Bourbaki

Contents

1 Sets

1.1 Why

We want to speak of a collection of objects, pre-specified or possessing some similar defining property.

1.2 Definition

A **set** is a collection of objects. We use the term object in the usual sense of the English language. So a set is itself an object, but of the peculiar nature that it contains other objects. In thinking of a set, then, we regularly consider the objects it contains. We call the objects contained in a set the **members** or **elements** of the set. So we say that an object contained in a set is a **member of** or an **element of** the set.

1.2.1 Notation

We denote sets by upper case latin letters: for example, A, B, and C. We denote elements of sets by lower case latin letters: for example, a, b, and c. We denote that an object A is an element of a set A by $a \in A$. We read the notation $a \in A$ aloud as "a in A." The $\in$ is a stylized ϵ, which possesses the mnemonic for element. We write $a \notin A$, read aloud as "a not in A," if a is not an element of A.

If we can write down the elements of A, we do so using a brace notation. For example, if the set A is such that it contains only the elements a, b, c, we denote A by $\{a, b, c\}$. If the elements of a set are well-known we introduce

the set in English and name it; often we select the name mnemonically. For example, let L be the set of latin letters.

If the elements of a set are such that they satisfy some common condition, we use the braces and include the condition. For example, if V is the set of vowels we denote V by $\{l \in L \mid l \text{ is a vowel}\}$. The $\mid$ is read aloud as "such that," the notation reads aloud as "l in L such that l is a vowel." We call the notation $\{l \in L \mid l \text{ is a vowel}\}$ **set-builder notation**. Set-builder notation is indispensable for sets defiined implicitly by some condition. Here we could have alternatively denoted V by $\{$"a", "e", "i", "o", "u"$\}$. We prefer the former, slighly more concise notation.

1.3 Two Sets

If every element of a first set is also an element of second set, we say that the first set is a **subset** of or is **contained in** the second set. Conversely, we say that the second set is a **superset** of or **contains** the first set. If a first set is a subset of a second set and the second set is a subset of the first set, we say the two sets are **equal**.

We call the set of subsets of a set A the **powerset** of A. We call the set which has no members the **empty set**. The empty set is contained in every other set.

1.3.1 Notation

Let A and B be sets. We denote that A is a subset of B by $A \subset B$. We read the notation $A \subset B$ aloud as "A subset B". We denote that A is equal to B by $A = B$. We read the notation $A = B$ aloud as "A equals B". We denote the empty set by $\varnothing$, read aloud as "empty." We denote the power set of A by 2^A, read aloud as "two to the A."

2 Cartesian Products

2.1 Why

We want to handle objects composed of elements from different sets.

2.2 Definition

Let A and B be sets. Construct a new set that is the ordered pairs of elements of A and B: the first element of the pair is an element of A and the second element of the pair is an element of B. The **cartesian product** of A with B is the set of all such ordered pairs.

We call an ordered pair **tuple**. Two tuples are equal if they have equal elements in the same order. Because of the ordering, if $A \neq B$, the cartesian product of A with B is not the same as the cartesian product of B with A. Only in the case that $A = B$ does this symmetry hold.

2.2.1 Notation

For sets A and B we denote the cartesian product of A with B by $A \times B$. We read the notation $A \times B$ as "A cross B." We denote elements of $A \times B$ by (a, b) with the understanding that $a \in A$ and $b \in B$. In this notation, we can write the observation that $A \times B \neq B \times A$, unless $A = B$.

3 Relations

3.1 Why

We want to relate elements of two sets.

3.2 Definition

A **relation** between two non-empty sets A and B is a subset of $A \times B$. So then, naturally, a relation on a single set C is a subset of $C \times C$.

3.2.1 Notation

As relations are sets, our de facto protocol is to denote them by upper case capital letters, for example, the letter R. Let R a relation on A and B. If $(a, b) \in R$, we often write aRb, read aloud as "a in relation R to b."

In many cases, though, we forego the set notation and use particular symbols. Often the symbols we use are meant to be suggestive of the relation. Some examples include $\sim$, $=$, $<$, $\leq$, $\prec$, and $\preceq$.

