ProofCheck's Logic and Set Theory

Definitions

Formal languages do not need to include definitions. But when they do not, formal reasoning is limited to the propositions which can feasibly be stated using only primitive terms. Nonetheless most authors treat definitions as "abbreviations," which stand apart from the formal language itself. As Russell put it:

"It is to be observed that a definition is, strictly speaking, no part of the subject in which it occurs. For a definition is concerned wholly with the symbols, not with what they symbolise. Moreover it is not true or false, being the expression of a volition, not of a proposition."*

In defense of definitions we might accuse Russell of an epistemic bias, devaluing what we know based on how we come to know it. From a deductive point of view it does not matter how we come to know whether a thing is true. Simply that it is true is the point. Axioms, definitions and theorems are all equally valid as references in a proof. Furthermore, failure to treat definitions formally has as a practical consequence reversion to the same sloppy sort of reasoning that formal reasoning was created to protect us from.

Hence we have no special definor symbols. A formula becomes a definition when we label it as such. To define a formula '$\underline{p}$' we use '$(\underline{p} \leftrightarrow \underline{q})$', which when used in a proof simply means that $\underline{p}$ is true if and only if $\underline{q}$ is true. To define a term 'x' we do not use '$(x = y)$.' Instead we use what would seem to be a special symbol in '$(x \equiv y)$'. However the distinction between the two notations reflects an important difference in meaning. By '$(x \equiv y)$' we mean that the terms x and y may be used interchangeably. By '$(x = y)$' we mean that there is some object which is denoted by both x and y. '$(x = y)$' has ontological implications, whereas '$(x \equiv y)$' does not. Further, by no means do we limit the employment of the expression '$(x \equiv y)$' to definitions.

Because maladroit definitions can disrupt the grammar of a formal language or the consistency of a deductive system, there is a need for "rules of definition" which restrict which sorts of formulas may be used as definitions. An overlooked contribution of Morse is his careful formulation of a set of such rules.†

Descriptions and Predicate Logic

The predicate logic used in ProofCheck, see Alps and Neveln [1981], is similar to the predicate logic of Bourbaki in that it is based on a choice operator. The choice operator takes a predicate $\underline{p}x$ as its operand and returns a single object an $x\underline{p}x$ satisfying the predicate. In a sense it is a variant of the set forming operator

* Whitehead and Russell [1927], p11.

† This rule set is given in the Appendix of Morse [1986]. Many of the consequences of this rule set were explicitly demonstrated in Neveln [1975] and Alps[1979].

often written '$\{x : \underline{\mathrm{p}}x\}$', which returns the set of all objects satisfying the predicate. When there is a unique object satisfying a predicate then this object must be what is denoted by the indefinite description. Thus 'an $x(x + x = 4)$' denotes 2. We call it an indefinite description because of its close relation to the definite description 'the $x\underline{\mathrm{p}}x$' used by Russell, Quine and Morse.* Clearly it is also true that 'the $x(x+x = 4)$' denotes 2. In the case where more than one object satisfies a predicate, the definite and indefinite descriptions differ. So although 'an $x(x > 4)$' has a denotation, 'the $x(x > 4)$' does not.†

In the case that there is no object satisfying a given predicate then neither description has a natural denotation. What should the descriptions refer to in this case? What, for example, is an $x(x \neq x)$ or the $x(x \neq x)$? Rather than conscripting some otherwise undistinguished object to serve as the denotation of such terms as Bourbaki does, we haven chosen to admit these terms into the language without forcing them to name something. The language therefore contains non-denoting expressions. We are consequently forced to distinguish these non-denoting expressions from the rest.

The Existence Predicate

By '$\mathrm{ex}\, y$' we mean that the term 'y' denotes an object. We may thus assert that $\neg\, \mathrm{ex}\, \mathrm{an}\, x(x \neq x)$ as well as that $\mathrm{ex}\, \mathrm{an}\, x(x + x = 4)$.

Examples

Valid statements:

1. $\mathrm{ex}(2 + 2)$
2. $\neg\, \mathrm{ex}(0/0)$
3. $\mathrm{ex}\, \mathrm{an}\, x(x > 1)$
4. $(y = \mathrm{an}\, x(x > 1) \rightarrow y > 1)$
5. $(\mathrm{ex}\, \mathrm{an}\, x\underline{\mathrm{p}}x \rightarrow \underline{\mathrm{p}}\, \mathrm{an}\, x\underline{\mathrm{p}}x)$

Invalid statements:

6. $(\underline{\mathrm{p}}y \rightarrow \mathrm{an}\, x\underline{\mathrm{p}}x = y)$
7. $(\underline{\mathrm{p}}y \rightarrow \mathrm{ex}\, \mathrm{an}\, x\underline{\mathrm{p}}x)$
8. $(\overline{2} = \mathrm{an}\, x(x > 1))$
9. $(3 = \mathrm{an}\, x(x > 1))$

Examples 1 through 5 should be self-explanatory. Example 2 is intended to suggest the artificiality of the prohibition in ordinary mathematics against "writing" non-denoting expressions. Example 5 expresses a general truth. Example 6 serves as a reminder that the indefinite description may choose an object other than the one we may have in mind. Although at first sight example 7 appears unobjectionable, it actually does not make sense if 'y' is a non-denoting term. Notice that if '$\neg\, \mathrm{ex}\, x$' is substituted for '$\underline{\mathrm{p}}x$' in both 5 and 7 a contradiction results. Examples 8 and 9, either of which could be true, are intended to demonstrate that undecidable statements in this logic are no calamity.

When this predicate logic is applied to a particular domain '$\mathrm{ex}\, x$' means that x is an object in that domain. Because set theory is taken as the basis for the rest of mathematics it follows that one can always read '$\mathrm{ex}\, x$' as 'x is a set'.

The Existence Quantifier

The existence quantifier, usually written with '$\exists$', asserts that an object satisfying a predicate exists. The existence predicate and the indefinite description just introduced make available a very natural definition of the existence quantifier:

---

* See Whitehead A.N. and Russell [1927] p31, Quine [1951] p146 or Morse [1986] p91.

