

# The typed theory of sets as an untyped theory

Randall Holmes (with deference to Quine and Resnik, and thanks to Thomas

January 19, 2024

## Contents

<b>1</b>	<b>Introduction</b>	<b>1</b>
<b>2</b>	<b>Our axioms</b>	<b>2</b>
<b>3</b>	<b>Theorems about general structure</b>	<b>4</b>
<b>4</b>	<b>The axiom of infinity and arithmetic</b>	<b>8</b>
<b>5</b>	<b>Relations and functions: more mathematical technology</b>	<b>10</b>
<b>6</b>	<b>Our theory and a general type theory</b>	<b>13</b>
<b>7</b>	<b>Finite axiomatization of the theory</b>	<b>18</b>
<b>8</b>	<b>Quine’s original axioms</b>	<b>19</b>
<b>9</b>	<b>The system of Resnik</b>	<b>20</b>

## 1 Introduction

We argue here that the simple typed theory of sets can be very neatly presented as an untyped (one-sorted) theory. This was first brought to my attention by Thomas Forster, but the original idea is due to Quine, who discusses it in under the heading “Type theory with general variables” in *Set theory and its logic*. We will discuss Quine’s work in this area (and later work by Resnik, closer to ours) later in the paper.

## 2 Our axioms

The theory we present is a one-sorted theory with equality and membership as primitive predicates. One more primitive notion is introduced in this section and one in the next.

We call the theory TTGV (for “type theory with general variables”) following Quine, though we will see that our TTGV differs significantly from Quine’s similarly motivated theory.

We refer to general objects of this theory simply as objects.

An important feature of any class of abstract objects is their identity conditions.

**Definition:** We say that an object  $x$  is a *nonempty set* iff  $(\exists y : y \in x)$ .

Nonempty sets are equal if and only if they have the same elements:

**Axiom of weak extensionality:**

$$(\forall xyz : z \in x \wedge (\forall w : w \in x \leftrightarrow w \in y) \rightarrow x = y)$$

We list some informal assertions which are not axioms for us but motivation for what follows.

1. Objects are of various kinds (probably more than one). Each object has one kind.
2. Sets of objects are sets of objects of a particular kind: all of the elements of a given set are of the same kind.
3. Each kind is a set (there is a set of all the objects of a given kind).
4. Any property of objects of a given kind determines a set.

A formal development follows.

**Definition:** We say that  $x$  and  $y$  are of the same type (or the same kind), written  $x \sim y$ , just in case  $(\exists z : x \in z \wedge y \in z)$ . This is motivated by the idea that all elements of a given set are of the same kind. This definition appears in Quine’s original treatment and is the central idea of this approach. One might expect this to be an equivalence relation, and indeed it is, but this will require proof. We do not assume this at this point.

**Axiom of types and definition of the type operator:**

$$(\forall x : \exists \tau : x \in \tau \wedge (\forall y : y \in \tau \leftrightarrow x \sim y))$$

We define  $\tau(x)$  as the unique  $\tau$  such that  $(\forall y : y \in \tau \leftrightarrow x \sim y)$ , noting that  $x \in \tau(x)$  so it is nonempty, and uniqueness is enforced by extensionality. The notation  $V_{\tau(x)}$  may also be used, suggesting that a type is a sort of universe.

The reader may have heard rumors that not all sets are nonempty.

**Axiom of empty sets and definition of sethood:** We introduce a primitive operation: for any type  $\tau(x)$  there is an object  $\emptyset_{\tau(x)}$ , which we call the empty set of the type of  $x$ . The axiom governing this construction is

$$\emptyset_{\tau(x)} \sim \tau(x) \wedge (\forall y : y \notin \emptyset_{\tau(x)}).$$

We define  $\mathbf{set}(A)$  [read “ $A$  is a set”] as  $(\exists x : x \in A \vee \emptyset_{\tau(x)} = A)$ . Notice that it is a consequence of this definition and the extensionality axiom that two sets are equal iff they have the same elements and are of the same type. Something which is not a set we call an atom. We define  $x \subseteq y$  as

$$\mathbf{set}(x) \wedge \mathbf{set}(y) \wedge x \sim y \wedge (\forall z : z \in x \rightarrow z \in y).$$

**Axiom of comprehension:** For any object  $x$  and any<sup>1</sup> formula  $\phi$ ,

$$(\exists A : A \sim \tau(x) \wedge (\forall y : y \in A \leftrightarrow (y \sim x \wedge \phi))).$$

**Definition:** We define  $\{y \in \tau(x) : \phi\}$  as the unique  $A$  such that

$$\mathbf{set}(A) \wedge A \sim \tau(x) \wedge (\forall y : y \in A \leftrightarrow (y \sim x \wedge \phi))$$

---

<sup>1</sup>Forster, in an earlier presentation, originally restricted the comprehension scheme to “stratified” formulas (ones which make sense in the typed theory of sets) but this appears to us to somewhat hide what is going on. It is telling that if we omit all suggestions that we are talking about the typed theory of sets, the fact that we are talking about the typed theory of sets nonetheless falls out. In effect, the only information that is obtained by allowing unstratified formulas in comprehension is the conclusion drawn shortly that a type is disjoint from the types which include its iterated power sets.

We need one more natural axiom and it will turn out that we have implemented the classical typed theory of sets, though this may not be obvious.

**Axiom of binary union:**

$$(\forall xy : x \sim y \rightarrow (\exists z : (\forall w : w \in z \leftrightarrow w \in x \vee w \in y)))$$

Notice that this is trivially true if either  $x$  or  $y$  has no elements (and in particular if either is an atom). Thus we can restrict our attention to sets in the following

**Definition:** For any sets  $x, y$ , with  $x \sim y$ , we define  $x \cup y$  as the unique set  $z$  such that  $(\forall w : w \in z \leftrightarrow w \in x \vee w \in y)$ .

**Definitions of sorts of non-set:** We say that  $x$  is an *atom* iff  $\neg \text{set}(x)$ . note that atoms have no elements. We say that  $x$  is an *individual* iff  $\neg(\exists y : x \sim \tau(y))$ . An individual is not a set, and no element of the type it belongs to is a set. Our axioms do not provide for the existence of either atoms or individuals, nor for the assumption that all individuals are of the same type, but these are important formal possibilities.

### 3 Theorems about general structure

We continue the development of TTGV. We need to do some bookkeeping.

**The relation of being of the same type is an equivalence relation:**

$x \sim x$  is evident because  $x \in \tau(x) \wedge x \in \tau(x)$ . I enjoyed writing that!

