The Multitudes Metalanguage and Theory

Multitudes

`M`

This document discusses and axiomatizes the Multitudes metalanguage and theory in the IFF. The theory of multitudes was presented in the paper

"Sets and Classes as Many", by John L. Bell. In *Journal of Philosophical Logic*, Vol. 29, 2000.

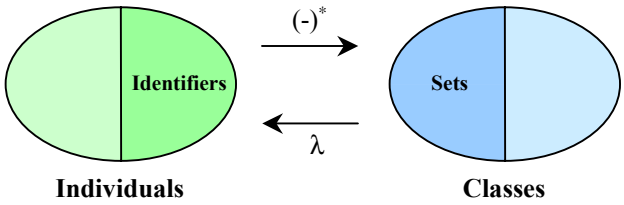

Figure 1: Architecture

As indicated in Table 1, some of the basic aspect of Multitudes (**`M`**) corresponds to the IFF-KIF metalanguage. The remainder of the basic aspect and more the specialized aspects of Multitudes, such as classes (**`M.C`**) and individuals (**`M.I`**), are explicitly represented in this document as the terminology in an IFF meta-ontology (IFF-M). The fundamental difference between the Multitudes metalanguage and the IFF-KIF metalanguage is that Multitudes represents two set-theoretic metalevels (set and class), whereas the IFF-KIF represents three metalevels (set, class and collection).

Table 1: Metalanguage Analogs

Multitudes Language $\mathcal{L}_M$	IFF-KIF and IFF-M
membership for classes '∈' and individuals 'ε'	'`(constant term*)`' IFF-KIF predicate notation
class collection	'`SET:class`' IFF-UCO collection
individual collection	generic IFF-KIF object
set *S* and identifier *I* collections	'`set.obj:set`' IFF-LCO class
pair operator '(x, y)'	'`[term1 term2]`' IFF-KIF pair notation
equality '=' for classes and individuals	'`(= term1 term2)`' IFF-KIF equality notation
—	'`collection-of`' IFF-KIF collection formation operator
class formation operator '$\{x \mid \varphi(x)\}$'	'`class-of`' IFF-KIF class formation operator
set formation operator	'`set-of`' IFF-KIF set formation operator
—	'`relation-of`' IFF-KIF relation formation operator
label λ and colabel ∗ functions	—

Following Bell (op. cit.), the theory **M** of *multitudes* or *classes-as-many* is formalized here in a two-sorted first order logic language $\mathcal{L}_{\mathbf{M}}$, which extends the IFF-KIF language with terms for the non-logical symbols $\in$, I, S, λ and $*$ as discussed below. In more detail, the metalanguage $\mathcal{L}_{\mathbf{M}}$ and theory (both collectively called Multitudes) are represented in IFF-KIF and IFF-M. The IFF representation of Multitudes has

- the usual logical symbols, connectives ($\neg$, $\wedge$, $\vee$, $\rightarrow$) and quantifiers ($\forall$, $\exists$), as represented in IFF-KIF,
- the abbreviations of IFF-KIF
  - restricted quantification,
  - **the** definite description, and
  - the collection/class/set/relation formation operators,
- two sorts, *class* and *individual*, represented as unary relations,
- two sub-sorts, *set S* and *identifier I*, also represented as unary relations,
- two unary functions, *label* 'λ' and *colabel* '$*$',
- two binary relations, *member* '$\in$' and member-individual 'ε',
- two equalities '=' on each sort,
- two meta-operations,
  - class formation from properties to classes
  - set-formation from properties to individuals
- three axioms schemes, *comprehension*, *separation* and *replacement*.

Table 2 lists the basic terms in the IFF-M meta-ontology.

Table 2: Terminology for the Objective Aspect of the Multitudes (meta) Ontology

	Other	Object	Morphism
M *basics*	member equal-classes class-of	class set	label
	member-individual equal-individuals label-of	individual identifier	colabel
M.C *class aspect*	subclass	full empty	singleton binary-union binary-intersection
		identifier	difference complement
		relation endorelation function	pair binary-product first second underlying-class source target field segment
M.I *individual aspect*	subindividual	empty	singleton binary-union binary-intersection
			union power

The multitudes language $\mathcal{L}_{\mathbf{M}}$ can be thought of as IFF-KIF augmented with the two sorts (collections) of individuals and classes, the two subcollections of identifiers and sets, the binary membership relation, and the label and colabel functions.

