

Demystifying ramified type theory and the axiom of reducibility

M. Randall Holmes

November 12, 2023

In this document, I will reveal the exact equivalence of Russell's ramified theory of types from Principia to a far simpler system, the simple typed theory of sets with a predicativity restriction, and I will demonstrate that the axiom of reducibility in this context is exactly equivalent to something much simpler, namely, the axiom of set union.

I begin with simple typed theories of sets. They are simple.

TST is a theory with sorts indexed by the natural numbers, with primitive predicates equality and membership. We say that each variable x has a sort $\mathbf{type}(x)$ which is a natural number, and that $x = y$ is well-formed iff $\mathbf{type}(x) = \mathbf{type}(y)$ and $x \in y$ is well-formed iff $\mathbf{type}(x) + 1 = \mathbf{type}(y)$. We assume that we have countably many variables of each type.

The axiom schemes of TST are: extensionality, all assertions of the shape $(\forall xy : x = y \leftrightarrow (\forall z : z \in x \leftrightarrow z \in y))$ are axioms [There is effectively one of these for each choice of the type of z]; comprehension, all assertions of the shape $(\exists A : (\forall x : x \in A \leftrightarrow \phi))$ are axioms [The object A , unique by extensionality, may be denoted by $\{x : \phi\}$, which has type one higher than that of x].

Note that nothing but the type discipline distinguishes these from the axioms of naive set theory.

In conformity with the needs of Principia, we can include the axiom of infinity in TST for our purposes here. The collection of finite sets can be defined (polymorphically, this is a definition scheme) as

$$\{F : (\forall I : \emptyset \in I \wedge (\forall A \in I : \forall x : A \cup \{x\} \in I) \rightarrow F \in I)\}$$

Of course this definition has prerequisite definitions, $\emptyset = \{x : x \neq x\}$ [this definition is polymorphic: the type of empty set referred to is deduced from

context] and $A \cup \{x\} = \{y : y \in A \vee y = x\}$. The axiom scheme of infinity then asserts

$$\{x : x = x\} \notin \{F : (\forall I : \emptyset \in I \wedge (\forall A \in I : \forall x : A \cup \{x\} \in I) \rightarrow F \in I)\} :$$

the universal set (over each type) is not finite (in the sense appropriate to subsets of that type).

The predicative simple theory of types (TSTP) has the same axiom scheme of extensionality and a restricted scheme of comprehension. The restriction is that $(\exists A : (\forall x : x \in A \leftrightarrow \phi))$ is an axiom as long as no variable appearing in ϕ is of type higher than the type of A , and any variable of the type of A appearing in ϕ is free (not quantified over). This is exactly analogous to the predicativity restriction in Principia, but phrased for a much simpler situation.

TSTP plus the Axiom (scheme) of Set Union, which asserts the existence for any A of appropriate type of $\bigcup A = \{x : (\exists a \in A : x \in a)\}$, is exactly equivalent to TST. Define $\iota(x)$ as $\{y : y = x\}$. We can then conveniently observe that $\{\iota^n(x) : \phi\}$ exists in TST for large enough n for any ϕ (the point being that the type of the variable over which we abstract can thus be made higher than that of any bound variable in ϕ).

Some paraphrase may be felt to be needed to make this absolutely clear. Let $y = \{x\}$ abbreviate the formula $(\forall z : z \in x \leftrightarrow z = y)$. $\{\iota(x) : \phi\}$ abbreviates $\{x_1 : (\exists x : x_1 = \{x\} \wedge \phi)\}$. Note that the additional variable which would be revealed by expanding $x_1 = \{x\}$ is of a type which already appears in ϕ . $\{\iota^n(x) : \phi\}$ abbreviates n applications of this operation: the thing to observe is that if n is taken to be large enough, the variable replacing $\iota^n(x)$ when we expand this will be the variable of highest type in the abstract, so the abstract will be of the form allowed in TSTP. Then observe that $\bigcup(\{\iota(x) : \phi\}) = \{x : \phi\}$, and more generally $\bigcup^n(\{\iota^n(x) : \phi\}) = \{x : \phi\}$, so the axiom of union together with the comprehension axiom of TSTP gives the full comprehension axiom of TST (which in its turn justifies both union and predicative comprehension, so there is an exact equivalence).