$x \sim y$  is equivalent to  $(\exists z : x \in z \wedge y \in z)$  which is equivalent to  $(\exists z : y \in z \wedge x \in z)$  which is equivalent to  $y \sim x$ .

If  $x \sim y$  and  $y \sim z$ , then  $x \sim y$  and  $z \sim y$ , that is,  $x \in \tau(y)$  and  $z \in \tau(y)$ , so  $x \sim z$ .

Thus types  $\tau(x)$  are equivalence classes under  $\sim$ , and make up a partition of the universe of our theory.

**Definition scheme for local type hierarchy:** We define  $\tau^1(x)$  as  $\tau(x)$  and  $\tau^{n+1}(x)$  as  $\tau(\tau^n(x))$ . This is purely a convenient scheme: the natural number indices here cannot be quantified over and do not represent

objects of our theory. We will define natural numbers as objects of the theory later, but these superscripts do not refer to them.

This notation can be extended (partially) to non-positive integer superscripts. We define  $\tau^0(x)$  as the unique type of which  $x$  is a subset, if there is one, and otherwise as  $x$  itself (in case  $x$  is an individual). We define  $\tau^{i-1}(x)$  for  $i < 1$  as the unique type belonging to  $\tau^i(x)$ , if there is one, and otherwise leave it undefined (this notation gives us no hint as to the individual to choose to be  $\tau^{i-1}(x)$  in case  $\tau^i(x)$  is a type of individuals, so we leave this vague).

When we refer to  $\tau^n(x)$  we normally intend  $n$  to be a positive integer, and will advise the reader if our intentions differ in some part of the text.

**The Russell non-paradox:** Define  $R_x$  as  $\{y \in \tau(x) : y \not\in y\}$ . We know that  $R_x \sim \tau(x)$ , so  $R_x \in \tau^2(x)$ . We know that  $R_x \in R_x$  if and only if  $R_x \in \tau(x) \wedge R_x \not\in R_x$ , from which we conclude that  $R_x \not\in \tau(x)$  (as contradiction would otherwise follow), so  $R_x \not\sim x$ , from which we draw the useful conclusion that  $x \not\sim \tau(x)$ . This means that  $\tau(x)$  and  $\tau^2(x)$  are disjoint, since any  $y$  which was in both sets would stand in the relation  $\sim$  to both  $x$  and  $\tau(x)$ .

This can be extended to show that  $\tau^n(x)$  is disjoint from  $\tau(x)$  for any concretely given natural number  $n$ . Define  $\{x\} = \iota(x)$  as

$$\{y \in \tau(x) : y = x\},$$

and define (schematically)  $\iota^0(x)$  as  $x$  and  $\iota^{n+1}(x)$  as  $\iota(\iota^n(x))$ . Note that  $\iota(x) \in \tau^2(x)$  and in general  $\iota^n(x) \in \tau^{n+1}(x)$ . Now define  $R_x^n$  as  $\{\iota^n(y) \in \tau^{n+1}(x) : \iota^n(y) \not\in y\}$ , or, in a more finicky format to make it clear that our axioms support this,

$$\{z \in \tau^{n+1}(x) : (\exists y : z = \iota^n(y) \wedge z \not\in y)\}.$$

$R_x^n \in \tau^{n+2}(x)$  and we see that  $\iota^n(R_x^n) \in R_x^n \leftrightarrow \iota^n(R_x^n) \in \tau^{n+1}(x) \wedge \iota^n(R_x^n) \not\in R_x^n$ , from which we conclude that  $\iota^n(R_x^n) \not\in \tau^{n+1}(x)$ , from which  $R_x^n \not\in \tau(x)$ , so  $R_x^n \not\sim x$ , from which we can conclude that  $\tau^{n+2}(x)$  cannot meet  $\tau(x)$ .

It is straightforward to establish that there is no particular use for the notation  $R_x^n$  going forward, as in fact  $R_x^n = \tau^{n+1}(x)$ .

**A final primitive for managing type hierarchy:** We add one more primitive notion and axioms governing it, for reasons which will become evident later.

**type order:** We introduce a primitive relation  $<_\tau$ ,  $x <_\tau y$  being read, “ $x$  is of lower type than  $y$ ”.

**axioms of type order:** These axioms govern this strict partial order.

1.  $(\forall x : \neg x <_\tau x)$ :  $<_\tau$  is irreflexive.
2.  $(\forall xyz : x <_\tau y \wedge y <_\tau z \rightarrow x <_\tau z)$ :  $<_\tau$  is transitive.
3.  $(\forall xy : x <_\tau y \leftrightarrow \tau(x) <_\tau \tau(y))$   $<_\tau$  on a pair of objects agrees with  $<_\tau$  on their respective types.
4.  $(\forall x : x <_\tau \tau(x))$  Type hierarchies increase relative to  $<_\tau$ .

The immediate application of this is to show using a finite set of axioms that  $\pi(x)$  is disjoint from  $\pi^i(x)$  for each concretely given  $i > 1$ : we have just proved this, but using countably many distinct instances of comprehension. We should see that this is basically all that this set of axioms does. It is a technical device to make the theory finitely axiomatizable; we shall see that it achieves this rather elegantly. There is no obstruction to  $<_\tau$  appearing in instances of the comprehension scheme (but we will see that this adds no sets or indeed any additional information about the type hierarchies at all!)

**Power sets and principal ultrafilters:** If  $x$  is a set we can define  $\mathcal{P}(x)$ , the power set of  $x$ , as

$$\{A \in \tau^2(x) : \text{set}(A) \wedge (\forall y : y \in A \rightarrow y \in x)\}.$$

If  $A$  is a nonempty set ( $u \in A$ ) and  $(\forall y : y \in A \rightarrow y \in x)$  then  $A$  has the same elements as  $\{y \in \tau(u) : u \in x\}$  and so

$$A = \{y \in \tau(u) : u \in x\} \in \tau^2(u).$$

Under these hypotheses  $x$  itself is a nonempty set containing  $u$  so  $x = \{y \in \tau(u) : y \in x\}$  by extensionality, and since  $x \in \tau(x)$  and

$$\{y \in \tau(u) : y \in x\} \in \tau^2(u)$$

we have  $\tau^2(u) = \tau(x)$ , so we have established  $A \in \tau(x)$  whence  $A \in \mathcal{P}(x)$ : any nonempty set all of whose elements are in  $x$  belongs to the same type as  $x$  and belongs to the power set of  $x$ . Clearly  $\emptyset_{\tau(x)}$  belongs to  $\mathcal{P}(x)$  and is the only elementless element of  $\mathcal{P}(x)$ .