† The problem of what to do with definite descriptions in this case was handled very differently by Russell, Quine, and Morse.

$$(\bigvee x \underline{\mathrm{p}} x \leftrightarrow \mathrm{ex}\,\mathrm{an}\ x \underline{\mathrm{p}} x)$$

We must emphasize however that the usual axiom

$$(\underline{\mathrm{p}} y \rightarrow \bigvee x \underline{\mathrm{p}} x)$$

is now equivalent to the contradiction producing example 7 above. We use the following modified version of this axiom instead:

$$(\mathrm{ex}\, y \wedge \underline{\mathrm{p}} y \rightarrow \bigvee x \underline{\mathrm{p}} x)$$

The reader needs to take this as a warning that customary inferences such as:

$$(y > 1 \rightarrow \bigvee x (x > 1))$$

are no longer *logically* valid. In this case we have available the fact that

$$(y > 1 \rightarrow \mathrm{ex}\, y)$$

and so of course it does follow that

$$(y > 1 \rightarrow \mathrm{ex}\, y \wedge y > 1 \rightarrow \bigvee x (x > 1))$$

In most cases such reasoning is available and can be used almost automatically. But using this logic places upon us the burden of being aware of such existence conditions as

$$(y > x \rightarrow \mathrm{ex}\, y \wedge \mathrm{ex}\, x)\ .$$

The reader will note that almost every definition in `common.tex` contains existence conditions whose main purpose is to clarify ordinary existential inferences.

Equality and Identity

Indeed with equality also it is always true that

$$(y = x \rightarrow \mathrm{ex}\, y \wedge \mathrm{ex}\, x)$$

However when terms are first introduced, whether in a definition or a theorem, it is often the case that their existence is not yet established and consequently we may not use the ordinary equality predicate, '$(x = y)$' in such situations. Here a special equality predicate '$(x \equiv y)$', which we call identity is used. Identity is equality in the sense of Leibniz. It carries no existence pre-suppositions. It asserts only that the terms 'x' and 'y' may be substituted for one another in valid logical inference. On the other hand'$(x = y)$' means that both 'x' and 'y' denote the same (existing) thing.

Examples

Valid statements:

1. $(x \equiv x)$
2. $(x = x \leftrightarrow \mathrm{ex}\, x)$
3. $(x = y \rightarrow x \equiv y)$
4. $(x \equiv y \wedge \mathrm{ex}\, x \rightarrow x = y)$
5. $(x \equiv y \rightarrow \underline{\mathrm{p}} x \leftrightarrow \underline{\mathrm{p}} y)$

6. $(\mathrm{ex}\, y \wedge y \equiv \mathrm{an}\ x\underline{\mathrm{p}}x \rightarrow \underline{\mathrm{p}}y)$
7. $(y \equiv \mathrm{an}\ x\underline{\mathrm{p}}x \rightarrow \mathrm{ex}\, y \leftrightarrow \bigvee x\underline{\mathrm{p}}x)$
8. $(y = \mathrm{an}\ x\underline{\mathrm{p}}x \rightarrow \bigvee x\underline{\mathrm{p}}x)$
9. $(y = \mathrm{an}\ x\underline{\mathrm{p}}x \rightarrow \underline{\mathrm{p}}y)$
10. $(y \equiv \mathrm{an}\ x\neg\, \mathrm{ex}\, x \rightarrow \neg\, \mathrm{ex}\, y)$

Invalid statements:
11. $(x = x)$
12. $(x \equiv y \rightarrow x = y)$
13. $(y \equiv \mathrm{an}\ x\underline{\mathrm{p}}x \rightarrow \underline{\mathrm{p}}y)$
14. $(\underline{\mathrm{p}}y \rightarrow y \equiv \mathrm{an}\ x\underline{\mathrm{p}}x)$
15. $(\mathrm{ex}\, y \wedge \underline{\mathrm{p}}y \rightarrow y = \mathrm{an}\ x\underline{\mathrm{p}}x)$
16. $\bigvee x\neg\, \mathrm{ex}\, x$

Negative Predicates

Because we believe it useful to arrange most contexts so that they include existence conditions, we define the negatives of ordinary predicates as illustrated in these examples:

$((x \neq y) \leftrightarrow (\mathrm{ex}\, x \wedge \mathrm{ex}\, y \wedge \neg(x = y)))$
$((x \notin y) \leftrightarrow (\mathrm{ex}\, x \wedge \mathrm{ex}\, y \wedge \neg(x \in y)))$
$((x \not\subset y) \leftrightarrow (\mathrm{ex}\, x \wedge \mathrm{ex}\, y \wedge \neg(x \subset y)))$

These negative predicates have the advantage over logical negations in that when used in a context they do not throw the user out of that context.

Examples
Valid statements:
1. $(x \notin y \rightarrow \neg(x \in y))$
2. $(x \notin y \rightarrow \mathrm{ex}\, x \wedge \mathrm{ex}\, y)$
3. $(A \neq \emptyset \leftrightarrow \bigvee x(x \in A))$
4. $(A \not\subset B \rightarrow \bigvee x(x \in A \wedge x \notin B))$
6. $(A = \emptyset \rightarrow \neg(A \neq \emptyset))$

Invalid statements:
7. $(x \notin y \leftrightarrow \neg(x \in y))$
8. $(\neg(A = \emptyset) \leftrightarrow \bigvee x(x \in A))$
9. $(\neg(A \subset B) \rightarrow \bigvee x(x \in A \wedge x \notin B))$
10. $(\neg(A \neq \emptyset) \rightarrow A = \emptyset)$

The Universal Quantifier

The universal quantifier, usually written with '∀', asserts that a predicate is satisfied by all existing objects.

Examples
Valid statements:
1. $\bigwedge x\, \mathrm{ex}\, x$
2. $(\bigwedge x\underline{\mathrm{p}}x \wedge \mathrm{ex}\, y \rightarrow \underline{\mathrm{p}}y)$
3. $(\bigwedge x\underline{\mathrm{p}}x \leftrightarrow \neg \bigvee x\neg\underline{\mathrm{p}}x)$

Invalid statements:
4. $(\bigwedge x\underline{\mathrm{p}}x \rightarrow \underline{\mathrm{p}}y)$

5. $(\underline{\mathrm{p}}y \to \neg \bigwedge x \neg \underline{\mathrm{p}}x)$

The Definite Description and Uniqueness Quantifier

'the $x\underline{\mathrm{p}}x$' denotes the unique x such that $\underline{\mathrm{p}}x$. Thus the $x(x + x = 4)$ is 2 but the $x(x > 1)$ does not exist. We have the following facts

$(y = \text{the } x\underline{\mathrm{p}}x \to \underline{\mathrm{p}}y)$

$(y = \text{the } x\underline{\mathrm{p}}x \to \bigwedge x(\underline{\mathrm{p}}x \to x = y))$

Descriptions, both defnite and indefinite facilitate making definitions. In our logic it is always valid to introduce a description, since there is no implicit assumption that the described object exists. Proving that the object exists does not have to be artfully completed without use of the term before the term can be written.