3.3 Properties

Let R a relation on a non-empty set A. R is **reflexive** if $(a, a) \in R$ for all $a \in A$. R is **transitive** if $(a, b) \in R \wedge (b, c) \in R \implies (a, c) \in R$ for all $a, b, c \in A$. R is **symmetric** if $(a, b) \in R \implies (b, a) \in R$ for all $a, b \in A$. R is **anti-symmetric** if $(a, b) \in R \implies (b, a) \notin R$ for all $a, b \in A$.

4 Functions

4.1 Why

We want a notion for a correspondence between two sets.

4.2 Definition

To each element of a first set we associate an element of a second set. We call this correspondence a **function**. We call the first set the **domain** and the second set the **codomain**. We say that the function **maps** elements from the domain into the codomain.

We call the codomain element associated with the domain element the **result** of **applying** the function to the domain element. We call the subset of ordered pairs whose first element is in the domain and whose second element is the corresponding result the **graph** of the function. The graph is a relation between the domain and codomain. So a function can be viewed as or specified as a relation between these two sets.

4.2.1 Notation

We often denote functions by lower case latin letters, especially f, g, and h. Of course, f is a mnemonic for function; g and h follow f in the alphabet.

Let A and B be two non-empty sets. When we want to be explicit that the domain of a function f is A and its codomain is B we write $f : A \to B$, read aloud as "f from A to B." For each element a in the domain, we denote the result of applying f to a by $f(a)$, read aloud "f of a." We sometimes drop the parentheses, and write the result as f_a, read aloud as "f sub a."

The set $\{(a, f(a)) \in A \times B \mid a \in A\}$ of ordered pairs is the graph of f. We often denote it by Γ_f; "gamma" is a mnemonic for graph.

Let $g : A \times B \to C$. We often write $g(a, b)$ or g_{ab} instead of $g((a, b))$. We read $g(a, b)$ aloud as "g of a and b". We read g_{ab} aloud as "g sub a b."

4.3 Properties

Let $f : A \to B$. The **image** of a set $C \subset A$ is the set $\{f(c) \in B \mid c \in C\}$. The **range** of f is the image of the domain. The **inverse image** of a set $D \subset B$ is the set $\{a \in A \mid f(a) \in B\}$.

The range need not equal the codomain; though it, like every other image, is a subset of the codomain. If the range and codomain are equal, we call the function **onto**. We say that the function maps the domain onto the range. This language suggests that every element of the codomain is used by f. It means that for each element b of the codomain, we can find an element a of the domain so that $f(a) = b$.

An element of the codomain may be the result of several elements of the domain. This overlapping, using an element of the codomain more than once, is a regular occurrence. If a function is a unique correspondence in that every domain element has a different result, we call it **one-to-one**. This language is meant to suggest that each element of the domain corresponds to one and exactly one element of the codomain, and vice versa.

4.3.1 Notation

Let $f : A \to B$. We denote the image of $C \subset A$ by $f(C)$, read aloud as "f of C." This notation is overloaded: for $c \in C$, $f(c) \in A$, whereas $f(C) \subset A$. Read aloud, the two are indistinguishable, so we must be careful to specify whether we mean an element c or a set C. The property that f is onto can be written succintly as $f(A) = B$. We denote the inverse image of $D \subset B$ by $f^{-1}(D)$, read aloud as "f inverse D."

5 Order Relations

5.1 Why

We want to handle elements of a set in a particular order.

5.2 Definition

Let R be a relation on a non-empty set A. R is a **partial order** if it is reflexive, transitive, and anti-symmetric.

A **partially ordered set** is a set and a partial order. The language partial is meant to suggest that two elements need not be comparable. For example, suppose R is $\{(a, a) \mid a \in A\}$; we may justifiably call this no order at all and call A totally unordered, but it is a partial order by our definition.

Often we want all elements of the set A to be comparable. We call R **connexive** if for all $a, b \in A$, $(a, b) \in R$ or $(b, a) \in R$. If R is a partial order and connexive, we call it a **total order**.

