case, which is (so far) not an objective of this project. Some potential design decisions are considered in the following sections. On the whole, this chapter should be considered just a sketch of the main argument: sufficient for a human reader, but insufficient for a detailed blueprint.

8.1 Many-sorted model theory

This is loosely based off the perspective on categorical logic offered by Johnstone in Volume 2 of *Sketches of an Elephant*, and takes heavy inspiration from the *Flypitch project*.

Definition 8.1. A Σ-*language* consists of a map $\mathsf{Functions} : \prod_{n:\mathbb{N}}(\mathsf{Fin}\,n \to \Sigma) \to \Sigma \to \mathsf{Type}_u$ and a map $\mathsf{Relations} : \prod_{n:\mathbb{N}}(\mathsf{Fin}\,n \to \Sigma) \to \mathsf{Type}_v$.

Definition 8.2. Let $\Phi : \Sigma \to \Sigma'$. Let L be a Σ-language and let L' be a Σ'-language. Then a Φ-*morphism* of languages $L \xrightarrow{\Phi} L'$ consists of a map

$$\mathsf{onFunction} : \prod_{n:\mathbb{N}} \prod_{A:\mathsf{Fin}\,n\to\Sigma} \prod_{B:\Sigma} \mathsf{Functions}_L(n, A, B) \to \mathsf{Functions}_{L'}(\Phi \circ A, \Phi(B))$$

and a map

$$\mathsf{onRelation} : \prod_{n:\mathbb{N}} \prod_{A:\mathsf{Fin}\,n\to\Sigma} \mathsf{Relations}_L(n, A) \to \mathsf{Functions}_{L'}(\Phi \circ A)$$

If L, L' are Σ-languages, we may simply say that a *morphism* of languages $L \to L'$ is a id_Σ-morphism $L \xrightarrow{\mathsf{id}_\Sigma} L'$.[1]

Definition 8.3. Let L, L' be Σ-languages. We define $L \oplus L'$ to be the Σ-language with

$$\mathsf{Functions}(n, A, B) = \mathsf{Functions}_L(n, A, B) \oplus \mathsf{Functions}_{L'}(n, A, B)$$

and

$$\mathsf{Relations}(n, A) = \mathsf{Relations}_L(n, A) \oplus \mathsf{Relations}_{L'}(n, A)$$

There are morphisms of languages $L, L' \to L \oplus L'$.

[1] It is crucial that $\mathsf{id} \circ f \equiv f \circ \mathsf{id} \equiv f$ definitionally.

Definition 8.4. Let L be a Σ-language, and let $M : \Sigma \to \mathsf{Type}_w$. An *$L$-structure* on M consists of a map

$$\mathsf{funMap} : \prod_{n:\mathbb{N}} \prod_{A:\mathsf{Fin}\,n\to\Sigma} \prod_{B:\Sigma} \mathsf{Functions}(n,A,B) \to \left(\prod_{i:\mathsf{Fin}\,n} M(A(i))\right) \to M(B)$$

and a map

$$\mathsf{relMap} : \prod_{n:\mathbb{N}} \prod_{A:\mathsf{Fin}\,n\to\Sigma} \mathsf{Relations}(n,A) \to \left(\prod_{i:\mathsf{Fin}\,n} M(A(i))\right) \to \mathsf{Prop}$$

We will write f^M for $\mathsf{funMap}(n,A,B,f)$, and similarly R^M for $\mathsf{relMap}(n,R)$. If M is an L-structure and an L'-structure, it is also naturally an $(L \oplus L')$-structure. If M is an L-structure and an L'-structure, then a morphism of Σ-languages $\Phi : L \to L'$ is called an *expansion* on M if it commutes with the interpretations of all symbols on M.

