3 *FUNCTIONS*

1. Introduction

Probably the most important concept of mathematics is the concept of *function*. Indeed, most of the traditional branches of mathematics (arithmetic, algebra, analysis) are principally concerned with various kinds of functions (addition, multiplication, exponentiation, sine, cosine, etc.)

Although it is not universal to mathematics, in set theory at least, functions are regarded as special kinds of relations, which in turn are regarded as special kinds of sets (sets of ordered-pairs, to be precise). The set-theoretic identification of functions with special sets of ordered-pairs is oftentimes described by saying that set theory identifies a function with its graph. In other words, if "two" functions have the same graph, they are identical. We often think of graphs as pictures, but what are they pictures of? Answer: sets of ordered-pairs.

A few mathematicians prefer to say that set theory *confuses* functions with their graphs. But if you ask them what more a function is, over and above its graph, you are not likely to get a philosophically satisfactory answer. When push comes to shove, they simply say a function is a set (or perhaps class) of ordered-pairs.

The explosion of mathematics in the last two centuries has occurred primarily because mathematicians gradually relinquished the antique and anthropocentric idea that a function *does* rather than simply *is*. The idea that a function *does* is related to the problem of confusing the meaning of, say, the plus-sign in arithmetic with what humans (or worse yet, electronic calculators) *do* in order to calculate $x+y$ for a particular x and y. This is like confusing a place (e.g., the continent Antarctica) with the directions for getting there. Indeed, in light of its modern analysis, the term 'function' is misleading, but so are many technical terms in the various exact branches of knowledge.

2. Functions as Relations

In order to understand functions, we introduce a few technical terms. In the following, R is (informally) presumed to be a relation, even though the definitions work whether R is a relation or not.

(D1) R is divergent $=_{df}$ $\exists xyz(xRy \;\&\; xRz \;\&\; y{\neq}z)$

(D2) R is convergent $=_{df}$ $\exists xyz(xRz \;\&\; yRz \;\&\; x{\neq}y)$

In other words, to say that R is divergent is to say that at least one thing bears R to at least two (distinct) things. For example, the relation of motherhood is divergent. On the other hand, the relation of daughterhood (restricted to females) is not divergent. Next, to say that R is convergent is to say that at least two (distinct) things bear R to at least one thing. The aforementioned daughterhood relation is convergent, whereas the motherhood relation is not convergent.

Neither of these terms is standard, although each is intuitively clear. Furthermore, they enable us to define some other terms, which are standard, but counterintuitive.

(D3) R is many-one $=_{df}$ R is not divergent

(D4) R is one-many $=_{df}$ R is not convergent

(D5) R is many-many $=_{df}$ R is both divergent and convergent

(D6) R is one-one $=_{df}$ R is neither divergent nor convergent.

The trouble with the first two terms is that they suggest something different from what they officially mean. The term 'many-one' suggests that many things are related to one, whereas in fact it means that one thing is not related to many. Similarly, whereas 'one-many' suggests that one thing is related to many, it in fact means that many things are not related to one. For this reason, we use the terms 'divergent' and 'convergent'.

Now, according to set theory (and modern mathematics), a function is simply a relation that is many-one (i.e., not divergent). To say R is a function is to say that, although many things may be related by R to one thing, no one thing is related by R to many. The following is the formal definition.

(D7) $fun[A] =_{df} \mathcal{R}[A]$ & $m\text{-}o[A]$

Since every function is a relation, we can apply many of the relational notions, including domain, range, and field. This allows us to define what it means to say that R is a function *from* one set A *to* another set B, formally defined as follows.

(D8) R is a function *from* A *into* B $=_{df}$ $fun[R]$ & $dom(R)=A$ & $ran(R) \subseteq B$

Notice the contrast between '=' and '⊆'. By changing the latter to '=', we obtain a further definition.

(D9) R is function *from* A *onto* B $=_{df}$ $fun[R]$ & $dom(R)=A$ & $ran(R)=B$

3. Functions, Arguments, Values

There are various notational customs in mathematics and set theory in regard to functions. First of all, it is customary to use lower case letters (especially, 'f', 'g', and 'h') to denote functions. Secondly, it is customary to introduce function-argument-value notation, as follows.

(D10) $f(a) =_{df} \iota x[afx]$

Note that (D10) informally presumes that f is a function and that a is an element of $dom(f)$, in virtue of which there is a unique thing to which a is related by f. However, as far are the official definition is concerned, $f(a)$ may be ill-defined, in which case $f(a)$=⋇.

It is customary to read '$f(a)$' as "f of a". Further terminology: f is the *function*, a is the *argument*, $f(a)$ is the *value* of the function f *at* the argument a.

As an intuitive example, let f be the child/father relation; specifically, x bears f to y iff x is fathered by y (equivalently, y fathers x). First of all, f is a relation (let's say, among humans); further, if xfy & xfz, then $y=z$, so f is many-one. Each person is fathered (biologically at least) by exactly one person (although, of course, two people may be fathered by the same person). It follows that f is a function. Thus, we can introduce the notation '*the* father of x' denoted '$f(x)$', which stands for 'the unique person y such that x is fathered by y'.

Before proceeding, it is important to note the difference in logical form between the following expressions.