There are two distinct sorts of entity, individuals *ind* and classes (or multitudes) *cls*. These are represented as disjoint IFF collections $\mathit{ind} \cap \mathit{cls} = \varnothing$.

```
(1) (KIF:collection individual)

(2) (KIF:collection class)

(3) (KIF:disjoint individual class)
```

There is a binary *member*ship relation $\mathit{mbr} = \in \subseteq \mathit{ind} \times \mathit{cls}$ between individuals and classes. This is not exactly like the usual membership relation, since it is two-sorted.

```
(4) (KIF:relation member)
    (= (KIF:collection1 member) individual)
    (= (KIF:collection2 member) class)
```

An individual *property* in $\mathcal{L}_{\mathbf{M}}$ is any formula $\varphi(x)$ involving one free individual variable $x \in \mathit{ind}$ called the *varying parameter*. The formula φ may contain other free individual or class variables called the *fixed parameters*. Each property $\varphi(x)$ in $\mathcal{L}_{\mathbf{M}}$ has an associated multitudes class $\{x \mid \varphi(x)\} \in \mathit{cls}$ called the class defined by φ or the *class-of* individuals satisfying φ. This class is represented by the gamma expression '`(γ (?x) φ)`'. The γ operator, which is represented in $\mathcal{L}_{\mathbf{M}}$ by the term '`class-of`', is like a lambda operator; but instead of associating a relation with a formula, it associates a class with a property. The γ operator is also like a quantifier in that it binds an individual variable (its varying parameter). Gamma expressions are recursively defined, since the property φ may itself contain other gamma expressions used in place of classes. Gamma expressions are typically used in definitions, which involve classes. As a very simple example, the gamma expression '`(class-of (x) (colabel ?x))`' is the multitudes class associated with the multitudes property '`(colabel ?x)`'. It has one varying parameter 'x', but no fixed parameters. Of course, this is the same as the class '`(colabel ?x)`'. In general, classes defined by properties are governed by the **axiom of comprehension**, which states

$$y \in \{x \mid \varphi(x)\} \text{ iff } \varphi(y)$$

for all individuals $y \in \mathit{ind}$. Hence, the property φ and the class $\{x \mid \varphi(x)\}$ are in this sense equivalent.

There is an *equality* relation defined on each sort, individual and class. These are equivalence relations – reflexive, symmetric and transitive. These equalities enforce the sorts.

```
(5) (KIF:relation equal-classes)
    (= (KIF:collection1 equal-classes) class)
    (= (KIF:collection2 equal-classes) class)
    (forall (?X (class ?X))
        (equal-classes ?X ?X))
    (forall (?X1 (class ?X1) ?X2 (class ?X2))
        (=> (equal-classes ?X1 ?X2)
            (equal-classes ?X2 ?X1)))
    (forall (?X1 (class ?X1) ?X2 (class ?X2) ?X3 (class ?X3))
        (=> (and (equal-classes ?X1 ?X2) (equal-classes ?X2 ?X3))
            (equal-classes ?X1 ?X3)))

(6) (KIF:relation equal-individuals)
    (= (KIF:collection1 equal-individuals) individual)
    (= (KIF:collection2 equal-individuals) individual)
    (forall (?x (individual ?x))
        (equal-individuals ?x ?x))
    (forall (?x1 (individual ?x1) ?x2 (individual ?x2))
        (=> (equal-individuals ?x1 ?x2)
            (equal-individuals ?x2 ?x1)))
    (forall (?x1 (individual ?x1) ?x2 (individual ?x2) ?x3 (individual ?x3))
        (=> (and (equal-individuals ?x1 ?x2) (equal-individuals ?x2 ?x3))
            (equal-individuals ?x1 ?x3)))
```

Equality of classes is governed by the **axiom of extensionality**:

$$\forall X_1\ \forall X_2\ (X_1 = X_2 \leftrightarrow \forall x\ (x \in X_1 \leftrightarrow x \in X_2)).$$

This expresses the idea that classes are uniquely determined by their elements.

```
(7) (forall (?X1 (class ?X1) ?X2 (class ?X2))
        (<=> (equal-classes ?X1 ?X2)
             (forall (?x (individual ?x))
                 (<=> (member ?x ?X1) (member ?x ?X2)))))
```