Definition 8.5. Let L be a Σ-language. A *morphism* of L-structures $M \to N$ consists of functions $h_A : M_A \to N_A$ such that for all $n : \mathbb{N}$, $A : \mathsf{Fin}\,n \to \Sigma$, and $B : \Sigma$, all function symbols $f : \mathsf{Functions}(n,A,B)$, and all $x : \prod_{i:\mathsf{Fin}\,n} M(A(i))$,

$$h_B(f^M(x)) = f^N(i \mapsto h_{A(i)}(x(i)))$$

and for all relation symbols $R : \mathsf{Relations}(n,A)$,

$$R^M(x) \to R^N(i \mapsto h_{A(i)}(x(i)))$$

Definition 8.6. Let L be a Σ-language, and let $\alpha : \mathsf{Type}_{u'}$ be a sort of variables of sort $S : \alpha \to \Sigma$. An *L-term on $\alpha : S$ of sort A*, the type of which is denoted $\mathsf{Term}_{\alpha:S}\,A$, is either

- a variable, comprised solely of a name $n : \alpha$ such that $S(n) = A$, or
- an application of a function symbol, comprised of some $n : \mathbb{N}$, a map $B : \mathsf{Fin}\,n \to \Sigma$, a function symbol $f : \mathsf{Functions}(n,B,A)$, and terms $t : \prod_{i:\mathsf{Fin}\,n} \mathsf{Term}_{\alpha:S}\,B(i)$.

Definition 8.7. Let L be a Σ-language, and let $\alpha : \mathsf{Type}_{u'}$ and $S : \alpha \to \Sigma$. We define the type of *L-bounded formulae on $\alpha : S$* with free variables indexed by α, and n additional free variables of sorts $f : \mathsf{Fin}\,n \to \Sigma$, denoted $\mathsf{BForm}^f_{\alpha:S}$, by the following constructors.

- $\mathsf{falsum} : \mathsf{BForm}^f_{\alpha:S}$;
- $\mathsf{equal} : \prod_{A:\Sigma} \mathsf{Term}_{\alpha\oplus\mathsf{Fin}\,n:S\oplus f}\,A \to \mathsf{Term}_{\alpha\oplus\mathsf{Fin}\,n:S\oplus f}\,A \to \mathsf{BForm}^f_{\alpha:S}$;
- $\mathsf{rel} : \prod_{m:\mathbb{N}} \prod_{B:\mathsf{Fin}\,m\to\Sigma} \prod_{R:\mathsf{Relations}(m,B)} \left(\prod_{i:\mathsf{Fin}\,m} \mathsf{Term}_{\alpha\oplus\mathsf{Fin}\,n:S\oplus f}\,B(i)\right) \to \mathsf{BForm}^f_{\alpha:S}$;
- $\mathsf{imp} : \mathsf{BForm}^f_{\alpha:S} \to \mathsf{BForm}^f_{\alpha:S} \to \mathsf{BForm}^f_{\alpha:S}$; and
- $\mathsf{all} : \prod_{A:\Sigma} \mathsf{BForm}^{A::f}_{\alpha:S} \to \mathsf{BForm}^f_{\alpha:S}$,

where the syntax $A :: f$ denotes the cons operation on maps from $\mathsf{Fin}\,n$.[2] An *L-formula on $\alpha : S$* with free variables indexed by α is an inhabitant of $\mathsf{BForm}^{\#[]}_{\alpha:S}$.[3] An *$L$-sentence* is an L-formula with free variables indexed by $\mathsf{Empty} : f$, where f is the unique function $\mathsf{Empty} \to \Sigma$. An *$L$-theory* is a set of L-sentences.

[2]This is implemented in mathlib as `Fin.cons`.
[3]The expression $\#[]$ is mathlib's syntax for the unique map $\mathsf{Fin}\,0 \to \Sigma$.

8.2 Term models

Definition 8.8. Let L be a Σ-language. A *1-L-formula* of sort A is an L-bounded formula on Empty with one additional free variable of sort $f : \mathsf{Fin}\,1 \to \Sigma$ given by $f(x) = A$.

Definition 8.9. Let L be a Σ-language. The *witness symbols* for L is the language L_W consisting of a single constant of sort A for every 1-L-formula of sort A.