(e1) $\wp(a)$
(e2) $f(a)$

Whereas '℘' is a *function-sign*, 'f' is a *singular-term.* There is also a function-sign involved in the second expression, but it is suppressed. This is made more clear in the following forms.

(f1) $p(a)$
(f2) $r(f,a)$

These are in the language of pure first-order logic: '$p(.)$' is a function-sign, as is '$r(,)$', which is the hidden function-sign in (e2). We might read (f2) as follows.

(r2) the result of applying (function) f to (argument) a

4. Injections, Surjections, Bijections, Permutations

In this section, we discuss some notational conventions, and associated terminology, summarized as follows.

(D11) $f{:}A \to B$ $=_{df}$ f is a function from A into B

(D12) $f{:}A \to B(\text{onto})$ $=_{df}$ f is a function from A onto B

(D13) $f{:}A \to B(1\text{-}1)$ $=_{df}$ $f{:}A \to B$ & f is $1\text{-}1$

(D14) $f{:}A \leftrightarrows B$ $=_{df}$ $f{:}A \to B(\text{onto})$ & f is $1\text{-}1$.

The terminology associated with these ideas is as follows. A function from A onto B is sometimes called a *surjection*, whereas a $1\text{-}1$ function is sometimes called an *injection*. Functions that are both $1\text{-}1$ and onto are frequently called *bijections*.

Bijections are important because they form the foundation of the theory of cardinal numbers. By way of preview, we can define a relation $\approx$, called *equipollence* (also called equipotence), as follows.

(d) $A \approx B$ $=_{df}$ $\exists f\, [\, f{:}A \leftrightarrows B\,]$

In other words, two sets are equipollent iff there is a bijection between them. One can prove that this relation, suitably relativized, is an equivalence relation, which is to say the following theorems hold.

(t1) $A \approx A$
(t2) $A \approx B \;\to B \approx A$
(t3) $A \approx B$ & $B \approx C$ $.\!\to$ $A \approx C$

More about this in Chapter 5.

A bijection between a set A and itself is called a *permutation*.

(D15) f is a permutation $=_{df}$ $\exists X\, [\, f{:}X \leftrightarrows X\,]$

A trivial example of a permutation is the *identity function* on any given set A, defined as follows.

(D16) $I_A =_{df} \{\, (x,y) : x,y \in A \;\&\; x{=}y\, \}$

By contrast a *complete permutation* is a permutation that is *irreflexive* (recall definition).

(D17) f is a complete permutation $=_{df}$ f is a permutation & f is irreflexive.

Notice that, if f is a complete permutation, then $f(x){\neq}x$. Thus, whereas the identity function I_A maps every element of A to itself, a complete (irreflexive) permutation maps every element of A to something else.

5. Images, Pre-Images; Inversion, Composition, and Restriction

Functions are a species of relation, so the various notions that apply generally to relations apply also to functions. We have already discussed domain and range. In the present section, we discuss the remaining ones.

The image of a set under a *function* f may be defined in the usual manner, or it may be *specially* defined in either of the following ways.

(d1) $f^{\rightarrow}(A) =_{df} \{ y : \exists x(x \in A \ \&\ y = f(x) \}$
(d2) $f^{\rightarrow}(A) =_{df} \{ f(x) : x \in A \}$

Similarly, the pre-image of A under f may be *specially* defined in either of the following ways.

(d3) $f^{\leftarrow}(A) =_{df} \{ x : \exists y(y \in A \ \&\ y = f(x)\}$
(d4) $f^{\leftarrow}(A) =_{df} \{ x : f(x) \in A\}$

A function is a relation, and may be inverted to obtain another relation; however, the inverse of a function need not be a function. Indeed, we have the following theorems, in this regard.

(t1) fun[R^{-1}] $\leftrightarrow$ one-many[R]
(t2) fun[f] $\rightarrow$. fun[f^{-1}] $\leftrightarrow$ *1-1*[f]

Similarly, two functions can be composed in the manner specified for general relations, and the result is itself a function.

(t3) fun[f] & fun[g] .$\rightarrow$ fun[$f \circ g$]

The only hang-up is that the domain of the composite $f \circ g$ may be obscure in relation to the domains of f and g.

An alternative definition of function composition defines it *de novo*, on the basis of the following *desideratum*.

(D) $f \circ g(a) = f(g(a))$

This doesn't *define* the composite function, but only suggests what should be true of it. It requires that the result of applying the composite function $f \circ g$ to an argument a may be calculated by applying g to a, and then applying f to the result. To obtain the official definition, one must specify the domain. The usual practice is to restrict the above functional concept (see later section) to the set $g^{\leftarrow}(dom(f))$, which are the elements that g maps into the domain of f.

Finally, the restriction of a function to a set A is specially defined as follows.

(d5) $f|_A =_{df} \{ (x,y) : y = f(x) \ \&\ x \in A \}$

In other words, $f|_A$ is just like f except that its domain is restricted.

6. Sequences

Earlier we saw how one can define an ordered-pair as a special sort of unordered-pair. At that time, however, we refrained from officially defining other sorts of ordered aggregates (ordered triples, ordered quadruples, etc.), although we suggested something like the following as one possibility.