A **totally ordered set** is a set together with a total order. The language is a faithful guide: we can compare any two elements. Still, we prefer one word to three, and so we will use the shorter term **chain** for a totally ordered set; other terms include **simply ordered set** and **linearly ordered set**.

5.2.1 Notation

We denote total and partial orders on a set A by $\preceq$. We read $\preceq$ aloud as "precedes or equal to" and so read $a \preceq b$ aloud as "a precedes or is equal to b." If $a \preceq b$ but $a \neq b$, we write $a \prec b$, read aloud as "a precedes b."

6 Natural Numbers

6.1 Why

We want to count.

6.2 Definition

We define the set of **natural numbers** implicitly. There is an element of the set which we call **one**. Then we say that for each element n of the set, there is a unique corresponding element called the **successor** of n which is also in the set. The **successor function** is the implicitly defined a function from the set into itself associating elements with their successors. We call the elements **numbers** and the refer to the set itself as the **naturals**.

To recap, we start by knowing that one is in the set, and the successor of one is in the set. We call the successor of one **two**. We call the successor of two **three**. And so on using the English language in the usual manner. We are saying, in the language of sets, that the essence of counting is starting with one and adding one repeatedly.

6.2.1 Notation

We denote the set of natural numbers by N, a mnemonic for natural. We often denote elements of N by n, a mnemonic for number, or m, a letter close to n. We denote the element called one by 1.

6.3 Induction

We assert two additonal self-evident and indispensable properties of these natural numbers. First, one is the successor of no other element. Second, if we have a subset of the naturals containing one with the property that it contains successors of its elements, then that set is equal to the natural numbers. We call this second property the **principle of mathematical induction.**

These two properties, along with the existence and uniqueness of successors are together called **Peano's axioms** for the natural numbers. When in familiar company, we refer to the set of natural numbers as the **naturals** and we freely assume Peano's axioms.

6.4 Notation

As an exercise in the notation assumed so far, we can write Peano's axioms: N is a set along with a function $s : N \to N$ such that

1. $s(n)$ is the successor of n for all $n \in N$.
2. s is one-to-one; $s(n) = s(m) \implies m = n$ for all $m, n \in N$.
3. There does not exist $n \in N$ such that $s(n) = 1$.
4. If $T \subset N$, $1 \in T$, and $s(n) \in T$ for all $n \in T$, then $T = N$.

6.5 Order

Let $\preceq$ be a relation on N where $a \preceq b$, $a, b \in N$ if we can obtain b by applying the successor function to a finitely many times. It happens that $\preceq$ is a total order, so $(N, \preceq)$ is a lattice.

7 Algebra

7.1 Why

We want to combine set elements to get other set elements.

7.2 Basics

Let A be a non-empty set. An **operation** on A is a function $g : A \times A \to A$. Operations map ordered pairs of elements of a set to elements of the same set. An **algebra** is a set and an operation.

7.2.1 Notation

Let A a set and $g : A \times A \to A$. We commonly forego the notation $g(a, b)$ and instead write $a\, g\, b$. We call this style **infix** notation.

Using lower case latin letters for every the elements and for the operation is confusing, but we often have special symbols for particular operations. Examples of such symbols include $+$, $-$, $\cdot$, $\circ$, and $\star$.

If we had a set A and an operation $+ : A \times A \to A$, we would write $a + b$ for the result of applying $+$ to (a, b). In denoting the algebra, we would say let $(A, +)$ be an algebra.

7.3 Operation Properties

Let $(A, +)$ be an algebra. $+$ is **commutative** if $a + b = b + a$ for all $a, b \in A$. In this case we say that $+$ **commutes**. $+$ is **associative** if

$(a + b) + c = a + (b + c)$ for all $a, b, c, \in A$.

8 Set Algebra

8.1 Why

We want to consider the elements of two sets together at once, and other such sets created from two sets.

8.2 Definitions

Let A and B be two sets. The **union** of A with B is the set whose elements are in either A *or* B *or* both. The keyword in the definition is *or*. The **intersection** of A with B is the set whose elements are in both A *and* B. The keyword in the definition is *and*. Viewed as operations, both union and intersection commute; this property justifies the language "with." The intersection is a subset of A, of B, and of the union of A with B.