Proposition 8.10. Let M be a nonempty L-structure. Then M has an L_W-structure such that for each 1-L-formula ϕ of sort A, there is a constant symbol $c : \mathsf{Functions}(0, \#[], A)$ such that

$$\forall x : M_A,\ \phi(x) \to \phi(c)$$

Proof. We define the interpretation of the constant for ϕ to be some $x : M$ such that $M \vDash \phi(x)$ if one exists, or an arbitrary $y : M$ otherwise.[4] This defines an L_W-structure for M, and clearly M satisfies the required property. □

Definition 8.11. For each $n : \mathbb{N}$, we define

$$L^{(0)} = L; \quad L^{(n+1)} = L^{(n)} \oplus (L^{(n)})_W$$

This forms a directed diagram of languages, which has a colimit $L^{(\omega)}$. There are natural morphisms $L \to L^{(\omega)}$ and $L^{(n)} \to L^{(\omega)}$ which are expansions on M.

Proposition 8.12. Let M be a nonempty L-structure. Then M has an $L^{(\omega)}$-structure such that for each 1-$L^{(\omega)}$-formula ϕ of sort A, there is a constant symbol $c : \mathsf{Functions}(0, \#[], A)$ such that

$$\forall x : M_A,\ \phi(x) \to \phi(c)$$

Then, by the Tarski–Vaught test (which must be proven in the many-sorted case), the set $N \subseteq M$ comprised of all of the interpretations of $L^{(\omega)}$-terms, is the domain of an $L^{(\omega)}$-elementary substructure of M.

This will take substantial work.

8.3 Ambiguity

Definition 8.13. Let L be an $\mathbb{N}$-language. A *type raising morphism* is a map of languages $L \xrightarrow{\mathsf{succ}} L$, where $\mathsf{succ} : \mathbb{N} \to \mathbb{N}$ is the successor function.

The remainder of this chapter is left unfinished until we finish the main part of the project.

[4] Use mathlib's `Classical.epsilon`.

Appendix A

Auxiliary results

A.1 Relations

Definition A.1. Let $R : \sigma \to \tau \to \mathsf{Prop}$. We define

- the *image* of R to be the set $\operatorname{im} R = \{y : \tau \mid \exists x : \sigma,\ x\ R\ y\}$;
- the *coimage* of R to be the set $\operatorname{coim} R = \{x : \sigma \mid \exists y : \tau,\ x\ R\ y\}$ (note that this is *not* the same as the category-theoretic coimage of a morphism);
- the *field* of R to be the set $\operatorname{field} R = \operatorname{coim} R \cup \operatorname{im} R$;
- the *image of R on s* : $\mathsf{Set}\,\sigma$ to be the set $\operatorname{im} R|_s = \{y : \tau \mid \exists x \in s,\ x\ R\ y\}$;
- the *coimage of R on t* : $\mathsf{Set}\,\tau$ to be the set $\operatorname{coim} R|_t = \{x : \sigma \mid \exists y \in t,\ x\ R\ y\}$;
- the *converse* of R to be the relation $R^{-1} : \tau \to \sigma \to \mathsf{Prop}$ such that $y\ R^{-1}\ x$ if and only if $x\ R\ y$;
- if $S : \tau \to \upsilon \to \mathsf{Prop}$, the *composition* $S \circ R : \sigma \to \upsilon \to \mathsf{Prop}$ is the relation given by $x\ (S \circ R)\ z$ if and only if there is $y : \tau$ such that $x\ R\ y$ and $y\ S\ z$;
- for a natural number n, the *nth power* of $R : \tau \to \tau \to \mathsf{Prop}$ is defined by $R^{n+1} = R \circ R^n$ and R^0 is the identity relation on $\operatorname{coim} R$;
- for an integer n, the *nth power* of $R : \tau \to \tau \to \mathsf{Prop}$ is defined by $R^{(n:\mathbb{Z})} = R^n$ and $R^{-(n:\mathbb{Z})} = (R^n)^{-1}$ for $n : \mathbb{N}$.[1]

We say that R is

- *injective*, if $s_1\ R\ t, s_2\ R\ t$ imply $s_1 = s_2$;
- *surjective*, if for every $t : \tau$, there is some $s : \sigma$ such that $s\ R\ t$;
- *coinjective*, if $s\ R\ t_1, s\ R\ t_2$ imply $t_1 = t_2$;
- *cosurjective*, if for every $s : \sigma$, there is some $t : \tau$ such that $s\ R\ t$;
- *functional*, if R is coinjective and cosurjective, or equivalently, for every $s : \sigma$ there is exactly one $t : \tau$ such that $s\ R\ t$;

[1] This may be implemented using `Int.negInduction` from mathlib.