(d1) $(a,b) =_{df} \{\ \{a\},\{a,b\}\ \}$

(d2) $(a,b,c) =_{df} \{\ \{a\},\{a,b\},\{a,b,c\}\ \}$

(d3) $(a,b,c,d) =_{df} \{\ \{a\},\{a,b\},\{a,b,c\},\{a,b,c,d\}\ \}$

etc.

This series of definitions is fine as far as it goes, but it does not go as far as we might like. It would be nice to have a more complex sort of ordered set as well, including infinite sequences (of various sorts). For this we need a more general organizational technique.

One such technique employs the notion of *sequence*, which is a special case of the more general notion of *family*, to be discussed in the next section.

The definition of sequence depends upon the notion of *natural number*. We do not formally define the natural numbers until the next chapter. For now, we use the numbers in a purely intuitive way. In particular, we suppose there are the individual numbers – 0, 1, 2, 3, etc., and we suppose further that there is a set that contains all the numbers. (As we see later, the latter requires a further axiom.)

Having the natural numbers and the set of all natural numbers, in addition to the notion of function, we can provide the following series of definitions.

(D18.1) $\langle a\rangle =_{df} \{\ (1,a)\ \}$

(D18.2) $\langle a,b\rangle =_{df} \{\ (1,a), (2,b)\ \}$

(D18.3) $\langle a,b,c\rangle =_{df} \{\ (1,a), (2,b), (3,c)\ \}$

etc.

The following theorems about these objects could serve as alternative definitions.

(t1.1) $\langle a\rangle = \iota f\,[\ dom(f)=\{1\}\ \&\ f(1)=a\ ]$

(t1.2) $\langle a,b\rangle = \iota f\,[\ dom(f)=\{1,2\}\ \&\ f(1)=a\ \&\ f(2)=b\ ]$

(t1.3) $\langle a,b,c\rangle = \iota f\,[\ dom(f)=\{1,2,3\}\ \&\ f(1)=a\ \&\ f(2)=b\ \&\ f(3)=c\ ]$

etc.

The entities so-defined are called (ordered) *n-tuples*, also *finite sequences*. Notice that every such entity is a *function* whose domain is some initial segment of positive integers (also to be officially defined later). This leads to the following series of definitions.

(d4.1) A 1-tuple is any function whose domain is $\{1\}$

(d4.2) A 2-tuple is any function whose domain is $\{1,2\}$

(d4.3) A 3-tuple is any function whose domain is $\{1,2,3\}$

etc.

(d5) A *finite-sequence* is an n-tuple for some number n.

(d6) An *infinite-sequence* is any function whose domain is the set of all positive integers.

Finite sequences have both intended and unintended (or at least, surprising) theorems about them. The intended theorems are the expected analogs to the Principle of Ordered-pairs.

(t2.1) $\langle a\rangle = \langle b\rangle \leftrightarrow a{=}b$

(t2.2) $\langle a,b\rangle = \langle p,q\rangle \leftrightarrow. \ a{=}p \ \& \ b{=}q$

(t3.3) $\langle a,b,c\rangle = \langle p,q,r\rangle \leftrightarrow. \ a{=}p \ \& \ b{=}q \ \& \ c{=}r$

etc.

As should be expected, 2-tuples are structurally analogous to ordered-pairs; in particular, they have a first and second component. Although 2-tuples and ordered-pairs are structurally equivalent, they are not "materially identical" (The elements of a set constitute its "matter"; extensionality says that materially identical sets are numerically identical.) In particular, we have the following theorem.

(t4) $\forall x \forall y \,[\, \langle x,y\rangle \neq (x,y) \,]$

A sequence is a function, so it can be meaningfully applied to any object in its domain. This gives us the following mildly surprising, but instructive, results.

(t5.1) $\langle a,b\rangle(1) = a$; $\langle a,b\rangle(2) = b$

(t5.2) $\langle a,b,c\rangle(1) = a$
$\langle a,b,c\rangle(2) = b$
$\langle a,b,c\rangle(3) = c$
etc.

The notation seems odd at first sight, but notice that it is just a special case of the general function-argument notation – $f(a)$. Indeed, the way '$f(a)$' is defined, 'f' can be *any* set-denoting expression. The *intended application* of the definition, of course, is when f is a function and a is in its domain. But a sequence is, by definition, a function, so it can be meaningfully applied to an argument from its domain.

7. Subscript Notation

Having defined sequences, we next turn to the rigorous presentation of a notational device that is quite common in mathematics and logic – subscripting. First, we point out that one use of subscripts is *purely metalinguistic*. For example, we can subscript variables and constants simply in order to create more variables and constants. The following illustrates this usage.

$$\forall x_1 \forall x_2 \,[\, x_1 + x_2 = x_2 + x_1]$$

The numerals '1' and '2' do not occur *essentially* in this formula; they are not categorimatic; they are not semantically significant. We could equally well choose some other form of decoration in order to distinguish the two variables from one another.

In this section, we wish to discuss how subscripts can be used in a way in which the numerals are significant. Let us start with an example. One way to write sequence-denoting expressions is as follows.

$\langle a_1, a_2 \rangle$
$\langle a_1, a_2, a_3 \rangle$
etc.