Existence proofs of described objects are often done using a special quantifier '$\dot{\bigvee} x\underline{\mathrm{p}}x$' which means that there is exactly one x having the property $\underline{\mathrm{p}}x$. It is sometimes written with '$\exists!$'. We can characterize it as existence plus uniqueness as follows:

$(\dot{\bigvee} x\underline{\mathrm{p}}x \leftrightarrow \bigvee x\underline{\mathrm{p}}x \wedge \bigwedge x \bigwedge y(\underline{\mathrm{p}}x \wedge \underline{\mathrm{p}}y \to x = y))$

Hence

$(\mathrm{ex}\,\text{the } x\underline{\mathrm{p}}x \leftrightarrow \dot{\bigvee} x\underline{\mathrm{p}}x)$

A typical example of a definition benefitting from the definite description would be the definition of $\mathrm{Max}\,A$, the largest element of a set A:

$(\mathrm{Max}\,A \equiv \text{the } x \in A \bigwedge y \in A(y \leq x))$

We need no guarantees about A in order to be justified in making this definition. The definition stands on its own. Of course $\mathrm{Max}\,A$ is not known to exist. Indeed it is the case that:

$(\mathrm{ex}\,\mathrm{Max}\,A \leftrightarrow \dot{\bigvee} x \in A \bigwedge y \in A(y \leq x))$

But whether $\mathrm{Max}\,A$ exists or not the definition can be useful. For example we can state that $\mathbf{N}$, the set of natural numbers, has no maximum by asserting that $\neg\,\mathrm{ex}\,\mathrm{Max}\,\mathbf{N}$. Further it is easy prove this statement by setting

$(n \equiv \mathrm{Max}\,\mathbf{N})$

and showing that $\mathrm{ex}\,n$ leads to a contradiction.

As examples of general properties of definite descriptions, we offer the following:

$(\bigwedge x(\underline{\mathrm{p}}x \leftrightarrow \underline{\mathrm{q}}x) \to \text{the } x\underline{\mathrm{p}}x \equiv \text{the } x\underline{\mathrm{q}}x)$

$(\bigwedge x(\underline{\mathrm{p}}x \to \underline{\mathrm{q}}x) \to \text{the } x\underline{\mathrm{p}}x \equiv \text{the } x(\underline{\mathrm{p}}x \wedge \underline{\mathrm{q}}x))$

Examples

Valid statements:

1. $(y = \mathrm{Max}\,A \to y \in A)$
2. $(y \equiv \mathrm{Max}\,A \wedge \mathrm{ex}\,y \to y = \mathrm{Max}\,A)$

3. $(y = \operatorname{Max} A \wedge x \in A \rightarrow x \leq y)$
4. $(y \equiv \operatorname{Max} A \wedge \bigwedge x \in A \bigvee z \in A(x < z) \rightarrow \neg \operatorname{ex} y)$
5. $(y \equiv \operatorname{Max} A \wedge \bigvee x \in A \bigwedge z \in A(z \leq x) \rightarrow \operatorname{ex} y)$
6. $(y \equiv \operatorname{Max} A \wedge \operatorname{ex} y \rightarrow \operatorname{ex} A)$
7. $\neg \operatorname{ex} \operatorname{Max} \emptyset$

Invalid statements:

8. $(y \equiv \operatorname{Max} A \rightarrow y \in A)$
9. $(\bigwedge x \in A(x \leq y) \rightarrow y = \operatorname{Max} A)$
10. $(y \equiv \operatorname{Max} A \wedge \neg \operatorname{ex} y \rightarrow \bigwedge x \in A \bigvee z \in A(x < z))$
11. $(y \equiv \operatorname{Max} A \wedge \neg \bigvee x \in A(x > y) \rightarrow \operatorname{ex} y)$
12. $(y \equiv \operatorname{Max} A \wedge \operatorname{ex} A \rightarrow \operatorname{ex} y)$

Sentence Logic

In addition to the usual notations '$(\underline{p} \wedge \underline{q})$', '$(\underline{p} \vee \underline{q})$', '$\neg \underline{p}$', '$(\underline{p} \rightarrow \underline{q})$', and '$(\underline{p} \leftrightarrow \underline{q})$' for conjunction, disjunction, negation, implication and equivalence respectively we use '$\top$' and '$\bot$' for Boolean true and false statements.

On occasion we also use an exactly once connective:

$(\underline{p} \dot{\vee} \underline{q} \dot{\vee} \underline{r})$

meaning that exactly one of $\underline{p}$, $\underline{q}$, and $\underline{r}$ is true. As an n-ary connector it can be defined using a recursive scheme exemplified by the following definition of the ternary form in terms of the binary:

$((\underline{p} \dot{\vee} \underline{q} \dot{\vee} \underline{r}) \leftrightarrow (\neg \underline{r} \wedge (\underline{p} \dot{\vee} \underline{q})) \vee (\underline{r} \wedge \neg(\underline{p} \vee \underline{q})))$

Plural Forms

Plural forms of sentences allow the use of one verb with two or more subjects. We enable plural forms with a definitional scheme based on commas.* This scheme allows forms such as the following:

$(a, b, < x \leftrightarrow a < x \wedge b < x)$
$(a, b, c, < x \leftrightarrow a < x \wedge b < x \wedge c < x)$
etc.

Modifiers such as those meaning distinct, strictly increasing, and disjoint are also allowed:

$(a, b, \neq, \in T \leftrightarrow a \neq b \wedge a, b, \in T)$
$(a, b, c, \neq, \in T \leftrightarrow a, b, \neq, \in T \wedge a, b, \neq c \wedge c \in T)$
etc.

$(a, b, <, \in T \leftrightarrow a < b \wedge a, b, \in T)$
$(a, b, c, <, \in T \leftrightarrow a, b, <, \in T \wedge a, b, < c \wedge c \in T)$
etc.

The disjointness scheme is based on a binary disjointness connector:

$((a \not| b) \leftrightarrow (a \cap b = \emptyset))$

$(a, b, \not|, \in T \leftrightarrow a \not| b \wedge a, b, \in T)$
$(a, b, c, \not|, \in T \leftrightarrow a, b, \not|, \in T \wedge a, b, \not| c \wedge c \in T)$

* For this purpose Morse used a scheme with a similar appearance based on commas and tuples: $(x, < d \leftrightarrow$ tupleis $x \wedge \bigwedge i \in$ bsdmn $x(\operatorname{crd} ix < d))$

etc.

Scope Notation

Providing for flexibility of expression in a formal language requires making its grammar more complicated, but if this flexibility is used to produce greater simplicity of expression, then we believe that the trade-off is worth it. Rigid syntax makes for obscurity.

Morse in his syntax allows scope qualifiers in formulas such as:

$(\bigwedge x \in A \underline{\mathrm{p}} x \leftrightarrow \bigwedge x (x \in A \rightarrow \underline{\mathrm{p}} x))$
$(\bigwedge x \in A \cup B \underline{\mathrm{p}} x \leftrightarrow \bigwedge x (x \in A \cup B \rightarrow \underline{\mathrm{p}} x))$
$(\bigwedge x ; \underline{\mathrm{p}} x \underline{\mathrm{q}} x \leftrightarrow \bigwedge x (\underline{\mathrm{p}} x \rightarrow \underline{\mathrm{q}} x))$
$(\bigwedge x > y + z ; \underline{\mathrm{p}} x \underline{\mathrm{q}} x \leftrightarrow \bigwedge x (x > y + z \wedge \underline{\mathrm{p}} x \rightarrow \underline{\mathrm{q}} x))$

These usages are available with any of the expressions involving an index variable. In each of these four formulas the scope of the index variable is restricted.