We require a set to be a class of some kind. Following Bell (op. cit.), to regard the class concept as "class as many" or "class of" requires that sets not be regarded as individuals. Instead, in this ontology (Figure 1) sets are classes, which are represented or labeled by individuals. For simplicity, we assume that labels are attached, not only to sets, but also to all classes. Hence, assigned to each class $X \in \mathit{cls}$ is an individual $\lambda(X) = \lambda X$ called its label. Hence dually, there is a reverse process called colabeling. Hence, assigned to an individual $x \in \mathit{ind}$ is a class $(x)^* = x^*$ called its colabel. Thus, there are *label* and *colabel* functions

$\mathit{lbl} = \lambda : \mathit{cls} \to \mathit{ind}$

$\mathit{colbl} = (\text{-})^* : \mathit{ind} \to \mathit{cls}$

Since there are individuals that do not label any class, the colabel function is a partial function whose domain is the range of the labeling function.

```
(8) (KIF:function label)
    (= (KIF:source label) class)
    (= (KIF:target label) individual)

(9) (KIF:partial-function colabel)
    (= (KIF:source colabel) individual)
    (= (KIF:domain colabel) (KIF:range label))
    (= (KIF:target colabel) class)
```

By Cantor's theorem, the number of classes of individuals is larger than the number of individuals that are labels; meaning that labeling function cannot be an injection. Because of this, we define a subcollection S of the collection of classes on which the labeling map is injective. Classes in S are called sets. There is a pair of predicates (unary relations or IFF collections), denoted by $I = \mathit{idn} \subset \mathit{ind}$ and $S = \mathit{set} \subset \mathit{cls}$, and defined on individuals and classes, which represent *identifiers* and *sets*, respectively.

```
(10) (KIF:collection identifier)
     (= identifier (KIF:range label))
     (KIF:subcollection identifier individual)

(11) (KIF:collection set)
     (KIF:subcollection set class)
```

We assume that each set is the colabel of an individual, and that this individual is the label of that set: $X = \lambda(X)^*$ for each set $X \in \mathit{set}$. We also assume: $x = \lambda(x^*)$ for each identifier $x \in \mathit{idn}$. In other words, the restriction of the colabeling map to identifiers is the inverse of the restriction of the labeling map to sets. Hence, the notions of labeling, colabeling, identifiers and sets are subject to the **labeling axioms**.

$S(X) = I(\lambda X)$, $I(x) = S(x^*)$, $S(X) \to (\lambda X)^* = S$ and $I(x) \to \lambda(x^*) = x$.

These axioms assert that the labeling and colabeling functions restrict to the set and identifier collections, and that these functions are inverse to each other on these restrictions.

```
(12) (forall (?X (class ?X))
         (=> (set ?X)
             (and (identifier (label ?X))
                  (equal-classes ?X (colabel (label ?X))))))

(13) (forall (?x (individual ?x))
         (=> (identifier ?x)
             (and (set (colabel ?x))
                  (equal-individuals ?x (label (colabel ?x))))))
```

The following are theorems expanding on the inverse property between restricted labeling and colabeling.

```
(14) (forall (?X (class ?X) ?Y (class ?Y))
         (=> (and (set ?X) (set ?Y) (equal-individuals (label ?X) (label ?Y)))
             (equal-classes ?X ?Y)))

(15) (forall (?x (individual ?x) ?y (individual ?y))
         (=> (and (identifier ?x) (identifier ?y) (equal-classes (colabel ?x) (colabel ?y)))
             (equal-individuals ?x ?y)))
```

For each class $X \in \mathit{cls}$ and each property $\varphi(x)$ in $\mathcal{L}_{\mathbf{M}}$, the following two classes are equal: $\{x \mid \varphi(x)\} \in \mathit{cls}$ and $\{x \in X \mid \varphi(x)\} \in \mathit{cls}$.

```
(16) (forall (?p (property ?p)
                 ?X (class ?X))
         (equal-classes
             (class-of (?x (member ?x ?X)) (?p ?x))
             (class-of (?x) (and (member ?x ?X) (?p ?x)))))
```

There is a gamma operator on individuals. Each property $\varphi(x)$ in $\mathcal{L}_{\mathbf{M}}$ defines the multitudes individual $[[x \mid \varphi(x)]] = \lambda\{x \mid \varphi(x)\} \in \mathit{ind}$. The individual γ operator is represented in $\mathcal{L}_{\mathbf{M}}$ by the term '`label-of`'.