What is the set-theoretic significance of the subscript notation? Well, we know from the earlier definition that $\langle a_1, a_2 \rangle$ is the function (call it f, for short) with the following properties.

$dom(f) = \{1,2\}$
$f(1) = a_1$
$f(2) = a_2$

Now, it is customary in mathematics to treat subscript notation as simply a variant of function-argument notation. Thus, we have the following official definition.

(D19) $f_i \quad =_{\text{df}} \quad f(i)$

This means, in particular, that

$$\langle a_1,a_2 \rangle \quad =_{\text{df}} \quad \langle\, a(1), a(2)\, \rangle$$

In order for this to be consistent with already existing notation, we must suppose that a is itself a function intimately related to the function $\langle a_1,a_2 \rangle$; in particular:

a restricted to $\{1,2\}$ is identical to $\langle a_1,a_2 \rangle$

The most economical way to accomplish this is simply to say:

$$a = \langle a_1,a_2 \rangle$$

So in the final analysis, the notation

$$\langle a_1,a_2 \rangle$$

is referentially convoluted; it is an expression that denotes a function but which contains two occurrences of another expression that denotes that every same function. This is grammatically coherent; an example of a similar construction in ordinary language is 'the oldest son of the father of Cain'.

Once we have both subscript-notation and sequence-notation, we can write expressions like the following.

$$\langle\, a_2, a_1\, \rangle$$

This looks incoherent; in particular, it looks like we are saying that the second component is the first component and the first component is the second component. Well, yes and no. For, actually the above expression contains reference to *two* different sequences (in principle, at least). There is the sequence $\langle a_2,a_1 \rangle$, and there is also the sequence a. Relative to sequence a, a_1 is first, and a_2 is second; but relative to sequence $\langle a_2,a_1 \rangle$, the order is reversed.

This is not as bizarre as it might seem at first. Suppose the sequence a is the alphabet in its conventional order – a_1 = 'a', a_2 = 'b', etc. Then the sequence $\langle a_2,a_1 \rangle$ is the sequence whose first component is the second letter of the alphabet, and whose second component is the first letter of the alphabet – i.e., the word 'ba'.

8. Families

Sequences are a species of a more general notion, called *family*. A family is (materially if not functionally) a function from one set, called the *index set*, usually denoted I, to another set S, called the *indexed-set*. This is fairly standard terminology. In the next section, however, I propose to use the expression 'indexed-set' in a special way, different from the official terminology, but nonetheless in keeping with mathematical usage.

Notice that a sequence is simply a family in which the index set I is some initial segment of positive integers.

The notation for families varies from author to author. Halmos (*Naive Set Theory*, a standard), for example, writes the following.

$$\{a_i\}(i \in I)$$
$$\{a_i\}$$

The former expression is official; the latter expression is used in place of the official expression when the indexing set I is tacitly understood. It has no official definition, however. Indeed, in its bare form, the notation is inconsistent with the notation for singletons, which is already entrenched in set theory. So it cannot be officially defined. It must rather be regarded as introduced in a purely informal way, without a corresponding official definition.

Whereas the shorthand expression conflicts with prior notation, the official expression does not. Its logical form is obscure, perhaps, involving as it does the bound variable 'i'. Like set-abstract notation, it involves some sort of variable-binding operator not countenanced by ordinary first-order logic. Its official definition goes as follows.

(d1) $\{a_i\}(i \in I) \quad =_{df} \quad \{ (i,a_i) : i \in I \}$

In other words, the expression refers to a function (alternatively named a), with domain I, which assigns the value a_i to each argument (index) i in I.

One occasionally sees the following notation for families.

$$\{a_i : i \in I\}$$

This conflicts with other notation, however. Using already existing notation one can in fact prove the following

(t1) $\{a_i : i \in I\} = ran(\{a_i\}(i \in I))$

which is analogous to the following theorem about functions.

(t2) $\{ f(x) : x \in dom(f) \} = ran(f)$

In other words, $\{a_i : i \in I\}$ is not the family but its range.

Since the above notation conflicts with other notation, we adopt similar-looking but non-conflicting notation for families, as follows.

(D20) $\langle a_i : i \in I \rangle \quad =_{df} \quad \{ (i,a_i) : i \in I \}$

The following theorem could serve as an alternative definition.

(t3) $\langle a_i : i \in I \rangle \quad =_{df} \quad \iota f\, [\, dom(f)=I \;\&\; \forall i(i \in I \rightarrow f(i)=a_i)\,]$

Notice, of course, that

$$\langle a_i : i \in I \rangle \;=\; \{a_i\}(i \in I)$$

It is also convenient to drop reference to I when it is understood from the context, in which case we simply write

$$\langle a_i \rangle$$

We cannot officially define the latter, however, since it conflicts with our notation for 1-tuples. We seldom have use for 1-tuples, so the conflict seldom actually arises.

The "practical" individuation of families is similar to that for sequences, which in turn is similar to that for ordered-pairs. In particular, we have the following analog of the Principle of Ordered-pairs.

(t4) $\langle a_i : i \in I \rangle = \langle b_i : i \in J \rangle \leftrightarrow.\; I=J \;\&\; \forall i(i \in I \rightarrow a_i=b_i)$

In other words, two families a and b are identical if and only if they have the same domain and they assign the same thing to each element of that domain.