The **symmetric difference** of A and B is the set whose elements are in the union but not in the intersection. The symmetric difference commutes because both union and intersection commute; this property justifies the language "and." The symmetric difference is a subset of the union.

Let C a set containing A. The **complement** of A in C is the symmetric difference of A and C. Since $A \subset C$, the union is C and the intersection is A. So the complement is the "left-over" elements of B after removing the elements of A.

8.2.1 Notation

Let A, B be sets. We denote the union of A with B by $A \cup B$, read aloud as "A union B." $\cup$ is a stylized U. We denote the intersection of A with B by

$A \cap B$, read aloud as "A intersect B." We denote the symmetric difference of A and B by $A \Delta B$, read aloud as "A symdiff B." "Delta" is a mnemonic for difference.

Let C be a set containing A. We denote the complement of A in C by $C - A$, read aloud as "C minus A."

8.2.2 Results

Proposition 1 *For all sets A and B the operations $\cup$, $\cap$, and Δ commute.*

Proposition 2 *Let S a set. For all sets $A, B \subset S$,*

$$(1) \quad S - (A \cup B) = (S - A) \cap (S - B)$$
$$(2) \quad S - (A \cap B) = (S - A) \cup (S - B).$$

9 Arithmetic

9.1 Why

Counting one by one is slow so we define an algebra on the naturals.

9.2 Sums and Addition

Let m and n be two natural numbers. If we apply the successor function to m n times we obtain a number. If we apply the successor function to n m times we obtain a number. Indeed, we obtain the same number in both cases. We call this number the **sum** of m and n. We say we **add** m to n, or vice versa. We call this symmetric operation mapping (m, n) to their sum **addition**.

9.2.1 Notation

We denote the operation of addition by $+$ and so denote the sum of the naturals m and n by $m + n$.

9.3 Products and Multiplication

Let m and n naturals. If we add n copies of m we obtain a number. If we add m copies of n we obtain a number. Indeed, we obtain the same number in both cases. We call this number the **product** of m and n. We say we **multiply** m to n, or vice versa. We call this symmetric operation mapping (m, n) to their product **multiplication**.

9.3.1 Notation

We denote the operation of multiplication by $\cdot$ and so denote the product of the naturals m and n by $m \cdot n$.

10 Equivalence Relations

10.1 Why

We want to handle at once all elements which are indistinguishable or equivalent in some aspect.

10.2 Definition

Let R be a relation on A R is an **equivalence relation** if it is reflexive, symmetric, and transitive.

For an element $a \in A$, we call the set of elements in relation R to a the **equivalence class** of a. The key observation, recorded and proven below, is that the equivalence classes partition the set A. A frequent technique is to define an appropriate equivalence relation on a large set A and then to work with the set of equivalence classes of A.

We call the set of equivalence classes the **quotient set** of A under R. An equally good name is the divided set of A under R, but this terminology is not standard. The language in both cases reminds us that $\sim$ partitions the set A into equivalence classes.

10.2.1 Notation

If R is an equivalence relation on a set A, we use the symbol $\sim$. When alone, $\sim$ is read aloud as "sim," but we still read $a \sim b$ aloud as "a equivalent to b." We denote the quotient set of A under $\sim$ by $A/\sim$, read aloud as "A quotient sim".

10.2.2 Results

todo

10.3 Order Relations

Here we survey a two other special relation on a set. Let R a relation on the non-empty set A. We call R **anti-symmetric** if for two nonequal elements $a, b \in A$, $(a, b) \in R \implies (b, a) \notin R$. If R is reflexive, transitive, and anti-symmetric then we call R a **partial order** on A.

A **partially ordered set** is a set together with a partial order. The language partial is meant to suggest that two elements need not be comparable.j For example, suppose R is $\{(a, a) \mid a \in A\}$; we may justifiably call this no order at all and call A totally unordered, but it is a partial order by our definition.