To paraphrase Bell, the membership relation $\mathit{mbr} = \in \subseteq \mathit{ind} \times \mathit{cls}$ between individuals and classes is a primitive of the IFF multitudes (meta) ontology (IFF-M). The fact that "$\in$" is not iterable – there are no "$\in$-chains" – allows this membership to have few intrinsic properties. This is in contrast to the binary *membership* relation $\varepsilon \subseteq \mathit{ind} \times \mathit{ind}$ on individuals define by: $x \varepsilon y$ iff $x \in y^*$ for all $x, y \in \mathit{ind}$. This is the usual membership relation, and it is one-sorted. Since it links entities of the same sort, it is iterable.

```
(17) (KIF:relation member-individual)
     (= (KIF:collection1 member-individual) individual)
     (= (KIF:collection2 member-individual) individual)
     (forall (?x (individual ?x) ?y (individual ?y))
         (<=> (member-individual ?x ?y)
              (member ?x (colabel ?y))))
```

The defined relation ε is the membership relation on individuals. The implication

$$I(x) \to x = [[y \mid y \varepsilon x]]$$

follows from the comprehension and labeling axioms.

```
(18) (forall (?x (individual ?x))
         (=> (identifier ?x)
             (equal-individuals ?x (label-of (?y) (member-individual ?y ?x)))))
```

The Class Aspect

M.C

Boolean Operations on Classes

There is a *full* class of individuals $V = \{x \mid x =_{ind} x\} \in \mathit{cls}$.

```
(19) (M:class full)
     (= full
        (M:class-of (?x) (M:equal-individuals ?x ?x)))

(20) (forall (?X (M:class ?X))
         (subclass ?X full))
```

There is an *empty* class of individuals $\varnothing = \{x \mid x \neq_{ind} x\} \in \mathit{cls}$.

```
(21) (M:class empty)
     (= empty
        (M:class-of (?x) (not (M:equal-individuals ?x ?x))))
```

There is a class of identifiers $I = \{x \mid I(x)\} \in \mathit{cls}$.

```
(22) (M:class identifier)
     (= identifier
        (M:class-of (?x) (M:identifier ?x)))
```

There is a *singleton* function

$\mathit{sgtn} = \{\text{-}\} : \mathit{ind} \to \mathit{cls}$

from individuals to classes, that maps an individual $a \in \mathit{ind}$ to the class $\{a\} = \{x \in \mathit{ind} \mid x = a\}$.

```
(23) (KIF:function singleton)
     (= (KIF:source singleton) M:individual)
     (= (KIF:target singleton) M:class)
     (forall (?a (M:individual ?a))
         (= (singleton ?a)
            (M:class-of (?x) (M:equal-individuals ?x ?a))))
```

There is a *binary union* function

$\cup : \mathit{cls} \times \mathit{cls} \to \mathit{cls}$

from pairs of classes to classes, that maps a class pair $(X, Y) \in \mathit{cls} \times \mathit{cls}$ to the class $X \cup Y = \{x \in \mathit{ind} \mid x \in X \vee x \in Y\}$.

```
(24) (KIF:function binary-union)
     (= (KIF:source binary-union) (KIF:binary-product [M:class M:class]))
     (= (KIF:target binary-union) M:class)
     (forall (?X (M:class ?X) ?Y (M:class ?Y))
         (= (binary-union [?X ?Y])
            (M:class-of (?x) (or (M:member ?x ?X) (M:member ?x ?Y)))))
```

There is a *binary intersection* function

$\cap : \mathit{cls} \times \mathit{cls} \to \mathit{cls}$

from pairs of classes to classes, that maps a class pair $(X, Y) \in \mathit{cls} \times \mathit{cls}$ to the class $X \cap Y = \{x \in \mathit{ind} \mid x \in X \wedge x \in Y\}$.

```
(25) (KIF:function binary-intersection)
     (= (KIF:source binary-intersection) (KIF:binary-product [M:class M:class]))
     (= (KIF:target binary-intersection) M:class)
     (forall (?X (M:class ?X) ?Y (M:class ?Y))
         (= (binary-intersection [?X ?Y])
            (M:class-of (?x) (and (M:member ?x ?X) (M:member ?x ?Y)))))
```