9. Indexed-sets

The notion of family is used in mathematics in two different ways. In order to reflect these two uses, I propose to use the words ‘sequence’ and ‘indexed-set’, and I propose to use the word ‘family’ as the *genus* term.

The difference between the two concepts rests upon the issue of whether repetitions are allowed or disallowed. Sometimes we want to allow repetition. For example, in describing words as sequences of letters, unless we allow repetition, we cannot describe words in which the same letter occurs twice or more.

Sometimes, however, we do not wish to allow repetitions. An everyday example of a sequence in which repetition is undesirable is house numbering. An address system that assigns the same house two different addresses (so for example, the Smith house is both house number 37 and house number 49) is unacceptable. Similarly, the same address should not be assigned to two different houses.

I propose to use the term ‘indexed-set’ for this particular use of family. I *unofficially* define it as follows.

(d) An *indexed-set* is (unofficially) an ordered-triple (A, L, f), consisting of a set A (to be labeled), a set L of labels, and a 1-1 function f that assigns a (unique) label to each element of A.

Thus, an indexed-set contains a set to be labeled, a set of labels, and a pairing of objects and their labels. The following is an example:

A = the set of houses on Main Street Anytown USA
L = the set of possible addresses
f = the function that pairs each house with its address

What I wish to show now is that every indexed-set corresponds to a family, in such a way that it is natural simply to define indexed-sets as special sorts of families. In order to see this, for any given indexed-set (as defined above), we construct the associated family as follows.

(R) Let (A, L, f) be an indexed-set as defined above. Then the "reconstruction" of (A, L, f) is the family $\langle a_i : i \in I \rangle$ defined so that:
(1) $a_i = f^{-1}(i)$
(2) $I = ran(f)$

In other words, the family (function) is the inverse of the labeling function (remember f is 1-1). This corresponds to the following idea in connection with house numbering.

the Smith-house is house number i if and only if

i is the label assigned to the Smith-house.

$s = h_i \leftrightarrow f(s) = i$

Having given the unofficial intuitive definition, and having given its reconstruction, we now give the official definition, which is much simpler.

(d1) An indexed-set is, by definition any 1-1 family.

In other words, an indexed-set is a family in which no term is repeated.

If we want to reconstitute an indexed-set (A,L,f) from the corresponding 1-1 family, we follow the following recipe.

(r) Given 1-1 family $\langle a_i : i \in I \rangle$, the associated indexed-set (A,L,f) is defined as follows.

(1) $A = ran(a)$
(2) $L = I$
(3) $f(x) = \iota i\,[\, a_i = x \,]$

10. Generalized Union and Intersection Revisited

The chief use of indexed-sets (1-1 families) is organizational. We begin with a set H, we choose an indexing set I (labels) of the appropriate size, then we define a 1-1 function h from I onto H, so that:

$H = \{h_i : i \in I\}$

The original set H itself has not changed. How could it? We have simply *redescribed* it as the range of the function h. We have organized H in our own minds, by affixing labels to its elements.

Re-describing a set H as the range of some function is occasionally quite useful. For example, it can be used in describing generalized union and intersection. Recall the official definitions.

(d1) $\bigcap\mathbb{C} =_{df} \{ x : \forall Y(Y\in\mathbb{C} \rightarrow x\in Y) \}$
(d2) $\bigcup\mathbb{C} =_{df} \{ x : \exists Y(Y\in\mathbb{C} \ \&\ x\in Y) \}$

Now, $\mathbb{C}$ can be redescribed so that $\mathbb{C} = \{A_i : i\in I\}$, for some family $\langle A_i : i\in I\rangle$. At the very least, we can let I be $\mathbb{C}$ itself, and we can let the indexing function be the identity function on I. In any case, we can rewrite the intersection and union expressions as follows.

$\bigcap\mathbb{C} = \bigcap\{A_i : i\in I\}$
$\bigcup\mathbb{C} = \bigcup\{A_i : i\in I\}$

These in turn lead us to the following notation and definitions.

(D21.1) $\bigcap_i A_i =_{df} \bigcap(\ ran(A)\ )$
(D21.2) $\bigcap_{i\in I} A_i =_{df} \bigcap\{A_i : i\in I\}$
(D22.1) $\bigcup_i A_i =_{df} \bigcup(\ ran(A)\ )$
(D22.2) $\bigcup_{i\in I} A_i =_{df} \bigcup\{A_i : i\in I\}$

The following facts summarize the relation between the two expressions.

(t1) $ran(A) = \{ A_i : i \in dom(A) \}$
(t2) $dom(A) = I \rightarrow ran(A) = \{A_i : i\in I\}$
(t3) $dom(A) = I \rightarrow \bigcap_i A_i = \bigcap\{A_i : i\in I\}$
(t4) $dom(A) = I \rightarrow \bigcup_i A_i = \bigcup\{A_i : i\in I\}$

There are a few useful theorems (exercises) for dealing with the union or intersection of a *family of sets*, as follows, which presume $I = dom(A)$.