In the first two formulas the scope is restricted by a formula implied by an alternating succession of connectors and terms, which becomes a valid formula if enclosed in parentheses:

$(x \in A \cup B)$

Because no term is allowed to succeed another without an intervening connector, a missing connector signals the end of the scope qualifier. Hence in the first formula the scope qualifier terminates after the 'A' because no connector occurs between 'A' and '$\underline{\mathrm{p}} x$'. (That '$\underline{\mathrm{p}} x$' is a formula and not a term might also be taken as the signal except that in other cases, an indexed union for example, a valid term such as '$\underline{\mathrm{u}} x$' may well occur in that location.)

In the last two formulas a semi-colon is used as the signal. Because the semi-colon is not classed as a connector it can be used in this way. It signals that the scope qualifier ends with the formula which comes next, and that if there is a preceding succession of terms and connectors then the implied preceding formula is conjoined with it to form the complete scope qualifier. In the last formula for example the scope qualifier is:

$((x > y + z) \wedge \underline{\mathrm{p}} x)$

We include in our syntax provision for usages in which the scope indication is deferred until the end of the expression in cases where there is an indexed term, such as '$\underline{\mathrm{u}} x$'. This term then occupies what was formerly the index position immediately following the initial symbol. For example we have the following forms of the indexed union: forms:

$(\bigcup \underline{\mathrm{u}} x \langle x \in A \rangle \equiv \bigcup x \in A \underline{\mathrm{u}} x)$
$(\bigcup \underline{\mathrm{u}} x \langle x ; \underline{\mathrm{p}} x \rangle \equiv \bigcup x ; \underline{\mathrm{p}} x \underline{\mathrm{u}} x)$
$(\bigcup \underline{\mathrm{u}} x \langle x \in A ; \underline{\mathrm{p}} x \rangle \equiv \bigcup x \in A ; \underline{\mathrm{p}} x \underline{\mathrm{u}} x)$

Analogous to these expressions we also allow variations in the classifier notation for sets so that

$(y \in \{\!\{ \underline{\mathrm{u}} x \langle x \in A \rangle \leftrightarrow \bigvee x \in A (y = \underline{\mathrm{u}} x) \wedge y \in \mathrm{U})$
$(y \in \{\!\{ \underline{\mathrm{u}} x \langle x ; \underline{\mathrm{p}} x \rangle \leftrightarrow \bigvee x ; \underline{\mathrm{p}} x (y = \underline{\mathrm{u}} x) \wedge y \in \mathrm{U})$
$(y \in \{\!\{ \underline{\mathrm{u}} x \langle x \in A ; \underline{\mathrm{p}} x \rangle \leftrightarrow \bigvee x \in A ; \underline{\mathrm{p}} x (y = \underline{\mathrm{u}} x) \wedge y \in \mathrm{U})$

With these expressions many notations for sets in wide use informally become available within our formal grammar. Similar expressions for 'the' and 'an', though available are unused in these pages. In cases where

the role of '$\underline{\mathrm{u}}x$' is played simply by the variable x we use the colon to indicate that the x is serving both as work horse and index:

$(\bigcup x \in A : \underline{\mathrm{p}}x \equiv \bigcup x\langle x \in A; \underline{\mathrm{p}}x\rangle)$
$(\mathsf{E}\, x \in A : \underline{\mathrm{p}}x \equiv \mathsf{E}\, x\langle x \in A; \underline{\mathrm{p}}x\rangle)$

Multiple quantification is based on the comma-based plural forms introduced above and repetition of the singly indexed form. For example we have

$(\bigwedge x, y, \in A\underline{\mathrm{p}}'xy \leftrightarrow \bigwedge x \bigwedge y(x, y, \in A \rightarrow \underline{\mathrm{p}}'xy))$
$(\bigwedge x, y\underline{\mathrm{p}}'xy \leftrightarrow \bigwedge x \bigwedge y\underline{\mathrm{p}}'xy)$

In each multiply indexed form, the repetition of indices is indicated by an alternation of variables and commas. In the first of these formulas this alternation terminates with '$,\in$'. It is because this expression is not a comma that we know that A is not another index variable.

In most indexed forms involving a term we pass the multiple index along to another form which in turn relies on repetition:

$(\bigcup \underline{\mathrm{u}}'xy\langle x, y, \in A\rangle \equiv \mathsf{E}\, z \bigvee x, y, \in A(z \in \underline{\mathrm{u}}'xy))$
$(\bigcup x, y, \in A; \underline{\mathrm{p}}'xy\underline{\mathrm{u}}'xy \equiv \mathsf{E}\, z \bigvee x, y, \in A; \underline{\mathrm{p}}'xy(z \in \underline{\mathrm{u}}'xy))$
$(z \in \mathsf{E}\, \underline{\mathrm{u}}'xy\langle x, y, \in A\rangle \leftrightarrow \bigvee x, y, \in A(z = \underline{\mathrm{u}}'xy \wedge z \in \mathrm{U}))$

In deferred scope forms of more than one index variable we allow the use of 'and' to separate scope qualifying formulas for distinct variables:

$(\bigcup \underline{\mathrm{u}}'xy\langle x \in A, y \in B\rangle \equiv \mathsf{E}\, z \bigvee x \in A \bigvee y \in B(z \in \underline{\mathrm{u}}'xy))$
$(\bigcup \underline{\mathrm{u}}'xy\langle x \in A, y \in B; \underline{\mathrm{p}}'xy\rangle \equiv \mathsf{E}\, z \bigvee x \in A \bigvee y \in B; \underline{\mathrm{p}}'xy(z \in \underline{\mathrm{u}}'xy))$

Local Variables

In definitions where some complicated term occurs more than once the definition can sometimes be shortened by using a temporary variable in place of the unweildy term. For example if we were to define the notion that the domain of a function f had no maximum we might use:

$\bigvee A = \mathrm{dmn}\, f \bigwedge x \in A \bigvee y \in A(y > x)$

The scope of the variable 'A' is restricted to a single value by the scope qualifying formula '$(A = \mathrm{dmn}\, f)$'. When a term is being defined and not a formula then a quantifier cannot be used. In these situations we have the following form available which is defined using a description and a quantifier:

$(\Diamond\, x\underline{\mathrm{u}}x \equiv \mathrm{the}\; t \bigvee x(t = \underline{\mathrm{u}}x))$

Suppose then that we wished to define the largest element of the domain of f, using 'A' to abbreviate 'dmn f'. Then using a scope qualifying formula '$(A = \mathrm{dmn}\, f)$', the following term would do:

$\Diamond\, A = \mathrm{dmn}\, f\, \mathrm{the}\; x \in A \bigwedge y \in A(y \leq x)$

This abbreviation can be read 'letting $A = \mathrm{dmn}\, f$'.