There is a class *difference* function

$\text{–} : \mathit{cls} \times \mathit{cls} \to \mathit{cls}$

from pairs of classes to classes, that maps a class pair $(X, Y) \in \mathit{cls} \times \mathit{cls}$ to the class $X - Y = \{x \in \mathit{ind} \mid x \in X \wedge x \notin Y\}$.

```
(26) (KIF:function difference)
     (= (KIF:source difference) (KIF:binary-product [M:class M:class]))
     (= (KIF:target difference) M:class)
     (forall (?X (M:class ?X) ?Y (M:class ?Y))
         (= (difference [?X ?Y])
            (M:class-of (?x) (and (M:member ?x ?X) (not (M:member ?x ?Y))))))
```

There is a class *complement* function

$$- : \mathit{cls} \to \mathit{cls}$$

from classes to classes, that maps a class $X \in \mathit{cls}$ to the class difference $V - X$.

```
(27) (KIF:function complement)
     (= (KIF:source complement) M:class)
     (= (KIF:target complement) M:class)
     (forall (?X (M:class ?X))
         (= (complement ?X))
            (difference [full ?Y])))
```

There is a *subclass* or *inclusion* binary relation $\mathit{sub} = \subseteq \subseteq \mathit{cls} \times \mathit{cls}$, where $X \subseteq Y$ holds for any class pair $(X, Y) \in \mathit{cls} \times \mathit{cls}$ iff $(\forall x)(x \in X \to x \in Y)$.

```
(28) (KIF:relation subclass)
     (= (KIF:collection1 subclass) M:class)
     (= (KIF:collection2 subclass) M:class)
     (forall (?X (M:class ?X) ?Y (M:class ?Y)
         (<=> (subclass ?X ?Y)
              (forall (?x (M:individual ?x))
                  (=> (M:member ?x ?X)
                      (M:member ?x ?Y))))))
```

Relations and Functions on Classes

The IFF top metalevel core (meta) ontology (IFF-KIF) contains a *binary product* operator. There is an *ordered pair* function

$$\mathit{pr} = (\text{-}, \text{-}) : \mathit{ind} \times \mathit{ind} \to \mathit{ind}$$

that takes pairs of individuals to individuals.

```
(29) (KIF:function pair)
     (= (KIF:source pair)
        (KIF:binary-product [M:individual M:individual]))
     (= (KIF:target pair) M:individual)
```

This is subject to the **axiom of ordered pairs**.

```
(30) (forall (?x (M:individual ?x) ?y (M:individual ?y)
              ?u (M:individual ?u) ?v (M:individual ?v))
         (<=> (M:equal-individuals (pair [?x ?y]) (pair [?x ?y]))
              (and (M:equal-individuals ?x ?u) (M:equal-individuals ?y ?v)))
```

There is a *binary product* function

$$\times : \mathit{cls} \times \mathit{cls} \to \mathit{cls}$$

from pairs of classes to classes, that maps a class pair $(X, Y) \in \mathit{cls} \times \mathit{cls}$ to the class $X \times Y = \{z \in \mathit{ind} \mid (\exists x \in \mathit{ind}, \exists x \in \mathit{ind})(x \in X \wedge y \in Y \wedge z = (x, y)\}$.

```
(31) (KIF:function binary-product)
     (= (KIF:source binary-product) (KIF:binary-product [M:class M:class]))
     (= (KIF:target binary-product) M:class)
     (forall (?X (M:class ?X) ?Y (M:class ?Y))
         (= (binary-intersection [?X ?Y])
            (M:class-of (?z)
                (exists (?x (M:individual ?x) ?y (M:individual ?y))
                    (and (M:member ?x ?X) (M:member ?x ?Y)
                         (M:equal-individuals ?z (pair [?x ?y])))))))
```

A *binary relation* $R \in \mathit{rel} \subseteq \mathit{cls}$ is class that is a subset $R \subseteq X \times Y$ of the product of its *first* $X \in \mathit{cls}$ and *second* $Y \in \mathit{cls}$ classes.

```
(32) (KIF:collection relation)
     (KIF:subcollection relation M:class)

(33) (KIF:function first)
     (= (KIF:source first) relation)
     (= (KIF:target first) M:class)

(34) (KIF:function second)
     (= (KIF:source second) relation)
     (= (KIF:target second) M:class)

(35) (forall (?R (M:class ?R))
         (=> (M:member ?R relation)
             (subclass ?R (binary-product [(first ?R) (second ?R)])))))
```