(t5) $x \in \bigcup_i A_i \leftrightarrow \exists i(\ i\in I \ \&\ x\in A_i\ )$
(t6) $x \in \bigcap_i A_i \leftrightarrow \forall i(\ i\in I \rightarrow x\in A_i\ )$

11. Generalized Cartesian-Products

Generalized union and intersection are the infinitary counterparts of binary union and intersection, respectively. Along quite similar lines, one can construct an infinitary counterpart of the binary Cartesian-product.

The binary Cartesian-product is officially defined as follows.

(d1) $A\times B =_{df} \{ (x,y) : x\in A \ \&\ y\in B \}$

If we wanted to take the Cartesian-product of a triple of sets A,B,C, we might define it in a number of different ways.

(d2) $\times(A,B,C) =_{df} (A\times B)\times C$
(d3) $\times(A,B,C) =_{df} A\times(B\times C)$
(d4) $\times(A,B,C) =_{df} \{ (x,y,z) : x\in A \ \&\ y\in B \ \&\ z\in C\}$,

Notice that these definitions do not agree with one another.

Having selected a definition for 3-way Cartesian-product, we still have to give a definition for 4-way Cartesian-product, etc. Fortunately, however, we don't have to do this. Now that we have the notions of sequence and family, we do not need to give individual definitions, but rather can define generalized Cartesian-product in one fell swoop.

Nonetheless, rather than jump right in, we begin by writing down a few special cases of the general definition, cases in which the family is a finite sequence.

(D23.1) $\times\langle A,B\rangle =_{df} \{ \langle x,y\rangle : x\in A \;\&\; y\in B \}$
(D23.2) $\times\langle A,B,C\rangle =_{df} \{ \langle x,y,z\rangle : x\in A \;\&\; y\in B \;\&\; z\in C \}$
etc.

Notice that an element of the Cartesian-product of a 2-tuple $\langle A,B\rangle$ is any 2-tuple whose first term is an element of A and whose second term is an element of B. Similarly, an element of the Cartesian-product of the 3-tuple $\langle A,B,C\rangle$ is any 3-tuple whose first term is in A, whose second term is in B, and whose third term is in C.

More generally, an element of the Cartesian-product of a sequence of sets is any sequence whose respective terms are elements of the respective sets.

An everyday analogy might be useful. In selecting a 4-course meal from a menu, one selects a first, a second, a third, and a fourth course. The menu basically tells you what the possible choices are for each course. A menu of four-course meals might even be regarded as a 4-tuple of possible choices, for example:

⟨soups, salads , entrees , desserts⟩

There are four sets of choices. Each combination of choices corresponds to one element of the Cartesian-product of the above 4-tuple. Each combination of choices corresponds to a possible meal-type, relative to menu. I say ‘type’, because, two people can have the same meal-type, but they cannot have the same meal, unless they share.

Also note that we need not assume that the sets are mutually-disjoint. In the restaurant example, it might happen that a particular fruit salad counts both as a salad and as a dessert.

We now turn to the general definition of Cartesian-product, which subsumes the earlier definitions.

(D24.1) $\times\langle A_i : i\in I\rangle =_{df} \{ \langle a_i : i \in I\rangle : \forall i(i\in I \rightarrow a_i\in A_i \}$
(D24.2) $\times_i A_i =_{df} \times\langle A_i : i \in dom(A) \rangle$

In other words, something is an element of the Cartesian-product of a family of sets, indexed by I, if and only if it is a family, also indexed by I, whose i-th term, a_i, is an element of the i-th set, A_i, for every index point i.

12. Cartesian-Exponentiation

In arithmetic, the notion of multiplication gives rise to a derivative notion – exponentiation. In particular, when we multiply a number m by itself n times, we obtain m raised to the n-th power, which is written m^n.

Cartesian-exponentiation derives from Cartesian-multiplication in a similar manner. In particular, we can talk about the 1st, 2nd, 3rd, etc., *Cartesian-power* of a set, which are defined as follows.

(D25.1) $A^2 =_{df} \times\langle A,A\rangle$

(D25.2) $A^3 =_{df} \times\langle A,A,A\rangle$

etc.

For example, $\langle A,A,A\rangle$ is a 3-tuple every term of which is identical. So an element of $\times\langle A,A,A\rangle$ is a 3-tuple every term of which is an element of A.

A little terminology is useful at this point. We begin with an official definition of the expression 'term'.

(d1) A *term* of a family f is any element of $ran(f)$.

The relation between term and family is analogous to the relation between element and set: sets are individuated by their elements; families are individuated by their terms (including their order).

(d2) A family of so-and-so's is a family every term of which is a so-and-so.

So, for example, a family of non-empty sets is a family every term of which is a non-empty set.

N-tuples are a species of family, so the above definition can be specialized to n-tuples. The following is a particular application.

(d3) An n-tuple of elements of A is an n-tuple every term of which is an element of A.

This allows us to define Cartesian-exponentiation as follows.

(d4) $A^n =_{df} \{ t : t$ is an n-tuple of elements of $A\}$

The following theorem shows that Cartesian-exponentiation is a special case of the Cartesian-product.

(t1) A^n = the Cartesian-product of the family whose domain is $\{1,2,...,n\}$ and whose range is $\{A\}$.