The Conditional Operator

The conditional operator restricts the context in which a term has a denotation.

$$((\underline{p} \mapsto y) \equiv \text{the } x(\underline{p} \wedge x = y))$$

When $\underline{p}$ is false the description becomes a non-denoting term. The described object fails to exist. The conditional operator is characterized by these two properties:

$$(\text{ex}(\underline{p} \mapsto y) \leftrightarrow \underline{p} \wedge \text{ex}\, y)$$

$$(z = (\underline{p} \mapsto y) \leftrightarrow \underline{p} \wedge z = y)$$

Often, satisfaction of the condition will ensure the existence of the object. The following theorems demonstrate the consequences of such a definition.

$$((\underline{p} \rightarrow \text{ex}\, y) \rightarrow (\text{ex}(\underline{p} \mapsto y) \leftrightarrow \underline{p}))$$

$$((\underline{p} \rightarrow \text{ex}\, y) \rightarrow (\underline{p} \leftrightarrow (\underline{p} \mapsto y) = y))$$

Using a condition in a definition means that whenever the defined object exists, all of the conditions restricting its context hold. Most of the definitions in the file `common.tex` include existence conditions. For example, we define the form '$.fx$' which denotes the value of the function f at the point x subject to the condition that f be a function. As a consequence the inference from the usage of the term '$.fx$' that f be a function need no longer be based on confidence in its author's stylistic integrity. It becomes a matter of logic. It works like this. In a standard way we define functionp f to mean that f is a function:

$$(\text{functionp}\, f \leftrightarrow (f \subset \text{U} \times \text{U} \wedge \bigwedge x \overline{\bigwedge} y((x, y) \in f)))$$

We then use this predicate as an existence condition in the definition of $.fx$ as follows:

$$(.fx \equiv (\text{functionp}\, f \wedge x \in \text{dmn}\, f \mapsto \text{the } y((x, y) \in f)))$$

Because of these existence conditions we have available contextual inferences such as:

$(y = .fx \rightarrow \text{functionp}\, f)$
$(.fx < .fy \rightarrow \text{functionp}\, f)$
$(.fx \neq \emptyset \rightarrow \text{functionp}\, f)$
$(y = .fx \rightarrow x \in \text{dmn}\, f)$
$(.fx < .fy \rightarrow x, y, \in \text{dmn}\, f)$

The hypothesis list on many of our theorems is much shorter than it would otherwise be precisely because of this kind of logic. The extent to which conditions of this sort should be used is a matter of stylistic preference. Some may prefer explicit reminders of each contextual assumption made and regard implications of the sort just illustrated as a form of concealment. Others may regard conditions included in this way as matter rightfully taken for granted and which ought not to be overly belabored. Our tendency has been to use these conditions liberally but nothing in our axiom system necessitates this policy.

We now give one more example. We define the disjoint union of two sets A and B as follows:

$$((A \sqcup B) \equiv (A \cap B = \emptyset \mapsto A \cup B \cup C))$$

When the term '$(A \sqcup B)$' occurs in a context which assures us that it is a denoting term this logically implies that $(A \cap B = \emptyset)$. In such a context $(A \sqcup B = A \cup B)$.

Examples

Valid statements:

1. $(y = .fx \rightarrow (x, y) \in f)$
2. $(\text{functionp}\, f \wedge (x, y) \in f \rightarrow y = .fx)$
3. $(y = .fx \wedge (x, y') \in f \rightarrow y' = y)$
4. $((x, y) \in f \wedge y \equiv .fx \rightarrow y = .fx)$
5. $(.fu < .fv \rightarrow \text{functionp}\, f)$
6. $(u \equiv v \rightarrow .fu \equiv .fv)$
7. $(\text{ex}(A \sqcup B) \leftrightarrow A \cap B = \emptyset)$
8. $(A \cup B = A \sqcup B \leftrightarrow A \cap B = \emptyset)$

Invalid statements:

9. $(y \equiv .fx \wedge (x, y') \in f \rightarrow y' = y)$
10. $(u = v \rightarrow .fu = .fv)$
11. $(C \equiv A \sqcup B \rightarrow A \cap B = \emptyset)$
12. $((A \cup B) \cap C = \emptyset \rightarrow (A \sqcup B) \cap C = (A \cap C) \sqcup (B \cap C))$
13. $((A \sqcup B) \cap C \equiv (A \cap C) \sqcup (B \cap C))$

Set Theory

The set theory used here is essentially that of Morse or Kelley.* Because of the pervasive role of existence conditions in what follows however, we urge the reader to review the basic definitions of set theory in order to clarify the role these conditions play.

Using set theory as our mathematical foundation means that all existing objects are sets. For this reason and no other it follows that $\text{ex}\, A$ is equivalent to A being a set (or 'class' actually if one prefers to reserve the term 'set' for those classes which are capable of membership.)

The fundamental undefined term of our set theory is '$\underline{\exists}\, x \underline{\text{p}} x$' which denotes the set of x such that $\underline{\text{p}} x$.* It is valuable to realize that this term unlike most terms we use, always has a denotation. No matter what predicate $\underline{\text{p}} x$ we employ, the set of objects which satisfies this predicate exists. In Zermelo-Frankel set theory, perhaps the most widely used set theory, this would not hold. The Zermelo-Frankel approach avoids the Russell paradox by postulating the existence of sets only when certain conditions are satisfied. Using our logic this would not rule out or even restrict, the use of a set-forming operator '$\underline{\exists}\, x \underline{\text{p}} x$'. In fact in Zermelo-Frankel set theory we could prove that $\neg\, \text{ex}\, \underline{\exists}\, x(x \notin x)$. However we prefer a setting in which it is always the case that $\text{ex}\, \underline{\exists}\, x \underline{\text{p}} x$. Hence we shall assume that our set theory is one like of that of Morse or Kelley in that it has a universe U whose elements are those sets which are capable of membership. In these set theories the basic set membership property is that:

$$(y \in \underline{\exists}\, x \underline{\text{p}} x \leftrightarrow \underline{\text{p}} y \wedge y \in \text{U})$$

This membership property avoids the Russell paradox and instead has as a consequence that

$$(\underline{\exists}\, x(x \notin x) \notin \text{U}).$$

We note that $\underline{\exists}\, x \underline{\text{p}} x$ always exists.

---

* See Morse [1986] or the appendix to Kelley [1955].