An *endorelation* $R \in \mathit{endo} \subseteq \mathit{rel}$ is relation that is a subset $R \subseteq X \times X$ of the product of its *underlying-class* $X \in \mathit{cls}$.

```
(36) (KIF:collection endorelation)
     (KIF:subcollection endorelation relation)

(37) (KIF:function underlying-class)
     (= (KIF:source underlying-class) endorelation)
     (= (KIF:target underlying-class) M:class)
     (= (KIF:restriction underlying-class first)
     (= (KIF:restriction underlying-class second)

(38) (forall (?R (relation ?R))
         (=> (M:member ?R endorelation)
             (subclass ?R (binary-product [(underlying-class ?R) (underlying-class ?R)])))))
```

A *function* $f \in \mathit{ftn} \subseteq \mathit{rel}$ is class that is a subset $f \subseteq X \times Y$ of the product of its *source* $X \in \mathit{cls}$ and *target* $Y \in \mathit{cls}$ classes, where $(\forall x \in \mathit{ind})(\exists! y \in \mathit{ind})((x, y) \in f)$. We use the IFF-KIF definite description operator "**the**" to define this.

```
(39) (KIF:collection function)
     (KIF:subcollection function relation)

(40) (KIF:function source)
     (= (KIF:source source) function)
     (= (KIF:target source) M:class)
     (= (KIF:restriction source first)

(41) (KIF:function target)
     (= (KIF:source target) function)
     (= (KIF:target target) M:class)
     (= (KIF:restriction target second)

(42) (forall (?f (relation ?f))
         (=> (M:member ?f function)
             (and (subclass ?f (binary-product [(source ?f) (target ?f)])
                  (forall (?x (M:individual ?x)) ?y (M:individual ?y))
                      (=> (M:member (pair ?x ?y) ?f)
                          (= ?y (the (?y1 (M:member (pair ?x ?y1) ?f)))))))))
```

There is a *field* function

$$\mathit{fld} : \mathit{endo} \to \mathit{cls}$$

from endorelations to classes, that maps an endorelation $R \in \mathit{endo}$ to the class $\mathit{fld}(R) = \{x \in \mathit{ind} \mid (\exists y \in \mathit{ind})(yRx \vee xRy\}$ of all related individuals.

```
(43) (KIF:function field)
     (= (KIF:source field) endorelation)
     (= (KIF:target field) M:class)
     (forall (?R (endorelation ?R))
         (= (field ?R)
            (M:class-of (?x)
```

```
            (exists (?y (M:individual ?y))
                (or (M:member (pair ?y ?x) ?R) (M:member (pair ?x ?y) ?R))))))
```

There is a *segment* or *below* function

$$\mathit{sgmt} = (\text{-}) : \mathit{endo} \to \mathit{cls}$$

from pairs of endorelations and individuals to classes, that maps a endorelation-individual pair $(R, x) \in \mathit{endo} \times \mathit{ind}$ to the class $R \times x = \{y \in \mathit{ind} \mid (x \neq y \wedge yRx)\}$.

```
(44) (KIF:function segment)
     (= (KIF:source segment) (KIF:binary-product [endorelation M:individual]))
     (= (KIF:target segment) M:class)
     (forall (?R (endorelation ?R) ?x (M:individual ?x))
         (= (segment [?R ?x])
            (M:class-of (?y)
                (and (not (M:equal-individuals ?x ?y))
                     (M:member (pair [?y ?x]) ?R)))))
```

The Individual Aspect

Boolean Operators on Individuals

There is a *singleton* function

$\mathit{sgtn} = [[\text{-}]] : \mathit{ind} \to \mathit{ind}$

on individuals, that maps an individual $a \in \mathit{ind}$ to the individual $[[a]] = \lambda\{a\} = \lambda\{x \mid x =_{\mathit{ind}} a\}$, which is the label of the singleton class.

```
(45) (KIF:function singleton)
     (= (KIF:source singleton) M:individual)
     (= (KIF:target singleton) M:individual)
     (forall (?a (M:individual ?a))
         (= (singleton ?a)
            (M:label (M.C:singleton ?a))))
     (forall (?a (M:individual ?a))
         (= (singleton ?a)
            (M:label-of (?x) (M:equal-individuals ?x ?a))))
```