Often we want all elements of the set A to be comparable. We call R **connexive** if for all $a, b \in A$, $(a, b) \in R$ or $(b, a) \in R$. If R is a partial order and connexive, we call it a **total order**.

A **totally ordered set** is a set together with a total order. The language is a faithful guide: we can compare any two elements. Still, we prefer one word to three, and so we will use the shorter term **chain** for a totally ordered set; other terms include **simply ordered set** and **linearly ordered set**.

10.3.1 Notation

We denote total and partial orders on a set A by $\preceq$. We read $\preceq$ aloud as "precedes or equal to" and so read $a \preceq b$ aloud as "a precedes or is equal to b." If $a \preceq b$ but $a \neq b$, we write $a \prec b$, read aloud as "a precedes b."

11 Direct Products

11.1 Why

We can profitably generalize the notion of cartesian product to families of sets indexed by the natural numbers.

11.2 Direct Products

The **direct product** of family indexed by a subset of the naturals is the set whose elements are ordered sequences of elements from each set in the family. The ordering on the sequences comes from the natural ordering on N. If the index set is finite, we call the elements of the direct product **n-tuples**. If the index set is the natural numbers, and every set in the family is the same set A, we call the elements of the direct product teh **sequences** in A.

11.2.1 Notation

For a family $\{A_\alpha\}_{\alpha \in I}$ of S with $I = \{1, \ldots, n\}$, we denote the direct product by

$$\prod_{i=1}^{n} A_i.$$

We read this notation as "product over alpha in I of A sub-alpha." We denote an element of $\prod_{i=1}^{n} A_i$ by $(a_1, a_2, \ldots, a_n)$ with the understanding that $a_1 \in A_1, a_2 \in A_2, \ldots, a_n \in A_n$.

If I is the set of natural numbers we denote the direct product by

$$\prod_{i=1}^{\infty} A_i.$$

We denote an element of $\prod_{i=1}^{\infty} A_i$ by (a_i) with the understanding that $a_i \in A_i$ for all $i = 1, 2, 3, \ldots$. If $A_i = A$ for all $i = 1, 2, 3, \ldots$, then (a_i) is a sequence in A.

12 Families

12.1 Why

We want to generalize operations beyond two objects.

12.2 Definition

We often refer to a collection of elemenets of some set via an index which takes value in another set. We call the indexing set the **index set** and set of indexed sets an **indexed family**. We can think of a function from the index set to the indexed family. If the elements indexed are something in particular, like sets, we call it an indexed family of sets. Often the indexing set is a subset of the natural numbers, or the natural numbers themselves, but it need not be.

12.2.1 Notation

Let S be a set. We often denote the index set by I, though this is not required. For each $\alpha \in I$, let $a_\alpha \in S$. We denote the family of a_α indexed with I by $\{a_\alpha\}_{\alpha \in I}$. The a_α notation is meant to evoke function notation and remind that we have a correspondence between the index set and the indexed set. We read this notation "a sub-alpha, alpha in I." If for each $\alpha \in I$ we have $A_\alpha \subset S$, we'd write $\{A_\alpha\}\alpha \in I$.

12.3 Family Set Algebra

We define the set whose elements are the objects which are contained in at least one family member the **family union**. We define the set whose elements are the obejcts which are contained in all of the family members the **family intersection**.

12.3.1 Notation

We denote the family union by $\cup_{\alpha \in I} A_\alpha$. We read this notation as "union over alpha in I of A sub-alpha." We denote family intersection by $\cap_{\alpha \in I} A_\alpha$. We read this notation as "intersection over alpha in I of A sub-alpha."