* Although there is nothing incorrect about the more usual notation '$\{x : \underline{\text{p}} x\}$' we use the '$\underline{\exists}\, x \underline{\text{p}} x$' notation because its "quantifer" style makes it more compatible with other bound variable notations we use.

* Even with other fundamental set-theoretic operations this is not the case. Our definitions of union, intersection, complement, etc., all include existence conditions. We define

$((A \cup B) \equiv (\mathrm{ex}\, A \wedge \mathrm{ex}\, B \mapsto \mathsf{E}\, x(x \in A \vee x \in B)))$
$((A \cap B) \equiv (\mathrm{ex}\, A \wedge \mathrm{ex}\, B \mapsto \mathsf{E}\, x(x \in A \wedge x \in B)))$
$(\sim A \equiv (\mathrm{ex}\, A \mapsto \mathsf{E}\, x \neg(x \in A)))$
$((A \subset B) \leftrightarrow (\mathrm{ex}\, A \wedge \mathrm{ex}\, B \wedge \bigwedge x(x \in A \rightarrow x \in B)))$

Consequently for example in order that $(A \cup B)$ exist both A and B must exist.

Existence conditions may occur implicitly or by inheritance as in the following set difference operations:

$((A \backsim B) \equiv (A \cap \sim B))$
$((A \triangle B) \equiv ((A \backsim B) \cup (B \backsim A)))$

Although in set theory most existence conditions simply require that the operands exist, the ordinary listing operation benefits from a more stringent condition. Suppose we were to use as a definition the following:

$(\{x\} \equiv (\mathrm{ex}\, x \mapsto \mathsf{E}\, t(t = x)))$

Because the fundamental set membership property excludes proper classes such as U from membership, we would have as a consequence of this bad definition that:

$(\{\mathrm{U}\} = \emptyset)$

Of course the same consequence would follow if we defined '$\{x\}$' without any existence condition. For this reason we require that all listed elements be capable of membership:

$(\{x\} \equiv (x \in \mathrm{U} \mapsto \mathsf{E}\, t(t = x)))$
$(\{x, y\} \equiv (x \in \mathrm{U} \wedge y \in \mathrm{U} \mapsto \mathsf{E}\, t(t = x \vee t = y)))$
$(\{x, y, z\} \equiv (x \in \mathrm{U} \wedge y \in \mathrm{U} \wedge z \in \mathrm{U} \mapsto \mathsf{E}\, t(t = x \vee t = y \vee t = z)))$

Hence we have instead that:

$(\mathrm{ex}\{x\} \leftrightarrow x \in \mathrm{U})$
$(\mathrm{ex}\{x, y\} \leftrightarrow x \in \mathrm{U} \wedge y \in \mathrm{U})$
$(\mathrm{ex}\{x, y, z\} \leftrightarrow x \in \mathrm{U} \wedge y \in \mathrm{U} \wedge z \in \mathrm{U})$

The set of all singletons of members in a set A is:

$(\mathrm{ss}\, A \equiv (\mathrm{ex}\, A \mapsto \mathsf{E}\{x\}\langle x \in A\rangle))$

The set of all subsets of a set A, or the power set of A, is

$(\mathrm{sb}\, A \equiv (\mathrm{ex}\, A \mapsto \mathsf{E}\, x\langle x \subset A\rangle))$

The element choosing operation because it is defined with a description does not of course in all cases result in a denoting term, but it requires no additional existence conditions:

* Using an assignment in a proof typically means proving a statement such as $(\underline{p} \wedge (A \equiv B) \rightarrow \underline{q})$ where this in turn implies that $(\underline{p} \rightarrow \underline{q})$ because we can move the assignment to the front to obtain $(A \equiv B \rightarrow (\underline{p} \rightarrow \underline{q}))$, substitute B for 'A' in this expression and get that $(\underline{p} \rightarrow \underline{q})$ by modus ponens since $(B \equiv B)$ is always true. This same approach only works with $(A = B)$ if we can prove $(B = B)$, i.e. that B exists.

$$(\text{mel}\, A \equiv \text{an}\; x(x \in A))$$

This notation is Morse's and is a play on the name of Zermelo, who originally formulated of the axiom of choice.

Examples

Valid statements:

1. $(\text{ex}(A \cup B) \rightarrow \text{ex}\, A \wedge \text{ex}\, B)$
2. $(\text{ex}\, A \wedge \text{ex}\, B \rightarrow \text{ex}(A \cup B))$
3. $(C = A \cup B \rightarrow \text{ex}\, A)$
4. $(D = C \cup (B \cap {\sim}A) \rightarrow \text{ex}\, A)$
5. $(A \cup B = B \leftrightarrow A \subset B)$
6. $(C = A \cup B \rightarrow A \subset C)$
7. $(A = B \leftrightarrow A \subset B \wedge B \subset A)$
8. $(A \cap (B \cup C) \equiv (A \cap B) \cup (A \cap C))$
9. $({\sim}(A \cup B) \equiv {\sim}A \cap {\sim}B)$
10. $(A \backsim A = \emptyset \leftrightarrow \text{ex}\, A)$

Invalid statements:

11. $(C \equiv A \cup B \rightarrow \text{ex}\, A)$
12. $(A \subset A \cup B)$
13. $(A \equiv B \leftrightarrow A \subset B \wedge B \subset A)$
14. $(A \cap (B \cup C) = (A \cap B) \cup (A \cap C))$
15. $({\sim}(A \cup B) = {\sim}A \cap {\sim}B)$
16. $(A \backsim A = \emptyset)$
17. $(A \backsim A \equiv \emptyset)$

Families of Sets

It is not uncommon to refer to an indexed collection A_i of sets as a family of sets especially when i ranges over a given index set I. Such terminology is often used in the explanation of indexed unions, intersections, and other bound variable forms and appears to refer implicitly to a function. However if we allow proper classes in the family of sets, something we shall in fact do, then we are forced to realize that the concept of function implicit here is not that embodied in the definition of a function as a set of ordered pairs. We need only consider an ordered pair (i, A_i) as a candidate for membership in a set of ordered pairs f, where A_i is a proper class and the difficulty becomes apparent. *

The difficulty disappears if we cede it to logic. Logic already handles formally similar expressions $\underline{\text{p}}x$ which we call predicates. These predicates are statements which are true or false. Russell even referred to them as propositional functions. The functions we need here are similar to these propositional functions except that they are terms, $\underline{\text{u}}x$. They refer to sets in the same way that predicates refer to statements. If we know the rules for substitution for predicates we can use these same rules to substitute for these functions.