There is a *binary union* function

$\cup : \mathit{ind} \times \mathit{ind} \to \mathit{ind}$

from pairs of individuals to individuals, that maps an individual pair $(a, b) \in \mathit{ind} \times \mathit{ind}$ to the individual $a \cup b = [[x \mid x \in a \vee x \in b]]$.

```
(46) (KIF:function binary-union)
     (= (KIF:source binary-union)
        (KIF:binary-product [M:individual M:individual]))
     (= (KIF:target binary-union) M:individual)
     (forall (?a (M:individual ?a) ?b (M:individual ?b))
         (= (binary-union [?a ?a])
            (M:label-of (?x)
                (or (M:member-individual ?x ?a) (M:member-individual ?x ?b)))))
```

There is a *binary intersection* function

$\cap : \mathit{ind} \times \mathit{ind} \to \mathit{ind}$

from pairs of individuals to individuals, that maps an individual pair $(a, b) \in \mathit{ind} \times \mathit{ind}$ to the individual $a \cap b = [[x \mid x \in X \wedge x \in Y]]$.

```
(47) (KIF:function binary-intersection)
     (= (KIF:source binary-intersection)
        (KIF:binary-product [M:individual M:individual]))
     (= (KIF:target binary-intersection) M:individual)
     (forall (?a (M:individual ?a) ?b (M:individual ?b))
         (= (binary-intersection [?a ?b])
            (M:label-of (?x)
                (and (M:member-individual ?x ?a) (M:member-individual ?x ?b)))))
```

There is a *union* function

$\cup : \mathit{ind} \to \mathit{ind}$

from individuals to individuals, that maps an individual $a \in \mathit{ind}$ to the individual $\cup a = [[x \mid \exists y(x \in y \vee y \in a)]]$.

```
(48) (KIF:function union)
     (= (KIF:source union) M:individual)
     (= (KIF:target union) M:individual)
     (forall (?a (M:individual ?a))
         (= (union ?a)
            (M:label-of (?x)
                (exists (?y)
                    (and (M:member-individual ?x ?y) (M:member-individual ?y ?a))))))
```

There is an *empty* individual that labels the empty class of individuals $0 = \lambda\varnothing = [[x \mid x \neq_{\mathit{ind}} x]] \in \mathit{ind}$.

```
(49) (M:class empty)
```

```
(= empty (M:label-of (?x) (not (M:equal-individuals ?x ?x))))
(= empty (label empty))
```

There is a *subindividual* or *inclusion* binary relation $\mathit{sub} = \subseteq \subseteq \mathit{ind} \times \mathit{ind}$, where $a \subseteq b$ holds for any individual pair $(a, b) \in \mathit{ind} \times \mathit{ind}$ $\underline{\text{iff}}$ $(\forall x)(x \in a \to x \in b)$.

```
(50) (KIF:relation subindividual)
     (= (KIF:collection1 subindividual) M:individual)
     (= (KIF:collection2 subindividual) M:individual)
     (forall (?a (M:individual ?a) ?b (M:individual ?b)
         (<=> (subindividual ?a ?b)
              (forall (?x (M:individual ?x))
                  (=> (M:member-individual ?x ?a)
                      (M:member-individual ?x ?b)))))
```

There is a *power* function

$$\wp : \mathit{ind} \to \mathit{ind}$$

from individuals to individuals, that maps an individual $a \in \mathit{ind}$ to the individual $\wp a = [[x \mid I(x) \wedge x \subseteq a]]$.

```
(51) (KIF:function power)
     (= (KIF:source power) M:individual)
     (= (KIF:target power) M:individual)
     (forall (?a (M:individual ?a))
         (= (power ?a)
            (M:label-of (?x)
                (and (M:identifier ?x) (subindividual ?x ?a)))))
```

By Cantor's theorem, there are classes that are not sets. These are called "proper" classes. Here are some classes – the two Russell classes and the least inclusive class – that can be explicitly demonstrated to be proper. The Russell classes are define by

$$R_1 = \{x \mid x \notin x^*\} \text{ and } R_2 = \{x \mid I(x) \wedge x \notin x^*\}.$$