Observe that any set  $y$  which contains  $x$  as an element is a subset of  $\tau(x)$  ( $y$  itself witnesses the fact that all its elements are of the same type as  $x$ ), and so an element of  $\mathcal{P}(\tau(x))$  and so an element of  $\tau^2(x)$ , so the collection of all sets which contain  $x$  is a subset of  $\tau^2(x)$  and an element of  $\tau^3(x)$ . Sethood of the collection  $B(x) = \{y : x \in y\}$  is not a feature of the usual set theory.

**Stratification:** We know now that if  $x \in y$  we have  $y \in \tau^2(x)$ . We would like to have a scheme of stratification, in which we have a fixed variable  $x$  and an assignment of natural numbers to all variables  $y$  such that  $y \in \tau^i(x)$ . Working upward, all is good: if  $y \in z$  and we know that  $y \in \tau^i(x)$ , we can conclude that  $z \in \tau^{i+1}(x)$ . Now suppose that we know that  $y \in z$  and that  $z \in \tau^{i+1}(x)$ . We know that  $z \in \tau^2(y)$  as well. We would like to conclude from  $\tau^2(y) = \tau^{i+1}(x)$  that  $\tau(y) = \tau^i(x)$ . We know that  $\tau(y) \in \tau^2(y) = \tau^{i+1}(x)$  and that  $\tau^i(x) \in \tau^{i+1}(x) = \tau^2(y)$ , so  $\tau(y) \sim \tau^i(x)$ , so  $\tau(y) \cup \tau^i(x)$  exists, so  $y \sim \tau^{i-1}(x)$  [here we allow the possibility that the superscript  $i - 1$  on  $\tau$  is 0, which we normally avoid], from which it follows that  $y \in \tau^i(x)$ . The axiom of binary union supports type inference downward.

If we have a formula  $\phi$  and a variable  $x$  which is connected to all variables in  $\phi$  in the obvious sense (by the transitive closure of the relation which obtains between variables occurring in the same atomic subformula of  $\phi$ ) then we can infer types of the form  $\tau^n(u)$  for every variable in  $\phi$  – we have  $u$  here instead of  $x$  because if we are computing types relative to  $x$  and encounter the subformula  $y \in x$ , we should change every type  $\tau^i(x)$  so far assigned to a variable to  $\tau^{i+1}(y)$ .

There will be more discussion of the uses of stratification. Note that it is possible that more than one type will be assigned to the same variable by the procedure we describe, which is not desirable.

## 4 The axiom of infinity and arithmetic

We give the development of arithmetic as a sample of mathematical work in this theory. We also need it for our pending finite axiomatization of the theory.

**Definition (finite sets):** We define  $\mathbb{F}_{\tau^2(x)}$ , the collection of all finite subsets of  $\tau(x)$ , as

$$\{F \in \tau^2(x) : (\forall I \in \tau^3(x) : (\emptyset_x \in I \wedge (\forall G \in I : (\forall y \in \tau(x) : G \cup \{y\} \in I))) \rightarrow F \in I)\}.$$

**Axiom of infinity:**  $(\forall x : \tau(x) \notin \mathbb{F}_{\tau^2(x)})$ : all types are infinite sets.

**The natural numbers and finite sets:** We define  $0_{\tau^2(x)}$  as  $\{\emptyset_{\tau(x)}\}$ , the set of all subsets of  $\tau(x)$  with zero elements.

We give a definition of an operation on all sets of sets which will be the successor operation when restricted to natural numbers. For any set  $A$  all of whose elements are sets, define  $\sigma(A)$  as

$$\{A \cup \{x\} : A \sim \tau(x) \wedge x \notin A\}.$$

Note that this will be a set by comprehension because any such  $A \cup \{x\}$  belongs to  $\tau^2(x)$ .

We define  $1_{\tau^2(x)} = \sigma(0_x)$ ,  $2_{\tau^2(x)} = \sigma(1_{\tau^2(x)})$ ,  $3_{\tau^2(x)} = \sigma(2_{\tau^2(x)})$  and so forth. Notice that for each concrete natural number  $n$ ,  $n_{\tau^2(x)}$  will be defined as the set of all subsets of  $\tau(x)$  with  $n$  elements. Notice also the at least apparent inconvenience that for each type  $\tau(x)$  we have different natural numbers for counting elements of  $\tau(x)$ , these natural numbers living in  $\tau^3(x)$ .

The set of natural numbers  $\mathbb{N}_{\tau^3(x)}$  is the set

$$\{n \in \tau^3(x) : (\forall I \in \tau^3(x) : 0_{\tau^2(x)} \in I \wedge (\forall k : k \in I \rightarrow \sigma(k) \in I) \rightarrow n \in I)\},$$

which belongs to  $\tau^4(x)$ .

We have an implementation of Peano arithmetic (or many implementations). We consider each of Peano's original axioms (in a form starting with 0 rather than 1).

1.  $0_{\tau^2(x)} \in \mathbb{N}_{\tau^3(x)}$  is evident.



2.  $n \in \mathbb{N}_{\tau^3(x)} \rightarrow \sigma(n) \in \mathbb{N}_{\tau^3(x)}$  is evident.
3.  $\sigma(n) \neq 0_{\tau^2(x)}$  is evident. Any element of  $\sigma(n)$  has an element.
4. The assertion  $(\forall mn \in \mathbb{N}_{\tau^3(x)} : \sigma(m) = \sigma(n) \rightarrow m = n)$  requires a little more attention.

We first use the axiom of infinity to argue that each natural number is nonempty. Notice that for any element  $A$  of  $\mathbb{F}_{\tau^2(x)}$  there is a set  $A \cup \{x\}$  with  $x \notin A$  which belongs to  $\mathbb{F}_{\tau^2(a)}$  [because  $A$  cannot be  $\tau(a)$ ]. so for any natural number  $n$ , if  $n$  contains an element of  $\mathbb{F}_{\tau^2(a)}$ ,  $\sigma(n)$  also contains an element of  $\mathbb{F}_{\tau^2(a)}$ .  $0_a$  obviously contains an element of  $\mathbb{F}_{\tau^2(a)}$ , namely  $\emptyset_{\tau(a)}$ . It follows that every natural number contains an element of  $\mathbb{F}_{\tau^2(a)}$ , and therefore that every natural number is nonempty.