In the move from arithmetic to analysis, exponentiation is generalized so that r^s is defined for any real numbers r and s, even irrational numbers. The same thing happens in set theory. In particular, by dropping the requirement that the exponent be a positive integer, we can generalize the notion of Cartesian-exponentiation (power) to arbitrary sets, as follows.

(D26) $A^I =_{df} \{ f : f{:}I\rightarrow A \}$

In other words, A^I is the set of all functions from I into A. That the notion, so defined, is indeed a species of Cartesian-product is seen in the following theorem.

(t2) A^I = the Cartesian-product of the family whose domain is I and whose range is $\{A\}$.

Unfortunately, the notation ⌜ A^I ⌝ conflicts with the notation ⌜ A^n ⌝, for the usual definition of numbers (see chapter on numbers). This conflict seldom causes any difficulty, however. The difference is between, for example, families with indexing set {0,1,2} and families with indexing set {1,2,3}.

Cartesian-exponentiation is useful in model theory (metalogic), through the following definitions.

(d5) Let A be any set, and let n be any positive integer. Then an *n-place relation* on A is, by definition, any subset of A^n.

(d5) Let A be any set, and let n be any positive integer. Then an *n-place function* on A is, by definition, any function from A^n into A.

In conventional first-order model theory, an interpretation-structure consists of a set D and an interpretation function I. The set D is the *domain of interpretation* (what the quantifiers range over). The interpretation function assigns an extension to every non-logical symbol; in particular, it assigns to every n-place predicate symbol an n-place relation on D, and it assigns to every n-place *function-sign* an n-place *function* on D.

As usual, we have a slight discrepancy (but not contradiction) with earlier terminology. A two-place relation on A *corresponds to* a binary relation on A , but it is not *identical to* any binary relation on A! This is because a two-place relation consists of 2-tuples, whereas a binary relation consists of ordered-pairs. Otherwise, however, they are pragmatically interchangeable. Also a 1-place function on A *corresponds to* a function on A, but it is not *identical to* any function on A. This is because a 1-tuple $\langle a \rangle$ corresponds to a, but is not identical to a.

13. The Axiom of Choice

Intimately related to the notion of Cartesian-product is the notion of *choice function*, which is officially defined as follows.

(D27.1) Let $\mathbb{C}$ be a collection of sets. Then a *choice function* on $\mathbb{C}$ is, by definition, any function f with the following properties.

(1) $dom(f) = \mathbb{C}$
(2) $\forall X(X \in \mathbb{C} \rightarrow f(X) \in X)$

(D27.2) $ch(\mathbb{C}) =_{df} \{ f : f \text{ is a choice function on } \mathbb{C}\}$

In other words, a choice function selects (chooses) from each set X in $\mathbb{C}$ an element of X.

Interestingly enough, every collection of choice functions is a Cartesian-product (the converse is not true). In other words, we have the following theorem.

(t1) for any $\mathbb{C}$, there is a family $\langle A_i : i \in I \rangle$ such that
$ch(\mathbb{C}) = \times\langle A_i : i \in I \rangle$

To see this, let $\mathbb{C}$ be any collection of sets. We need to construct a corresponding family $\langle A_i\!: i \in I\rangle$ such that ch($\mathbb{C}$) = $\times\langle A_i\!: i \in I\rangle$. Well, let I=$\mathbb{C}$, and let A_i=i for all i in I. Now consider the Cartesian-product of the resulting family – $\times_i A_i$. Consider an arbitrary element, f. It is a function from I (i.e., $\mathbb{C}$) such that $f(i) \in A_i$. But A_i=i, so $f(i) \in i$. In other words, f is a choice function on $\mathbb{C}$. The converse direction is argued in a similar manner.

Defining the set of choice functions on a collection $\mathbb{C}$ in no way ensures that such sets are legitimate. A number of problems can arise. If $\mathbb{C}$ is empty, then there are no choice functions on $\mathbb{C}$. Also, if one of the sets in $\mathbb{C}$ is empty, then there are no choice functions on $\mathbb{C}$.

But suppose that $\mathbb{C}$ is non-empty, and suppose every element of $\mathbb{C}$ is non-empty; i.e., $\mathbb{C}$ is a non-empty collection of non-empty sets. Can one prove the existence of at least one choice function on $\mathbb{C}$. If $\mathbb{C}$ is finite, then the answer is "yes", but if $\mathbb{C}$ is infinite, then the answer is "no".

When faced with the inability to prove the existence of a desirable set, the solution, as usual, is to add a further axiom. In this particular case, the additional axiom is called the Axiom of Choice. This axiom has many different formulations. The formulation I especially like uses the notion of Cartesian-product.

(AC) The Cartesian-product of any non-empty family of non-empty sets is non-empty.

A brief explanation of the terminology. A non-empty family is a family whose domain is non-empty. A family of so-and-so's is a family every term of which is a so-and-so (i.e., every element of the range is a so-and-so). Accordingly a family of non-empty sets is a family every term of which is a non-empty set.

It is not hard to prove that (AC) is equivalent to the following alternative formulation (AC*).

(AC*) If $\mathbb{C}$ is a non-empty collection of non-empty sets, then there exists at least one function f such that for every X in $\mathbb{C}$, $f(X) \in X$.