- *cofunctional*, if R is injective and surjective, or equivalently, for every $t : \tau$ there is exactly one $s : \sigma$ such that $s \; R \; t$;
- *one-to-one*, if R is injective and coinjective;
- *bijective*, if R is functional and cofunctional;
- *permutative*, if $R : \tau \to \tau \to \mathsf{Prop}$ is one-to-one and has equal image and coimage.

We also define the *graph* of a function $f : \sigma \to \tau$ to be the functional relation $R : \sigma \to \tau \to \mathsf{Prop}$ given by $(x, y) \in R$ if and only if $f(x) = y$. Most of these definitions are from `https://www.kylem.net/math/relations.html`, and most of these are in mathlib under `Mathlib.Logic.Relator`.

Proposition A.2.

1. $R : \tau \to \tau \to \mathsf{Prop}$ is permutative if and only if it is one-to-one and for all $x : \tau$, there exists y such that $x \; R \; y$ if and only if there exists y such that $y \; R \; x$.
2. If $R, S : \tau \to \tau \to \mathsf{Prop}$ are permutative and $\mathsf{coim}\, R \cap \mathsf{coim}\, S = \emptyset$, then $R \sqcup S$ is permutative and has coimage $\mathsf{coim}\, R \cup \mathsf{coim}\, S$.
3. If $R, S : \tau \to \tau \to \mathsf{Prop}$ are permutative and $\mathsf{coim}\, R = \mathsf{coim}\, S$, then $R \circ S$ is permutative and has coimage equal to that of R and S.
4. If R is permutative, then $\mathsf{coim}\, R^n = \mathsf{coim}\, R = \mathsf{im}\, R = \mathsf{im}\, R^n$ for any natural number or integer n.
5. If R is permutative and $s_1, s_2 \subseteq \mathsf{coim}\, R$, then the image of R on s_1 is equal to s_2 if and only if the coimage of R on s_2 is equal to s_1.

Definition A.3. Let $s : \mathsf{Set}\, \tau$. An *orbit restriction* for s (over some type σ) consists of a set $t : \mathsf{Set}\, \tau$ disjoint from s, a function $f : \tau \to \sigma$, and a permutation $\pi : \sigma \simeq \sigma$, such that for each $u : \sigma$, the set $t \cap f^{-1}(u)$ has cardinality at least $\max(\aleph_0, \#s, \#d)$.

An orbit restriction encapsulates information about how orbits should be completed.

- t is the *sandbox*, the set inside which all added items must reside.
- f is a *categorisation function*, placing each value $x : \tau$ into a category $u : \sigma$.
- π is a permutation of categories. If x has category u, then anything that x is mapped to should have category $\pi(u)$.

Proposition A.4 (completing restricted orbits)**.** Let $R : \tau \to \tau \to \mathsf{Prop}$ be a one-to-one relation, and let (t, f, π) be an orbit restriction for $\mathsf{field}\, R$ over some type σ. Then there is a permutative relation T such that

- $\mathsf{coim}\, T \subseteq \mathsf{field}\, R \cup t$;
- $\#\, \mathsf{coim}\, T \le \max(\aleph_0, \#\, \mathsf{coim}\, R)$;
- $R \le T$; and
- if $(x, y) \in T$, then $(x, y) \in R \vee f(y) = \pi(f(x))$.

Proof. For each $u : \sigma$, define an injection $i_u : \mathsf{field}\, R \times \mathbb{N} \to \tau$ where $\mathsf{im}\, i_u \subseteq t \cap f^{-1}(u)$. Define a relation S on τ by the following constructors.

$$\begin{aligned}
\forall x \in \operatorname{coim} R \setminus \operatorname{im} R, (i_{\pi^{-1}(f(x))}(x,0), x) \in S \\
\forall n : \mathbb{N}, \forall x \in \operatorname{coim} R \setminus \operatorname{im} R, (i_{\pi^{-n-2}(x)}(x, n+1), i_{\pi^{-n-1}(x)}(x,n)) \in S \\
\forall x \in \operatorname{im} R \setminus \operatorname{coim} R, (x, i_{\pi(f(x))}(x,0)) \in S \\
\forall n : \mathbb{N}, \forall x \in \operatorname{im} R \setminus \operatorname{coim} R, (i_{\pi^{n+1}(f(x))}(x,n), i_{\pi^{n+2}(f(x))}(x, n+1)) \in S
\end{aligned}$$