Consider the following definition of the indexed intersection. For the sake of simplicity we omit for the moment the existence conditions:

$$(\bigcap x; \underline{\text{p}}x\underline{\text{u}}x \equiv \mathsf{E}\, y \bigwedge x; \underline{\text{p}}x(y \in \underline{\text{u}}x))$$

Although we might refer to the family of sets $\underline{\text{u}}x$ where x ranges over the set $\mathsf{E}\, x\underline{\text{p}}x$, in fact no mathematical description of the nature of the $\underline{\text{u}}x$ is called for here. Any well-formed mathematical term may be substituted for it. Morse referred to these things as "schematic expressions". It should be remembered that Euler at times defined ordinary mathematical functions as "analytical expressions". **

* Indeed in many set theories there is no such ordered pair.

** Rather than concluding that Euler was naive it is better to realize that the boundary between logic and mathematics undergoes historical shifts.

If we consider that we defined the binary union $(A \cup B)$ with an existence condition on each of the two terms A and B, it makes sense to impose an existence condition on each $\underline{\mathrm{u}}x$ in the family, that is on each $\underline{\mathrm{u}}x$ for which the predicate $\underline{\mathrm{p}}x$ is true:

$$(\bigcap x; \underline{\mathrm{p}}x\underline{\mathrm{u}}x \equiv (\bigwedge x; \underline{\mathrm{p}}x \operatorname{ex} \underline{\mathrm{u}}x \mapsto \mathsf{E}\, y \bigwedge x; \underline{\mathrm{p}}x(y \in \underline{\mathrm{u}}x)))$$

Similarly we define the indexed union as follows:

$$(\bigcup x; \underline{\mathrm{p}}x\underline{\mathrm{u}}x \equiv (\bigwedge x; \underline{\mathrm{p}}x \operatorname{ex} \underline{\mathrm{u}}x \mapsto \mathsf{E}\, y \bigvee x; \underline{\mathrm{p}}x(y \in \underline{\mathrm{u}}x)))$$

One important principle on which Morse's grammar is based is that no well-formed expression may begin another. As a consequence we may not use $\bigcup x$ by itself to mean $\bigcup y \in xy$. Rather than devise a third pair of cup and cap symbols we use Morse's sigma and pi notations for the union and intersection of a set of sets:

$$(\bigtriangledown x \equiv (\operatorname{ex} x \mapsto \bigcup y \in xy))$$
$$(\prod x \equiv (\operatorname{ex} x \mapsto \bigcap y \in xy))$$

Classes as Extensions of Predicates

The logical view of a class as the extension of a predicate, that is the class of all objects satisfying the predicate, lends itself to the reading of membership statements as ordinary subject predicate assertions. Hence '$(x \in \mathrm{fnt})$' can be read: x is finite. In this vein we define

$$(\mathrm{disjoint} \equiv \mathsf{E}\, A \bigwedge x, y, \neq, \in A(x \not\cap y))$$

Hence '$(A \in \mathrm{disjoint})$' can be read: A is disjoint. Continuing in this vein we define

$$(\mathrm{partition}\, A \equiv \mathsf{E}\, B \in \mathrm{disjoint}(\emptyset \notin B \wedge \bigtriangledown B = A))$$

Hence '$(B \in \mathrm{partition}\, A)$' can be read: B is a partition of A.

Ordered Pairs

In *Principia Mathematica* the ordered pair was not defined. Pairs occurred but were handled syntactically. The first to treat the pair as having a semantics was Kuratowski who as a result of discussions with Łukasiewicz* defined the pair as follows:

$$((a, b) \equiv \{\{a\}, \{a, b\}\})$$

Unfortunately when used with proper classes it fails to have the desirable property that:

$$((a, b) = (c, d) \leftrightarrow a = c \wedge b = d)$$

We use Morse's ordered pair * which has this property as well as the following two:

$$(\operatorname{ex}(a, b) \leftrightarrow \operatorname{ex} a \wedge \operatorname{ex} b)$$
$$((a, b) \in \mathrm{U} \leftrightarrow a \in \mathrm{U} \wedge b \in \mathrm{U})$$

* See Kuratowski [1977].
* See pp 85-88 of Morse [1986].

The forms we use for accessing the two coordinates of a pair p are $\mathrm{crd}'\, p$ and $\mathrm{crd}''\, p$. Thus

$$(p = (a, b) \to \mathrm{crd}'\, p = a \wedge \mathrm{crd}''\, p = b)$$

The Cartesian product of two sets is defined

$$((A \times B) \equiv (\mathrm{ex}\, A \wedge \mathrm{ex}\, B \mapsto \mathsf{E}(x, y)\langle x \in A, y \in B\rangle))$$

Relations and Functions

A relation is a set of ordered pairs.

$$(\mathrm{relationp}\, R \leftrightarrow (R \subset \mathrm{U} \times \mathrm{U}))$$

The inverse of a relation is the relation:

$$(\mathrm{inv}\, R \equiv (\mathrm{relationp}\, R \mapsto \mathsf{E}(y, x)\langle x, y \in R\rangle))$$

The composition of two relations R and S is defined by

$$((R \circ S) \equiv (\mathrm{relationp}\, R \wedge \mathrm{relationp}\, S \mapsto \mathsf{E}(x, z)\langle x, z; \bigvee y((x, y) \in R \wedge (y, z) \in S)\rangle))$$

A function f is a special relation:

$$(\mathrm{functionp}\, f \leftrightarrow (\mathrm{relationp}\, f \wedge \bigwedge x \overline{\bigwedge} y((x, y) \in f)))$$

The following definition allows for the more natural syntax '$(f \in \mathrm{function})$' which can be used whenever we are sure that f can not be a proper class:

$$(\mathrm{function} \equiv \mathsf{E}\, f\, \mathrm{functionp}\, f)$$

We define the domain and range of a function:

$$(\mathrm{dmn}\, f \equiv (\mathrm{functionp}\, f \mapsto \mathsf{E}\, x \bigvee y((x, y) \in f)))$$
$$(\mathrm{rng}\, f \equiv (\mathrm{functionp}\, f \mapsto \mathsf{E}\, y \bigvee x((x, y) \in f)))$$

The value of a function f at a point x is

$$(.\, fx \equiv (\mathrm{functionp}\, f \wedge x \in \mathrm{dmn}\, f \mapsto \mathrm{the}\, y((x, y) \in f)))$$

In the same vein we define

$$(\mathrm{one\text{–}to\text{–}onep}\, f \leftrightarrow (\mathrm{functionp}\, f \wedge \bigwedge x, y(.\, fx = .\, fy \to x = y)))$$
$$(\mathrm{one\text{–}to\text{–}one} \equiv \mathsf{E}\, f\, \mathrm{one\text{–}to\text{–}onep}\, f)$$