We then argue that for any natural number  $n$ ,

$$n = \{A - \{x\} : x \in A \wedge A \in \sigma(n)\}.$$

This is clearly true of  $0_{\tau^2(a)}$ :  $1_{\tau^2(a)}$  is the set of all singletons in  $\tau^2(a)$ , and the set of sets obtained by removing one element from the singleton contains exactly the empty set in  $\tau^2(a)$ . Suppose it is true of  $m \in \mathbb{N}_{\tau^3(a)}$ : we argue that it is true of  $\sigma(m)$ . Any set in  $\sigma(m)$  is of the form  $A - \{x\}$  for some  $A$  in  $\sigma(\sigma(m))$  [this relies on the fact that we know that  $\sigma(\sigma(m))$  is inhabited]: if  $B \in \sigma(m)$  there is a  $B \cup \{x\} \in \sigma(\sigma(m))$  and  $(B \cup \{x\}) - \{x\} = B$ . Now consider any set of the form  $A - \{x\}$  where  $A \in \sigma(\sigma(m))$  and  $x \notin A$  is of appropriate type: we need to show that  $A - \{x\} \in \sigma(m)$ .  $A - \{x\} = (B \cup \{y\}) \cup \{z\}$  where  $B \in m$ . If  $x$  is either  $y$  or  $z$ , it can wlog be taken to be  $z$ , and  $A - \{z\} = B \cup \{y\} \in \sigma(m)$  follows. Otherwise, we have  $A - \{x\} = (B \cup \{y\}) - \{x\} \cup \{z\}$  and by hypothesis  $(B \cup \{y\}) - \{x\} \in m$ , so  $A - \{x\} = (B \cup \{y\}) - \{x\} \cup \{z\} \in \sigma(m)$ . This shows that we can exactly compute  $n$  from  $\sigma(n)$  for all  $n$ , which establishes the fourth Peano axiom.

5. The assertion that any set  $S$  which contains  $0_{\tau^2(x)}$  and contains  $\sigma(n)$  if it contains  $n$ , contains all elements of  $\mathbb{N}_{\tau^3(x)}$  follows immediately from the definition of  $\mathbb{N}_{\tau^3(x)}$ .

Note that  $\mathbb{F}_{\tau^2(x)}$  can be shown to be the set of all elements of  $\tau^2(x)$  which belong to some element of  $\mathbb{N}_{\tau^3(x)}$ .

We have given the original five axiom formulation of Peano arithmetic, and we haven't yet said anything about addition or multiplication.

For the moment, we supply some definitions. A more extensive treatment of arithmetic relations and operations may appear later.

**Definition (order on natural numbers):** For  $m, n \in \mathbb{N}_{\tau^3(x)}$ , we define  $m \leq n$  as holding iff there are  $A \in m$  and  $B \in n$  with  $A \subseteq B$ .

**Definition (addition of natural numbers):** If  $m, n \in \mathbb{N}_{\tau^3(x)}$ , we define  $m + n$  as  $\{A \cup B : A \in m \wedge B \in n \wedge A \cap B = \emptyset_{\tau^2(x)}\}$ .

**Definition (raising natural numbers in type):** If  $n \in \mathbb{N}_{\tau^3(x)}$  and  $A \in n$ ,  $T(n)$  is the natural number in  $\mathbb{N}_{\tau^4(x)}$  containing  $\iota''A$ .

We leave to the reader the exercise of showing that the natural numbers are closed under addition and the operation has expected properties.

We remark in closing on our subscripting convention. Notice that where appropriate we apply subscripts to concepts with the type we are working with as a parameter: the subscript is the type which contains the subscripted object as a subset (i.e., the type belonging to the same type as the subscripted object).

## 5 Relations and functions: more mathematical technology

**ordered pairs, relations and functions:** For any  $x \sim y$  we define the unordered pair  $\{x, y\}$  as  $\{z \in \tau(x) : z = x \vee z = y\}$ <sup>2</sup>. Notice that  $\{x, x\} = \{x\}$ . We can then define  $(x, y)$  as  $\{\{x\}, \{x, y\}\}$  as long as  $x \sim y$ . Notice that  $(x, y) \in \tau^3(x) = \tau^3(y)$ .

Further, if  $A, B \in \tau^2(x)$  we can define  $A \times B$  as

$$\{c \in \tau^4(x) : (\exists ab : c = (a, b) \wedge a \in A \wedge a \in B)\}.$$

---

<sup>2</sup>This entire essay could be recast as an essay on the enormous effects of the unrestricted axiom of pairing. Zermelo comments on the axiom of pairing as in effect a repudiation of typing in his original 1908 paper on the axioms of set theory. In Zermelo set theory, the fact that quantifiers over the entire universe can appear in instances of separation has enormous power; in this theory, the same logical feature gives no particular power, as in effect all quantifiers are bounded in a way deducible from the syntax.

We can then define relations with domain  $A$  and codomain  $B$  as subsets of  $A \times B$  as usual. If  $R \subseteq A \times B$ , we define  $x R y$  as  $(x, y) \in R$ . Notice that if  $A, B \in \tau^2(x)$ ,  $R \in \tau^5(x)$ . But we do need to note that we have binary relation symbols already in use which cannot be understood in this way.  $x \in y$  cannot be understood in this way because  $x \sim y$  cannot hold.  $x = y$  and  $x \subseteq y$  cannot be understood in this way because their use is not restricted to a single type. We could define relations  $=_x = \{(x, y) \in \tau^3(x) : x = y\}$  and  $\subseteq_x = \{(A, B) \in \tau^5(x) : A \subseteq B\}$ ; these are restrictions of our logical relations of equality and subset to fixed types.

Now we can define a function from  $A$  to  $B$  as a subset of  $A \times B$  with the property that for each  $a \in A$  there is exactly one  $b \in B$  such that  $(a, b) \in f$ : we say  $f : A \rightarrow B$  to mean that  $f$  is a function from  $A$  to  $B$ . For each  $a \in A$ , we define  $f(a)$  as the unique  $b$  such that  $(a, b) \in f$ . [Note that if  $x$  and  $f(x)$  live in  $\tau(u)$ ,  $f$  will live in  $\tau^4(u)$ .] And, again, we need to remember that we are already using function symbols which cannot be understood in this way: we make heavy use of function symbols which apply to objects of many types and may have outputs at a different type than their inputs.

**elementwise images:** If  $F$  is a function or an operation, we define  $F^{\ulcorner}A = \{x \in \tau^0(A) : (\exists y : x = F(y))\}$ .

**tuples of length  $n$ :** We define  $n$ -tuples for any natural number  $n$  (noting that our 2-tuples do not coincide with pairs). We define an  $n$ -tuple as a function  $t$  whose domain is the set of natural numbers  $m$  in  $\mathbb{N}_x$  with  $1 \leq m \leq n$  and whose range is a subset of  $\iota^2 \ulcorner \tau(x)$ : we write  $t_i = y$  when  $t(i) = \iota^2(y)$ . We write  $[t_1, \dots, t_n]$  as notation for an  $n$ -tuple.