Clearly, $R_1 \subseteq R_2$.

```
(52) (M:class russell1)
     (= russell1
        (M:class-of (?x) (not (M:member ?x (M:colabel ?x)))))

(53) (M:class russell2)
     (= russell2
        (M:class-of (?x)
            (and (M:identifier ?x) (not (M:member ?x (M:colabel ?x))))))

(54) (M.C:subclass russell1 russell2)
```

A class X is said to be *inclusive*, denoted by $\mathit{inc}(X)$, (loosely) when any set x that is a subclass of X is also a member of X; that is, (more precisely) when any identifier x, whose colabel set x^* is a subclass of X, is a member of X; that is, (symbolically) when X satisfies

$$(\forall x)[(I(x) \wedge x^* \subseteq X) \to x \in X].$$

```
(55) (KIF:collection inclusive)
     (KIF:subcollection inclusive M:class)
     (forall (?X (M:class ?X))
         (<=> (inclusive ?X)
              (forall (?x (M:identifier ?x))
                  (=> ((M:colabel ?x) M.C:subclass ?X) (M:member ?x ?X)))))
```

The *least inclusive class* E is defined by

$$E = \{x \mid (\forall X)(\mathit{inc}(X) \to x \in X)\}.$$

For any identifier x, if $x^* \subseteq R_2$ then $x \in R_2$. Hence, R_2 is an inclusive set. That is, $E \subseteq R_2 \subseteq I$.

```
(56) (class least-inclusive-class)
     (= least-inclusive-class
        (M:class-of (?x)
            (forall (?X) (=> (inclusive ?X) (M:member ?x ?X)))))
```

```
(57) (M.C:subclass least-inclusive-class russell2)
     (M.C:subclass russell2 M.C:identifier)
```

Theorem. None of the classes R_1, R_2 or E is a set.

```
(58) (not (M:set russell1))
     (not (M:set russell2))
     (not (M:set least-inclusive-class))
```

An individual a is said to be *transitive*, denoted ***trans***(a), when it contains members of all of its members; that, (more precisely) when for any individuals x and y, if x is a member of a and y is a member of x, then y is also a member of a:

$(\forall x)(\forall y)[(x \in a \wedge y \in x) \rightarrow (y \in a)]$.

```
(59) (KIF:collection transitive)
     (KIF:subcollection transitive M:individual)
     (forall (?x (M:identifier ?x))
         (<=> (transitive ?x)
              (forall (?x (M:identifier ?x) ?y (M:identifier ?y))
                  (=> ((M:member-individual ?x ?a) (M:member-individual ?y ?x))
                      (M:member-individual ?y ?a)))))
```

Extensions of M

Following Bell, here are some extensions to **M** in which the existence of certain sets, or equivalently identifiers, is asserted.

Zermelo-Fraenkel

`M.Z`

The system **ZM** is obtained by adding to M the following axioms.

Empty set: The empty individual is a set (identifier): $I(0)$.

```
(1) (M:identifier M.I:empty)
```

Singletons: The singleton of any set (identifier) is a set (identifier): $(\forall x)[I(x) \rightarrow I([[x]])]$.

```
(2) (forall (?x)
        (=> (M:identifier ?x)
            (M:identifier (M.I:singleton ?x))))
```

Union: The union of any set (identifier) is a set (identifier): $(\forall x)[I(x) \rightarrow I(\cup x)]$.

```
(3) (forall (?x)
        (=> (M:identifier ?x)
            (M:identifier (M.I:union ?x))))
```

Power set: The power of any set (identifier) is a set (identifier): $(\forall x)[I(x) \rightarrow I(\wp x)]$.

```
(3) (forall (?x)
        (=> (M:identifier ?x)
            (M:identifier (M.I:power ?x))))
```

Separation: The label-of (label of class-of) of any property is a set (identifier): $(\forall x)[I(x) \rightarrow I([[x \mid \varphi(x)]])]$.

This is an axiom scheme, just as is comprehension and replacement.

Infinity: There is a set (identifier) that is infinite (contains zero and next):
$(\exists u)[I(u) \wedge (0 \in u) \wedge (\forall x \in u\ x \cup [[x]] \in u)]$.

```
(5) (exists (?ux)
        (=> (M:identifier ?u)
            (and (M:member-individual M.I:empty ?u)
                 (forall (?x (M:member-individual ?x ?u))
                     (M:member-individual (M.I:binary-union ?x (M.I:singleton ?x)) ?u)))))
```