We discuss a method of defining n -tuples in TSTP with an essential formal advantage. We define $\langle x, y \rangle$ as $\{\{x\}, \{x, y\}\}$ as usual. We define individual natural numbers [in each type with index at least 2] in a way standard for this theory: $0 = \{\emptyset\}$; $n + 1 = \{A \cup \{x\} : A \in n \wedge x \notin A\}$. An n -tuple $\langle x_0, \dots, x_{n-1} \rangle$ is defined as $\{\langle 0, x_0 \rangle, \langle 1, x_1 \rangle, \dots, \langle n-1, x_{n-1} \rangle\}$. An n -tuple

(x_0, \dots, x_{n-1}) is of type three higher than the common type of the x_i 's. Note that 0-tuples, 1-tuples, and 2-tuples make sense. 0-tuples are empty, 1-tuples (x_0) are not the same as x_0 (they are not even of the same type) and 2-tuples (x_0, x_1) are not to be confused with pairs $\langle x_0, x_1 \rangle$. We note that we are implicitly requiring that the type of the x_i 's be at least 2, since that is the minimal type of a natural number as we have defined it.

It is possible to define an ordered pair $\langle x, y \rangle$ in TSTP which is the same type as its projections x and y , for x, y above a constant type. The definition is a variation of Quine's ordered pair, modified because the usual definition of \mathbb{N} violates the predicativity restrictions of TSTP. This would then allow (x_0, x_1, \dots, x_n) to be defined simply as $\langle x_0, (x_1, \dots, x_n) \rangle$, the ordered pair and the 2-tuple coinciding in this case, and the 1-tuple (x) coinciding with x . so that n -tuples would be of the same type as their projections as well. The overhead of the modified Quine definition of the ordered pair is in our view rather high, though, so we will use the familiar pair. The apparently simpler way of defining n -tuples given in this paragraph is incompatible with the type system, and if it is modified to suit the type system, it will have the nasty consequence that the type displacement between a tuple and its projections will depend on the length of the tuple, which we need to avoid.

We now discuss the theory of types in Russell's Principia (but not presented in the unintelligible way he presents it). This is a theory of propositional functions with arbitrary arity, and with an elaborate type scheme. There is a type of individuals for which our notation is 0. The order of type 0 is 0. A type notation of order $k > 0$ is a sequence of type notations of order $< k$ with k appended. A type notation (t_0, \dots, t_{n-1}, k) is inhabited by propositional functions f for which $f(x_0, \dots, x_{n-1})$ is defined (and a proposition) just in case each x_i is of type t_i .

A propositional function may be thought of as a reified n -ary relation on arguments of the types listed in the notation for its type.

The comprehension axiom of Russell's theory asserts that for any formula $\phi[x_0, \dots, x_{n-1}]$ with x_i of type t_i , there is a propositional function f of type (t_0, \dots, t_{n-1}, k) such that $f(x_0, \dots, x_{n-1}) \leftrightarrow \phi[x_0, \dots, x_{n-1}]$ holds for any values of the x_i 's if ϕ contains no constants or free variables of types of order greater than k , and no bound (quantified) variables of type $\geq k$. It should be noted that he doesn't say that he has a comprehension axiom: but this is the one implicit in what he does say.

This is not Russell's presentation. Russell defines no notation for types whatsoever, which makes his book a bit hard to follow in places. We are

going to brutally assume extensionality (though we don't have to talk about it). He thought this was an Issue. But the handling of types, orders, and restrictions on quantification in definitions of propositional functions follows Russell. [Russell appears to identify his propositional functions with actual pieces of text, though it is not clear he is doing exactly this, an aspect we are just not going to talk about.]

Russell's theory interprets TSTP absolutely directly (as long as we assume extensionality, which we do). Type 0 of TSTP may be taken to be the type of individuals or any other fixed type one chooses. If type n of TSTP is interpreted by a type t of order k , type $n + 1$ is interpreted by type $(t, k + 1)$. If x is a type i variable and y is a type $i + 1$ variable, we interpret $x \in y$ (in the TSTP sense) as $y(x)$. It is clear from our description of the interpretation that this is well-typed. It should also be clear that the translation of the comprehension axiom of TSTP will hold in Russell's theory as we present it.

Now the really interesting thing is that TSTP interprets Russell's theory. We describe the interpretation. Each type of order k is implemented by a subset of type $4k + 2$ in TSTP. The type of individuals is simply interpreted as type 2 of TSTP, or any desired subset of type 2. For any type notation τ , let $\text{order}(\tau)$ be defined as its order.