The advantage of tuples defined in this way is that they have a fixed displacement in type from elements of  $\tau(x)$ : from the equation  $t(i) = \iota^2(y)$  we see that  $t \in \tau^6(x)$ , three types higher than  $\iota^2(y) \in \tau^3(x)$ . The details of this number do not matter: all we need to know is that there is such a uniform type displacement and it does not depend on the length of the tuple.

**concepts of degree  $n$ :** We use terminology taken from Quine for  $n$ -ary relations considered with a particular purpose in mind. We define  $D_x^n$  as the set of all  $n$ -tuples in  $\tau^6(x)$ , and say that a set  $C$  is a concept of

degree  $n$  precisely if  $C \subseteq D_x^n \cup \{T^3(n)\}$  for some  $n, x$  and  $T^3(n) \in C$ : the appearance of  $T^3(n)$  is to ensure that empty concepts of different degree are distinct.

**concatenation of  $n$ -tuples:** For  $t \in D_x^m$  and  $u \in D_x^n$ , we define  $t + u$  as the unique element  $v$  of  $D_x^{m+n}$  such that  $v_i = t_i$  for  $i$  in the domain of  $t$ , and  $v_{m+i} = u_i$  for  $i$  in the domain of  $u$ .

**basic operations on concepts:** If  $A \in D_x^m$  and  $B \in D_x^n$ , we define

1.  $A * B$  as  $\{t + u : t \in A \wedge u \in B\} \cup \{T^3(m+n)\}$  This will belong to  $D_x^{m+n}$ .
2.  $A/B$  (if  $n \leq m$ ) as  $\{t \in \tau^6(x) : (\exists u \in B : t+u \in A)\} \cup \{T^3(m-n)\}$ . This will belong to  $D_x^{m-n}$ .
3.  $\Delta A$  as  $\{t + t : t \in A\} \cup \{T^3(2m)\}$ . This will belong to  $D_x^{2m}$ .
4.  $A^c$  as  $\{t \in D_x^m : t \notin A\} \cup \{T^3(m)\}$ . This will belong to  $D_x^m$ .

**adjustment of the types of relations:** Some complications in our development could be simplified if we could define an ordered pair  $(x, y) \sim x$ .

And indeed this can be done for some types. We proceed to define what is known as the Quine ordered pair  $(a, b)_x$ . The need for the type locating parameter  $x$  will become clear.

Define  $\sigma_0(y)$  as  $\sigma(y)$  if  $y$  is a natural number, and as  $y$  otherwise. Note that this requires that  $y$  belong to a type  $\tau^3(x)$ . Define  $\sigma_1(z)$  as  $\{\sigma_0(y) \in \tau(z) : y \in z\}$  and  $\sigma_2(z)$  as  $\sigma_1(z) \cup \{0_x\}$ , where we require  $z \in \tau^4(x)$ . We then define  $(a, b)_x$  as  $\{\sigma_1(u) \in \tau(a) : u \in a\} \cup \{\sigma_1(u) \in \tau(b) : u \in b\}$ , where we must have  $a, b \in \tau^6(x)$ . This is demonstrably an ordered pair, and it is of the same type as its projections. The apparent disadvantage is that it is defined only in types which contain sufficiently iterated power sets of a type. We will subsequently explain why we consider this disadvantage only apparent.

An easy remark here is that if we assume that there are no individuals, then every type is of the form  $\tau^i(x)$  for each concrete  $i$ , and the Quine pair is universally definable.

**accommodation of heterogenously typed relations and functions:** Notice that this formal system does not accommodate representation of functions between distinct types. In certain cases this can be managed, and arguably these are the only important cases.

The general idea is that a relation from  $\tau^m(x)$  to  $\tau^n(x)$  can be coded as a relation from  $\iota^{n-\min(m,n)}\tau^m(x)$  to  $\iota^{m-\min(m,n)}\tau^n(x)$ : in general, elements of  $\tau(x)$  can be coded in  $\tau^{n+1}(x)$  by elements of  $\iota^n\tau(x)$ .

The underlying idea which makes this seem sufficient is that types really do not seem to have anything to do with each other unless one is an iterated image of the other under the  $\tau$  operation.

We do not make the presumption that Quine does that the types are exactly the ones of the simply typed theory of sets (we cannot even express it), but we do not explicitly provide any machinery that goes beyond this.

We note under this heading that one way the theory could be extended is with a primitive ordered pair with the following axioms:

**primitive notion:** For any  $x, y$  (with no presumption that they are of the same type) there is an object  $(x, y)$ .

**typing of pairs:** If  $x \sim y$  and  $z \sim w$ , then  $(x, z) \sim (y, w)$ .

**basic property of pairs:** If  $x = y$  and  $z = w$ , then  $(x, z) = (y, w)$ .

We note the possibility of extending the theory in this way. This allows general definitions of relation types and allows simulation of the more complicated type theory of Russell and Whitehead. It also increases the ability of the theory to introspect on its own structure considerably, and we are not inclined to adopt it. We may occasionally have reason to refer to this extension.

## 6 Our theory and a general type theory

The usual simple typed theory of sets (which we call TST) is a multisorted theory with types indexed by the natural numbers. It has equality and membership as primitive predicates.

We handle the typing by providing a function from variables to natural numbers under which each natural number has a countable inverse image. A

variable  $x$  then has sort  $\mathbf{type}(x)$ . One could also include numerical indices as typographical components of the variables:  $\mathbf{type}(x)$  could be a feature of  $x$ . We do not decorate our variables with superscripts, but if a variable has a superscript we do follow the convention  $\mathbf{type}(x^i) = i$ .

Atomic subformulas  $x = y$  are well-formed iff  $\mathbf{type}(x) = \mathbf{type}(y)$ . Atomic subformulas  $x \in y$  are well-formed iff  $\mathbf{type}(x) + 1 = \mathbf{type}(y)$ .

The axioms we adopt for TST are

**(weak) extensionality:** All well typed formulas of the shape  $(\forall xyz : z \in x \rightarrow (x = y \leftrightarrow (\forall w : w \in x \leftrightarrow w \in y)))$ . It is usual to strengthen this so that any object, not just nonempty objects, has identity determined by its extension, but we keep it parallel to our base theory. We do have reasons for preferring weak extensionality.

**comprehension:** For any formula  $\phi$  and variable  $A$  not appearing in  $\phi$ ,  $(\exists A : (\forall x : x \in A \leftrightarrow \phi))$  is an axiom if it is well-formed. Notice that for example  $(\exists A : (\forall x : x \in A \leftrightarrow x \notin x))$  is not an axiom because  $x \in x$  is not a well-formed atomic subformula, as  $\mathbf{type}(x)$  cannot be  $\mathbf{type}(x) + 1$ .