Note that $(x, y) \in S$ implies $f(y) = \pi(f(x))$. Finally, as $R \sqcup S$ is permutative, it satisfies the required conclusion. □

Proposition A.5 (completing orbits)**.** Let $R : \tau \to \tau \to \mathsf{Prop}$ be a one-to-one relation. Let $s : \mathsf{Set}\,\tau$ be an infinite set such that $\#\,\mathsf{field}\,R \leq \#s$, and $\mathsf{field}\,R$ and s are disjoint. Then there is a permutative relation S such that $R \leq S$ and $\mathsf{coim}\,S \subseteq \mathsf{field}\,R \cup s$.

Proof. Define the orbit restriction (s, f, π) for $\mathsf{field}\,R$ over Unit. Note that for this to be defined, we used the inequality

$$\#s \geq \max(\aleph_0, \#\,\mathsf{field}\,R)$$

Use proposition A.4 to obtain a permutative relation T extending R defined on $\mathsf{field}\,R \cup s$. □

A.2 Cardinal arithmetic

Lemma A.6 (mathlib)**.** Let $\#\mu$ be a strong limit cardinal. Then there are precisely $\#\mu$-many subsets of μ of size strictly less than $\mathsf{cof}(\mathsf{ord}(\#\mu))$.

Proof. Endow μ with its initial well-ordering. Each such subset is bounded in μ with respect to this well-ordering as its size is less than $\mathsf{cof}(\mathsf{ord}(\#\mu))$. So it suffices to prove there are precisely $\#\mu$-many bounded subsets of μ.

$$\begin{aligned}
\#\{s : \mathsf{Set}\,\mu \mid \exists \nu : \mu, \forall x \in s, x < \nu\} &= \# \bigcup_{\nu : \mu} \{s : \mathsf{Set}\,\mu \mid \forall x \in s, x < \nu\} \\
&\leq \sum_{\nu : \mu} \#\{s : \mathsf{Set}\,\mu \mid \forall x \in s, x < \nu\} \\
&= \sum_{\nu : \mu} \#\{s : \mathsf{Set}\{x : \mu \mid x < \nu\}\} \\
&= \sum_{\nu : \mu} \underbrace{2^{\#\{x : \mu \mid x < \nu\}}}_{< \mu} \\
&\leq \mu
\end{aligned}$$

□

Bibliography

[1] Scott Fenton. New Foundations set theory developed in metamath, 2015. URL: https://us.metamath.org/nfeuni/mmnf.html.

[2] Theodore Hailperin. A set of axioms for logic. *Journal of Symbolic Logic*, 9(1):1–19, 1944. doi:10.2307/2267307.

[3] M. Randall Holmes. The Equivalence of NF-Style Set Theories with “Tangled” Theories; The Construction of ω-Models of Predicative NF (and more). *The Journal of Symbolic Logic*, 60(1):178–190, 1995. URL: http://www.jstor.org/stable/2275515.

[4] M. Randall Holmes and Sky Wilshaw. NF is Consistent, 2024. arXiv:1503.01406.

[5] W. V. Quine. New Foundations for Mathematical Logic. *American Mathematical Monthly*, 44:70–80, 1937. URL: https://api.semanticscholar.org/CorpusID:123927264.

[6] Ernst P. Specker. The Axiom of Choice in Quine’s New Foundations for Mathematical Logic. *Proceedings of the National Academy of Sciences of the United States of America*, 39(9):972–975, 1953. URL: http://www.jstor.org/stable/88561.

[7] Ernst P. Specker. Typical Ambiguity. In Ernst Nagel, editor, *Logic, Methodology and Philosophy of Science*, pages 116–123. Stanford University Press, 1962.

[8] Sky Wilshaw, Yaël Dillies, et al. New Foundations is consistent, 2022–2024. URL: https://leanprover-community.github.io/con-nf/.