The implementation $\text{set}(t_0, \dots, t_{n-1}, k)$ in TSTP of a type in the PM system is defined as

$$\mathcal{P}\{(\iota^{4(k-\text{order}(t_1)-1)}(x_1), \dots, \iota^{4(k-\text{order}(t_{n-1})-1)}(x_{n-1})) : \bigwedge_i x_i \in \text{set}(t_i)\}.$$

Please note that the recursion here is metatheoretical: we can describe each of these sets in the language of TSTP but the recursion itself cannot be described in the language of TSTP, only each instance of it.

If f is in $\text{set}(t_0, \dots, t_{n-1}, k)$ and $(\bigwedge_i : x_i \in \text{set}(t_i))$, what we mean by $f(x_0, \dots, x_{n-1})$ (in the language of Russell's theory) is

$$(\iota^{4(k-\text{order}(t_1)-1)}(x_1), \dots, \iota^{4(k-\text{order}(t_{n-1})-1)}(x_{n-1})) \in f$$

The multiplier of 4 is needed because $(x_0, \dots, x_{n-1}) \in f$, the simple membership of a tuple in a representation of an n -ary relation, has a type displacement of 4 from the type of the x_i to the type of f . Each projection is being adjusted by application of the singleton operator to the same type, 4 types below the type $4k + 2$ of the implementation of f in TSTP.

The complicated thing that we need to verify is that Russell's comprehension axiom (as we call it, he might not call it that) holds in the interpretation.

The executive summary is that it works correctly because the implementation of a type of order k as a set is always a subset of type $4k + 2$. Definitions of propositional functions can be checked: if they meet the conditions described in the comprehension axiom, their translations meet the conditions required for TSTP. It might be a little disturbing that the order k type and the order $k + 1$ type over the same sequence of arguments in the Russell sense are not relations on the same domain in our sense: the domain of the type of order $k + 1$ is the elementwise image under ι^4 (the fourfold singleton operation) of the domain of the type of order k . The domains are exactly correlated, and the possible failure of Union in TSTP means that it is possible that there are more relations on images under ι^4 on a domain than those that are correlated in the obvious way (elementwise image under ι^4) with relations on the domain.

And if the axiom of set union holds in TSTP, the theory interpreted is the simplified theory of types of Ramsey: relations on the image of a domain under fourfold singleton correlate exactly with relations on the domain, and there is no reason to draw the distinctions. The representation of the simplified theory can be decluttered: all types can have minimal order, and the definitions become much more natural with less mention of ι^4 (a type with arguments of different orders will still require ι^4 in its translation).

One might ask, what does the axiom of reducibility have to do with the axiom of union? On the logical level, both are methods of implementing impredicative quantification. The actual way that reducibility on the PM side becomes union on the TSTP side can be sketched most easily by considering types of classes over a single type τ . If τ is of order k , there are types $(\tau, k + i)$ for each positive integer i which you can think of as inhabited by subclasses of type τ (unary predicates of type τ objects). As i increases, one may discover more subclasses of type τ . In TSTP, the set $\mathbf{set}(\tau)$ representing type τ (a subclass of type $4k + 2$) has an unequivocal power set $\mathcal{P}(\mathbf{set}(\tau))$ of type $4k + 2$ and never itself acquires any more subsets. The representation $\mathbf{set}((\tau, k + 1))$ is for technical reasons actually the type $4k + 6$ set $\mathcal{P}(\{(x) : x \in \mathbf{set}(\tau)\})$. Remember that (x) is not identified with x (it is actually $\{0, x\}$). The representation $\mathbf{set}((\tau, k + i))$ is $\mathcal{P}(\{(\iota^{4(i-1)}(x)) : x \in \mathbf{set}(\tau)\})$, and the fact that this type can have inhabitants not correlated with the inhabitants of types of lower order depends on the fact that in TSTP there can be sets of iterated singletons of elements of a set A which are not elementwise images under the iterated singleton operation of subsets of A (precisely because union can fail: union tells you that the subsets of the elementwise image

under singleton of A are exactly the elementwise images under singleton of subsets of A): this is how the idea in PM that there can be alternative power sets with more inhabitants is implemented in TST, where a set has only one power set of the next higher type.

This is made less transparent by the need to use cumbersome machinery with added type increments to represent tuples, because PM supports types of relations of arbitrary arity. This note might be improved by providing as an alternative an account of how a type level pair improves the situation (which would require the demonstration that the Quine pair can be modified to be predicative).