Our theory TTGV can in general terms interpret TST using some  $\tau(x)$  as type 0, and  $\tau^{i+1}(x)$  as type  $i$  (replacing type  $j$  variables with variables restricted to  $\tau^{j+1}(x)$ ). However, the language of our theory affords no obvious way to describe the sequence of  $\tau^i(x)$ 's (and in fact provably cannot describe this sequence).

To get a type theory which our theory can interpret nicely, we introduce a more general type system. We have a system of notations for types. There is a supply of atomic notations for types, some of which are specified as types of individuals. For any type notation,  $\tau^+$  is a type notation. For any type notation not reducing to atomic notation for a type of individuals,  $\tau^-$  is a type notation. A notation  $(u^-)^+$  or  $(u^+)^-$  reduces to  $u$ , where  $u$  is any type notation, and these are the only reduction rules. Now we provide a map from variables to reduced type notations, and stipulate that  $x = y$  is well-formed iff  $\mathbf{type}(x) = \mathbf{type}(y)$  and  $x \in y$  is well-formed iff either  $(\mathbf{type}(x))^+ = \mathbf{type}(y)$  or  $(\mathbf{type}(y))^- = \mathbf{type}(x)$ .

We call this theory TSTG. Its axioms are exactly as above mod the more liberal type system.

Now we can describe a putative interpretation of the language of TSTG in our untyped theory TTGV. Given a model of TTGV, we provide an atomic

type notation  $\chi(x)$  correlated with each type  $\tau(x)$  satisfying  $\chi(x)^+ = \chi(\tau(x))$  and  $\chi(x)^- = \chi(\tau^{-1}(x))$  if this is defined. Recall that  $\tau^{-1}(x)$  is clearly the reduction rules are valid. We then stipulate that any free variable of type  $u$  is to represent an element of the  $\tau(x)$  associated with  $u$ , and any quantifier binding a variable of type  $u$  is to be replaced with a (no longer typed) variable restricted to the associated  $\tau(x)$ .

To promote it from putative interpretation to actual interpretation, we need to verify that the interpretations of the axioms of the type theory hold. The axiom of weak extensionality of the type theory definitely holds because of the axiom of weak extensionality of our theory.

The axiom of comprehension holds similarly obviously: the set  $\{y^u : \phi\}$  of the type theory is implemented by  $\{y \in \tau(x) : \phi^*\}$  (where  $\phi^*$  is the translation of  $\phi$  via the interpretation) where  $\tau(x)$  is associated with the type  $u$ , and notice that this belongs to  $\tau^2(x)$ , which is associated with the type  $u^+$  as required.

That was unexciting and direct. The excitement comes with the converse interpretation of our theory in the language of the typed theory.

For purposes of this interpretation, we extend the language of the typed theory to an unsorted language, providing that for  $x$  and  $y$  of different types,  $x = y$  is false, and for  $x$  and  $y$  of types such that  $x \in y$  is ill-formed as a formula of the language of the type theory,  $x \in y$  is false.

We then claim that this extension of atomic subformulas makes a model of the typed theory (with disjoint domains for the types) a model of our theory with a single domain which is the union of all the types. Note that any model of the typed theory can be converted to one in which all the domains are disjoint by replacing each domain  $D_u$  interpreting type  $u$  with  $D_u \times \{u\}$  with the obvious induced redefinition of equality and membership in the model.

Verification of the axioms is straightforward, with the sole exception of the scheme of comprehension. The difficulty is that the comprehension scheme of our theory allows the construction of sets  $\{y \in \tau(u) : \phi\}$  where  $\phi$  may not be in any obvious way the translation of a formula of the typed theory.

We describe a procedure for converting formulas of the language of TTGV into equivalent formulas which are typable (at least to a great extent). We do not for the moment consider formulas in which  $<_\tau$  occurs. We regard a formula  $\phi$  as *nearly typable* if each variable  $u$  in  $\phi$  is free (and so can be assigned type  $\tau(u)$ ) or bound in a subformula  $(\forall u \in \tau(b) : \psi)$ , where  $u$  does not occur in  $b$ , in which case it can be assigned type  $\tau(b)$ , or (the unstratified

exception) ranges over all types other than a concrete finite list of exceptions in a formula in which the only free variables are the ones determining the exceptional types. We explain how to eliminate the unbounded quantifier from  $(\forall x : \phi)$  where  $\phi$  is a nearly typable formula. We first reconstrue this as  $(\forall \tau(b) : (\forall x \in \tau(b) : \phi))$ . We then consider that there are a finite collection of special types to consider: if a variable  $u$  occurs in an atomic subformula with a variable  $v$  assigned type  $\tau^i(x)$ , there is a specific type we have to consider applying to  $x$ : if  $u$  is assigned  $\tau(a)$  and the atomic subformula is  $v = u$  then  $x$  “should” be assigned  $\tau^{1-i}(a)$ ; if the atomic subformula is  $u \in v$ ,  $x$  “should” be assigned type  $\tau^{2-i}(a)$ ; if the atomic subformula is  $v \in u$ ,  $x$  “should” be assigned type  $\tau^{-i}(a)$ . Let  $\{\tau(a_i)\}$  be the finite list of types which should be assigned to  $x$  in this sense, other than types  $\tau^i(x)$ . We can further reconstrue the formula as

$$(\forall x_1 \in \tau(a_1) : \phi[x_1/x]) \wedge \dots \wedge (\forall x_n \in \tau(a_n) : \phi[x_n/x]) \\ \wedge (\forall \tau(b) : (\tau(b) \neq \tau(a_1) \wedge \dots \wedge \tau(b) \neq \tau(a_n) \rightarrow (\forall x \in \tau(b) : \phi)))$$

The understanding here is that the variables  $x_i$  are fresh variables.