And we define arrow notations:

$$(.\, f : A \to B \leftrightarrow (\mathrm{functionp}\, f \wedge \mathrm{dmn}\, f = A \wedge \mathrm{rng}\, f \subset B))$$
$$(.\, f : A \rightarrowtail B \leftrightarrow (\mathrm{one\text{–}to\text{–}onep}\, f \wedge \mathrm{dmn}\, f = A \wedge \mathrm{rng}\, f \subset B))$$
$$(.\, f : A \twoheadrightarrow B \leftrightarrow (\mathrm{functionp}\, f \wedge \mathrm{dmn}\, f = A \wedge \mathrm{rng}\, f = B))$$
$$(.\, f : A \leftrightarrow B \leftrightarrow (\mathrm{one\text{–}to\text{–}onep}\, f \wedge \mathrm{dmn}\, f = A \wedge \mathrm{rng}\, f = B))$$

We also have inverse image and image notations:

$$(^*fB \equiv (\text{relationp}\, f \wedge \text{ex}\, B \mapsto \mathsf{E}\, x \bigvee y \in B((x,y) \in f)))$$
$$(_*fA \equiv (\text{relationp}\, f \wedge \text{ex}\, A \mapsto \mathsf{E}\, y \bigvee x \in A((x,y) \in f)))$$

For function construction we have:

$$(\lambda\!\!\!/\, x \underline{\text{u}} x \equiv \mathsf{E}(x, \underline{\text{u}} x)\langle x \rangle)$$
$$(\lambda\!\!\!/\, x \in A \underline{\text{u}} x \equiv \mathsf{E}(x, \underline{\text{u}} x)\langle x \in A \rangle)$$
$$(\lambda\!\!\!/\, x; \underline{\text{p}} x \underline{\text{u}} x \equiv \mathsf{E}(x, \underline{\text{u}} x)\langle x; \underline{\text{p}} x \rangle)$$

The backwards lambda notation, read "lonzo", is due to Morse, and is a play on the name of Alonzo Church, creator of the lambda calculus. For two index variables we have:

$$(\lambda\!\!\!/\, x, y \underline{\text{u}}' xy \equiv \mathsf{E}((x,y), \underline{\text{u}}' xy)\langle x, y \rangle)$$
$$(\lambda\!\!\!/\, x, y; \underline{\text{p}}' xy \underline{\text{u}}' xy \equiv \mathsf{E}((x,y), \underline{\text{u}}' xy)\langle x, y; \underline{\text{p}}' xy \rangle)$$

The restriction of a function to a subset of its domain is defined:

$$(\text{strc}\, fA \equiv (\text{ex}\, A \mapsto \lambda\!\!\!/\, x(x \in A \mapsto .\, fx)))$$

Case Operators

The following case operators are used principally in the definition of functions

$$((x \diamond y) \equiv \text{the}\, z(z = x \vee z = y))$$

$$((x \triangleright y) \equiv (x \diamond (\neg\, \text{ex}\, x \mapsto y)))$$

The first operator is used mainly in situations where not both x and y exist. In this case '$(x \diamond y)$' denotes the one which does exist. The second operator is similar except that if both x and y exist '$(x \triangleright y)$' denotes x whereas '$(x \diamond y)$' becomes non-denoting. A typical example of their use would be in the definition of absolute value which normally is written using separate lines and a brace to separate the cases:

$$|x| \equiv \begin{cases} x, & \text{if } x \geq 0; \\ (-x), & \text{otherwise.} \end{cases}$$

Although this picturesque syntax suggests that the definition resists ready assimilation into a formal language, with a case operator the definition formalizes without difficulty:

$$(|x| \equiv (x \geq 0 \mapsto x \triangleright -x))$$

The Natural Numbers

The set of natural numbers is $\mathbf{N}$. We have that a set A is finite if and only if the number of its elements, $\#A$, is a natural number:

$$(A \in \text{fnt} \leftrightarrow \#A \in \mathbf{N})$$

The natural numbers consist of the even numbers and the odd numbers:

$$(\text{even} \equiv \mathsf{E}(2 \cdot n)\langle n \in \mathbf{N} \rangle)$$

$$(\text{odd} \equiv \mathsf{E}(2 \cdot n + 1)\langle n \in \mathbf{N}\rangle)$$

Given a family of sets $\underline{\mathrm{u}}x$ we define the symmetric difference of that family:

$$(\bigtriangleup x \underline{\mathrm{u}} x \equiv \mathsf{E}\, y(\#\, \mathsf{E}\, x(y \in \underline{\mathrm{u}} x) \in \text{odd}))$$

Thus y is in the symmetric difference if and only if y belongs to an odd number of members of the family.

A finite sequence is a function whose domain is an interval of the form 0 through $(n-1)$ where n is a natural number:

$$(\text{Int}\, n \equiv (n \in \mathbf{N} \mapsto \mathsf{E}\, k \in \mathbf{N}\, (k \leq n)))$$
$$(\text{fsqnc} \equiv \mathsf{E}\, f \in \text{function} \bigvee n \in \mathbf{N}\, (\text{dmn}\, f = \text{Int}\, n))$$

For finite sequences we have defined a number of utilities:

$$(\text{first}\, f \equiv .\, f\, \text{Min}\, \text{dmn}\, f)$$
$$(\text{last}\, f \equiv .\, f\, \text{Max}\, \text{dmn}\, f)$$
$$(\text{ddmn}\, f \equiv (\text{dmn}\, f \backsim \{\text{Max}\, \text{dmn}\, f\}))$$
$$(\text{Rev}\, f \equiv (f \in \text{fsqnc} \mapsto \diamondsuit\, n(\text{dmn}\, f = \text{Int}\, n \mapsto \swarrow j \in \text{Int}\, n\, .\, f(n - j - 1))))$$
$$(\text{concat}\, fg \equiv \diamondsuit\, m = \text{Max}\, \text{dmn}\, f \swarrow n(.\, fn \triangleright .\, g(n - m - 1)))$$
$$(\text{splice}\, fg \equiv (\text{first}\, g = \text{last}\, f \mapsto \diamondsuit\, m = \text{Max}\, \text{dmn}\, f \swarrow n(.\, fn \triangleright .\, g(n - m))))$$

The first and last items in a finite sequence often need access. The "diminished domain" of a finite sequence is often used in statements relating consecutive items in a finite sequence. Rev f is the reverse of f. The concat fg concatenates two finite sequences f and g. splice fg does nearly the same thing but only works if $(\text{last}\, f = \text{first}\, g)$. If $(\text{dmn}\, f = \text{Int}\, m)$ and $(\text{dmn}\, g = \text{Int}\, n)$ then $(\text{dmn}\, \text{concat}\, fg = \text{Int}(m + n))$ and $(\text{dmn}\, \text{splice}\, fg = \text{Int}(m + n - 1))$.

A sequence is a function whose domain is the set of natural numbers. The set of sequences whose range is a subset of the set A is defined by:

$$(\text{sqnc} \equiv \mathsf{E}\, f \in \text{function}(\text{dmn}\, f = \mathbf{N}))$$