In other words,

$$\mathbb{C}\neq\varnothing \;\;\&\;\; \forall X(X\in\mathbb{C} \rightarrow X\neq\varnothing) \;\; .\!\rightarrow\;\; ch(\mathbb{C}) \neq \varnothing.$$

It is somewhat harder to prove the following alternative version.

(AC**) Every relation R includes at least one function with the same domain as R.

Other famous equivalents of the Axiom of Choice are Zorn's Lemma, and the Well-Ordering Theorem, given as follows.

(ZL) A partially ordered set has a maximal element if every linearly ordered subset has an upper bound.

(WO) Every set is the field of some well-ordering relation.

See later chapter on well-ordering relations.

14. Definitions for Chapter 3

1.	div$[R]$	$=_{df}$	$\exists xyz[xRy\ \&\ xRz\ \&\ y\neq z]$	[divergent]
2.	con$[R]$	$=_{df}$	$\exists xyz[xRz\ \&\ yRz\ \&\ x\neq y]$	[convergent]
3.	$m\text{-}o[R]$	$=_{df}$	$\sim$div$[R]$	[many-one]
4.	$o\text{-}m[R]$	$=_{df}$	$\sim$con$[R]$	[one-many]
5.	$m\text{-}m[R]$	$=_{df}$	div$[R]$ & con$[R]$	[many-many]
6.	$1\text{-}1[R]$	$=_{df}$	$m\text{-}o[R]\ \&\ o\text{-}m[R]$	[one-one]
7.	fun$[A]$	$=_{df}$	$\mathcal{R}[A]\ \&\ m\text{-}o[A]$	[function]
8.	see #11			
9.	see #13			
10.	$f(a)$	$=_{df}$	$\iota x[\ afx\ ]$	[function-argument-value]
11.	$f{:}A\to B$	$=_{df}$	fun$[f]$ & $dom(f)=A$ & $ran(f)\subseteq B$	[into]
12.	$f{:}A\to B(1\text{-}1)$	$=_{df}$	$f{:}A\to B\ \&\ 1\text{-}1(f)$	[one-one]
13.	$f{:}A\to B$(onto)	$=_{df}$	$f{:}A\to B\ \&\ ran(f)=B$	[onto]
14.	$f{:}A\leftrightarrows B$	$=_{df}$	$f{:}A\to B(1\text{-}1)\ \&\ f{:}A\to B$(onto)	[bijection]
15.	perm$[f]$	$=_{df}$	$\exists X[f{:}X\leftrightarrows X]$	[permutation]
16.	I_A	$=_{df}$	$\{\ (x,y) : x,y\in A\ \&\ x=y\ \}$	[identity function on A]
17.	comperm$[f]$	$=_{df}$	perm$[f]$ & irr$[f]$	[complete permutation]
18.	$\langle a\rangle$	$=_{df}$	$\{(1,a)\}$	[1-tuple]
	$\langle a,b\rangle$	$=_{df}$	$\{(1,a), (2,b)\}$	[2-tuple]
	$\langle a,b,c\rangle$	$=_{df}$	$\{(1,a), (2,b), (3,c)\}$	[3-tuple]
	etc.			
19.	f_i	$=_{df}$	$f(i)$	[subscript notation]
20.	$\langle a_i : i\in I\rangle$	$=_{df}$	$\{\ (i,a_i): i\in I\ \}$	[family]
21.	$\bigcap_i A_i$	$=_{df}$	$\bigcap(\ ran(A)\ )$	[generalized intersection-1]
	$\bigcap_{i\in I} A_i$	$=_{df}$	$\bigcap\{A_i : i\in I\}$	[generalized intersection-2]
22.	$\bigcup_i A_i$	$=_{df}$	$\bigcup(ran(A))$	[generalized union-1]
	$\bigcup_{i\in I} A_i$	$=_{df}$	$\bigcup\{A_i : i\in I\}$	[generalized union-2]
23.	$\times\langle A,B\rangle$	$=_{df}$	$\{\ \langle x,y\rangle : x\in A\ \&\ y\in B\ \}$	[Cartesian-product]
	$\times\langle A,B,C\rangle$	$=_{df}$	$\{\ \langle x,y,z\rangle : x\in A\ \&\ y\in B\ \&\ z\in C\ \}$	
	etc.			
24.	$\times\langle A_i : i\in I\rangle$	$=_{df}$	$\{\ \langle a_i{:}i\in I\rangle : \forall i(i\in I\rightarrow a_i\in A_i)\ \}$	[General Cartesian-product]
25.	A^2	$=_{df}$	$\times\langle A,A\rangle$	[Cartesian power]
	A^3	$=_{df}$	$\times\langle A,A,A\rangle$	
	etc.			
26.	A^I	$=_{df}$	$\{\ f : f{:}I\to A\}$	[General Cartesian power]
27.	ch$[f,\mathbb{C}]$	$=_{df}$	fun$[f]$ & $dom(f)=\mathbb{C}$ & $\forall X(X\in\mathbb{C}\rightarrow f(X)\in X)$	[choice function]
	ch$(\mathbb{C})$	$=_{df}$	$\{\ f$: ch$[f,\mathbb{C}]\ \}$	