Note further that in this formula we can eliminate all atomic subformulas  $u = v$  in which  $u$  and  $v$  are assigned types which cannot be equal, and all atomic subformulas  $u \in v$  in which the type assigned to  $v$  cannot be the image under  $\tau$  of the type assigned to  $u$  (by replacing them with the False): notice that all occurrences of  $x$  in the final conjunct will be eliminable in this way, except occurrences in bounds  $\tau^i(x)$  and ones occurring with variables actually assigned type  $\tau^0(x)$ ,  $\tau(x)$ , or  $\tau^2(x)$  appropriately, because all types other than  $\tau(x)$  deducible for occurrences of  $x$  by stratification are eliminated: in this formula, as transformed, nothing assigned a type  $\tau^i(x)$  will occur with anything not assigned a type of the form  $\tau^j(x)$  so  $x$  will not be connected to any free variable. Logical manipulations allow us to move all variables other than those with types  $\tau^i(x)$  out of the scope of the quantifier, so in fact the final conjunct can be arranged to have no free variables but the ones determining the exceptional types [note that variables not connected to  $x$  will not be connected to the  $a_i$ ’s either]. We note further that subformulas  $\tau(b) \neq \tau(c)$  which restrict variables  $b$  ranging over all types are transformed as well: if  $c$  is assigned type  $\tau(d)$  (which would happen if we were reducing an unbounded quantifier over  $c$ ) then  $\tau(c)$  is simply replaced with  $\tau(d)$ . [If formulas  $x <_\tau y$  are supposed present, these can be assigned truth values if  $x$  and  $y$  are connected: there is quite possibly a general way of handling



this by extending type assignments so that truth values of such formulas are decided and they are eliminated.]

Existential quantifiers are handled similarly using existential quantification and disjunction.

We say that variables are connected if they stand to each other in the relation determined as the transitive closure of the relation of being in the same atomic subformula, other than with a variable ranging over types. Notice that when a formula is completely transformed using the procedure described, each pair of connected bound variables  $x, y$  assigned types  $\tau(a), \tau(b)$  must have  $\tau(b) = \tau^i(a)$  for some integer  $i$ . Comprehension for such formulas follows from stratified comprehension: to show that  $\{x : \phi\}$  exists using stratified comprehension it is sufficient to be able to type all bound variables connected to  $x$ . Stratification of all free variables can be forced by making them all different, and exploiting the fact that free variables are implicitly universally quantified. Failures of stratification for variables not connected to  $x$  can be handled by extracting all variables connected to  $x$  from the scope of quantifiers over such variables; unstratified closed formulas [which will have quantifiers over all but finitely many types] do not present a problem for comprehension (they can be eliminated by case expansion, what happens when the formula is true and what happens when it is false).

We would love to claim that all formulas are equivalent to stratified formulas, but the final conjuncts in the expansions of unbounded quantifiers are not necessarily stratified. They *would* be stratified if every assertion about types which is true of all but finitely many types is in fact true of all types. This condition could be arranged by adding an ambiguity axiom [of which more later]; it can be enforced by a much simpler manipulation: take the model of TSTG which we start, make countably many disjoint copies, and then assign truth values of false to atomic subformulas which are badly typed. In this structure, any typable assertion true of all types but a concrete finite list of types is in fact true of all types, and the restrictions of final conjuncts to all but finitely many types can be dropped, so that we simply get stratified formulas:  $(\forall \tau(b) : (\tau(b) \neq \tau(a_1) \wedge \dots \tau(b) \neq \tau(a_n) \rightarrow (\forall x \in \tau(b) : \phi)))$  is equivalent to  $(\forall \tau(b) : (\forall x \in \tau(b) : \phi))$ . So there is a way to immerse any model of TSTG in a model of TTGV in which every formula is equivalent to a stratified formula, using the scheme that asserts that  $\tau^n(x) \neq \tau(x)$  for  $n > 1$  and a scheme asserting that  $(\forall \tau(b) : (\tau(b) \neq \tau(a_1) \wedge \dots \tau(b) \neq \tau(a_n) \rightarrow (\forall x \in \tau(b) : \phi)))$  is equivalent to  $(\forall \tau(b) : (\forall x \in \tau(b) : \phi))$  [where  $\phi$  is a stratified closed formula].

## 7 Finite axiomatization of the theory

Stratified comprehension is finitely axiomatizable: we will insert a proof of this here using the calculus of concepts machinery, just for fun.

The transformation above shows (with some thought) that the untyped comprehension scheme is fully axiomatized by the finite axiomatization of stratified comprehension plus the scheme asserting that  $\tau^i(x) \neq \tau(x)$  for  $i > 1$  (needed for the elimination of bad atomic formulas). The axioms of type order prove all instances of this scheme, so our theory with the (novel) axioms of type order is finitely axiomatizable.

Here is the finite axiomatizability argument. Suppose that TTGV has a finite list of axioms. Each of these axioms is a theorem of our formulation of TTGV so has a proof using stratified comprehension (finitely axiomatized) and finitely many instances of the scheme  $\tau^i(x) \neq \tau(x)[i > 1]$ . Thus all of TTGV can be axiomatized using stratified comprehension and finitely many instances of the type disjointness scheme. But this is impossible, because NFU is consistent, so cyclic NFU of order  $n$  is consistent (in which types  $0 - (n-1)$  are distinct and type  $n$  is type 0, and cyclic order NFU extended to assign false to unstratified atomic formulas satisfies stratified comprehension and as many axioms in the type disjointness scheme as desired (by taking  $n$  large enough) so it would be possible to prove type  $n$  distinct from type 0 in extended cyclic NFU of order  $n$  for large enough  $n$ , which is impossible. If we strength extensionality to assert that all atoms are individuals, the same result can be shown in the same way if NF is consistent.

It is interesting to observe that the axioms of type order do absolutely nothing. In any model of TTGV without the axioms of type order, the relation which holds between any  $\tau(x)$  and any  $\tau^i(x)$  with  $i > 1$  and otherwise does not hold, satisfies the axioms of type order, without any change in the capabilities of the theory at all. Note that any atomic subformula in  $<_\tau$  can be eliminated in favor of a truth value by the stratification procedure above as soon as the two variables involved are assigned connected types.

But, curiously, as seen above, TTGV without the axioms of type order is not finitely axiomatizable.

## 8 Quine's original axioms

Quine handles this somewhat differently, and here I believe I have made a significant formal improvement. Quine schematically defines the types of the usual type theory.

**definition of “being of the previous type”:**  $x \text{ PT } y$  is defined as

$$(\exists zw : x \in w \wedge w \in z \wedge y \in z).$$

**definition of type 0:**  $T_0(x)$  is defined as  $(\forall y : \neg y \text{ PT } x)$ .

**definition of next type:** For each concrete natural number  $n$ ,  $T_{n+1}(x)$  is defined as  $(\forall y : T_n(y) \rightarrow y \text{ PT } x)$

He then stated his axioms schematically.

**Quine's comprehension axiom:** For any formula  $\phi$ ,

$$(\exists A : T_{n+1}(A) \wedge (x \in A \leftrightarrow (T_n(x) \wedge \phi))).$$

**Quine's extensionality axiom:**

$$(\forall xyz : T_{n+1}(x) \wedge T_{n+1}(y) \wedge (\forall w : T_n(w) \rightarrow (w \in x \leftrightarrow w \in y)) \wedge x \in z \rightarrow y \in z).$$

I preserve the form of this axiom, which reflects defining equality in terms of membership, but it could be phrased differently.