12.3.2 Results

Proposition 3 *For an indexed family* $\{A_\alpha\}_{\alpha \in I}$ *in* S*, if* $I = \{i, j\}$ *then*

$$\cup_{\alpha \in I} A_\alpha = A_i \cup A_j$$

and

$$\cap_{\alpha \in I} A_\alpha = A_i \cap A_j.$$

Proposition 4 *For an indexed family* $\{A_\alpha\}_{\alpha \in I}$ *in* S*, if* $I = \varnothing$*, then*

$$\cup_{\alpha \in I} A_\alpha = \varnothing$$

and

$$\cap_{\alpha \in I} A_\alpha = S.$$

Proposition 5 *For an indexed family* $\{A_\alpha\}_{\alpha \in I}$ *in* S*.*

$$C_S(\cup_{\alpha \in I} A_\alpha) = \cap_{\alpha \in I} C_S(A_\alpha)$$

and

$$C_S(\cap_{\alpha \in I} A_\alpha) = \cup_{\alpha \in I} C_S(A_\alpha).$$

13 Nets

13.1 Why

We want to generalize the notion of sequence.

13.2 Definition

Recall that a sequence is a function on the naturals. The naturals are ordered and have the property that we can always go further out. If handed two natural numbers m and n, we can always find another, for example $\max\{m, n\} + 1$, larger than m and n. We might think of larger as being further out from the first natural number, namely 1. These observations motivate defininining a directed set.

Definition 6 *A **directed set** is a set D with a partial order $\preceq$ satisfying one additional property: for all $a, b \in D$, there exists $c \in D$ such that $a \preceq c$ and $b \preceq c$.*

Definition 7 *A **net** is a function on a directed set.*

A sequence, then, is a net. The directed set is the set of natural numbers and the partial order is $m \preceq n$ if $m \leq n$.

13.2.1 Notation

Directed sets involve a set and a partial order. We commonly assume the partial order, and just denote the set. We use the letter D as a mnemonic for directed.

For nets, we use function notation and generalize sequence notation. We denote the net $x : D \to A$ by $\{a_\alpha\}$, emulating notation for sequences. The use of α rather than n reminds us that D need not be the set of natural numbers.

14 Categories

14.1 Why

We generalize the notion of sets and functions.

14.2 Definition

A **category** is a collection of objects together with a set of **maps** for each ordered pair of objects. The set of maps has a binary operation called **composition**, whose induced algebra is associative and contains identities.

As the fundamental example, consider the category whose objects are sets and whose maps are functions. The sets are the objects of the category. The functions are the maps. The rule of composition is ordinary function composition. The map identities are the identity functions. We call this category the **category of sets.**

14.2.1 Notation

Our notation for categories is guided by our generalizing the notions of set and functions.

We denote categories with upper-case latin letters in script; for example, $\mathcal{C}$. We read $\mathcal{C}$ aloud as "script C." Upper case latin letters remind that the category is a set of objects. The script form reminds that these objects may themselves be sets.

We denote the objects of a category by upper-case latin letters, for example A, B, C; an allusion to the idea that these generalize sets. We

denote the maps for an ordered pair of objects (A, B) by $A \to B$; an allusion to the function notation. We denote members of $A \to B$ using lower case latin letters f, g, h, in allusion function notation.

15 Groups

15.1 Why

We generalize the algebraic structure of addition over the integers.

15.2 Definition

Let $(A, +)$ be an algebra.

We call $e \in A$ an **identity** if (1) $e + a = e$ and (2) $a + e = e$ for all $a \in A$. If only (1) holds, we call e a **left identity**. If only (2) holds, we call e a **right identity**.

We call $b \in A$ an **inverse** of $a \in A$ if (1) $b + a = e$ and (2) $a + b = e$. If only (1) holds, we call e a **left inverse**. If only (2) holds, we call e a **right inverse**.

A **group** is an algebra $(A, +)$ where $+$ is associative, there exists an identity element in A, and there exists an inverse for each element of A. A **commutative group** is a group $(A, +)$ where $+$ commutes. A commutative group is also called an **Abelian group**.

15.2.1 Notation

TODO

16 Homomorphism

16.1 Why

We name a function which preserves group structure.

16.2 Definition

A **homomorphism** from group $(A, +)$ to group $(B, \tilde{+})$ is a function $f : A \to B$ such that $f(e_A) = f(e_B)$ for identities $e_A \in A$ and $e_B \in B$ and $f(a + a') = f(a)\tilde{+}f(a')$ for all $a, a' \in A$.

16.2.1 Notation

$\boxed{TODO}$