These are actually not the axioms as he first states them: this is the original extensionality axiom together with a modified version of the comprehension axiom which he states later as a consequence of the assumption that all elements of type  $n + 1$  objects belong to type  $n$ , which his original axioms (astonishingly) do not imply.

This theory is not quite the same as ours. To begin with, it has what we regard as a formal defect: there is no need to axiomatize the theory with schemata with concrete natural numbers as indices, as we have demonstrated with our axiomatization. Quine does observe that he cannot prove and cannot even actually say that every object belongs to some type. Further, his theory says nothing at all about objects which do not belong to a type. In

our theory, it is immediate that every object belongs to a type, but the types may not be restricted to the familiar ones.

Quine says more about individuals than we do. Quine asserts that all individuals belong to the same type. We have not felt the need to do this, but we could. We are more tempted to assert that there are no individuals at all.

We think that our presentation is superior to Quine's for a number of reasons. Our presentation is finitely axiomatizable, and does not allude to the simple typed theory of sets at all in its formulation [or much less obviously]: the fact that it is actually a presentation of the simple typed theory of sets unfolds in the development. We dispute something that Quine says: there is a strong place for systematic ambiguity in this theory; we are not through with this device when we transition to a one-sorted theory. But this also comes out in the development.

The axioms as selected above from Quine's treatment allow us to prove that all elements of a type  $n + 1$  object are of type  $n$ : for any  $x$  of type  $n + 1$  there is  $x^*$  of type  $n + 1$  containing exactly the type  $n$  elements of  $x$ , and then by his original formulation of extensionality,  $x^* = x$ , so in fact all elements of  $x$  are of type  $n$ . In our formulation, as will be seen, the axiom of binary union is used to prove the analogous assertion.

Finally, our theory differs from Quine's quite deliberately in allowing atoms as well as empty sets.

## 9 The system of Resnik

What Quine did was a kludge. The presence of meta theoretic natural number parameters corresponding exactly to the types reveals that he is not really describing an autonomously motivated system.

Resnik gives a genuine one-sorted theory with one-sorted motivation from which type theory falls out, as we do.

We list his seven axioms, staying closer to our own notation.

**Definition:**  $x \sim y$  means  $(\exists z : x \in z \wedge y \in z)$ . Resnik defines  $x = y$  as  $(\forall z : x \in z \leftrightarrow y \in z)$ . So does Quine; for us equality is a logical primitive, but the comprehension axiom of any of these theories should make this definition harmless.

**Ax 1:**  $(\forall x : (\exists y : (\forall z : z \in y \leftrightarrow z \sim x)))$ . This is almost the same as our axiom of types: ours has the extra clause  $x \in y$  to ensure that  $\sim$  is reflexive. Strangely, the axiom of comprehension has to be used to fill in this detail in Resnik's system.

**Ax 2:**  $(\forall xyw : y \in x \wedge y \in w \rightarrow x \sim w)$ . Sets which meet have the same type. Our argument for this depends strongly on our form of extensionality, which asserts that nonempty sets with the same extension are equal regardless of type. I'm wondering whether this is provable from Resnik's other axioms.

**Ax 3:**  $(\forall uvwx : y \in x \wedge u \in x \wedge y \in w \wedge v \in w \rightarrow (\exists t : y \in t \wedge u \in t \wedge v \in t))$ . This axiom is used to support transitivity of  $\sim$ . I believe that it is redundant. If  $y \in x \wedge u \in x$  then we have  $y \in \tau(u)$ , where  $\tau(u)$  witnesses Ax 1 with  $x := u$ . Similarly we have  $v \in \tau(u)$ .  $u \in \tau(u)$  is not a consequence of Ax 1 (as it is in our formulation) but it does hold here because  $u$  belongs to some set by the hypotheses. So we can choose  $\tau(u)$  as  $t$ .

**Ax 4:**  $(\forall vwxy : y \in x \wedge v \in w \wedge x \in z \wedge w \in z \rightarrow y \sim v)$  This does what the axiom of binary union does for us. We say that because  $x$  and  $w$  have the same type, they have a union, and of course this union will contain  $y$  and  $v$ . We have considered this exact statement as an axiom, but union seemed more clearly a natural axiom.

**Definition:**  $xPTy$  is defined (following Quine) as  $(\exists zw : x \in w \wedge w \in z \wedge y \in z)$ .  $T_0(x)$  ( $x$  is an individual) is defined as meaning  $\neg(\exists y : yPTx)$ : nothing belonging to the same type as  $x$  has elements.

**Ax 5:**  $(\exists x : T_0(x))$  We do not commit ourselves to the existence of any individuals. But it is natural to do so given the historical origin of this theory.

**Ax 6:**  $(\neg T_0(x) \wedge x \sim y \wedge (\forall z : z \in x \leftrightarrow z \in y) \wedge x \in w) \rightarrow y \in w$ . The form of this looks peculiar to us because Resnik treats equality as a defined notion, but it is the axiom of extensionality. It is a bit different from ours: it is weaker in that it does not force equality of nonempty sets with the same extension (we do not need Ax 2 because our extensionality axiom flatly asserts that nonempty sets with the same extension are equal and so of course of the same type); it allows individuals with the

same empty extension to be distinct but any empty object in a type is the only empty object in that type. This is natural; we are more liberal in allowing many atoms in each type.

**Ax 7:** For any formula  $\phi$ ,  $(\forall z : (\exists y : w \sim z \wedge (\forall x : x \in y \leftrightarrow (\phi \wedge x \in z))))$ . This is Zermelo's axiom scheme of separation, with the extra proviso that the set defined is of the same type as the bounding set. We could have done this, and of course it is an immediate consequence of our approach, in which the bounding set is always a type.

The maneuver for showing that a general object belongs to a set is rather strange here, and I want to be sure that Resnik actually realizes that he has to do it. For an arbitrary  $x$ , there is  $w \sim x$  with empty extension, by axiom 7...and incidentally, some set contains both  $x$  and  $w$ , so  $x$  belongs to a set.

That said, this theory is the same as ours with stronger extensionality and the positive assertion that there are individuals. I think that my axiomatics are cleaner, and that there are really good reasons to consider the possibility of atoms in addition to empty sets.

There is an error in Resnik. He claims that he can prove that all individuals are of the same type. This does not follow from his axioms. I think part of the issue is that he defines ST in two different ways and does not seem to realize that they are not equivalent.