2 Basic Set Theory

We assume a basic knowledge of mathematics as introduced in [3, chapters I and II]. We briefly summarize the mathematical prerequisites in set theory [11, ch. 1, Basic set theory] and logic [26–28].

Contents

2.1 Notations

2.1.1 Term Notations

Let $\mathbb{N}$, $\mathbb{N}^+$, $\mathbb{Z}$, and $\mathbb{R}$ be the set of all natural, strictly positive naturals[1], integer, and real numbers.[2] Terms are symbolic expressions built out of constants (such as 0 and −42), mathematical variables such as x, y, and so on (denoting immutable but unknown entities such as x in the expression $x+1$), and operations such as + (sum), −, (difference), × (product), |_| (absolute value), and so forth. We write $p \triangleq P$ to define the mathematical object p (the *definiendum*) to be equal to the mathematical expression P (the *definiens*), for example, $2 \triangleq 1 + 1$. We use mathematical variables x, p, Q, and so forth to denote mathematical entities that are fixed over time and possibly unknown (see [28, ch. I] "On the use of variables"). We use program variables x, p, Q, and so forth to store values and change them over time by assignments.

2.1.2 Predicate Notations

Let $\mathbb{B} \triangleq \{\mathtt{tt}, \mathtt{ff}\}$ be the set of Boolean truth values tt (true) and ff (false).[3] We can state properties of numbers (also called assertions, conditions, constraints, facts, predicates, propositions, requirements, statements, etc.) by logical predicates for which the value is either tt or ff. Relations such as $(x - 1) + 1 = x$ or $x < x + 1$ are basic logical predicates relating terms by relation operations =

1. Other notations for positive naturals are $\mathbb{N}^*$, $\mathbb{N}_1$, $\mathbb{N}_{>0}$, $\mathbb{Z}^+$, and so forth. We use $\mathbb{N}^+$ because a superscript or subscript * is often interpreted as "zero or more" whereas $^+$ is often interpreted as "one or more." Originally, Richard Dedekind [4] and Giuseppe Peano [22] used $\mathbb{N}$ for positive naturals and $\mathbb{N}_0$ to include 0 [23].
2. The series of historical footnotes aims at showing that good notations may take decades if not centuries to emerge.
3. This notation is not classic. For example George Boole [1] uses 0 and 1, Giuseppe Peano uses V and Λ, David Hilbert and Wilhelm Friedrich Ackermann [9] use ϒ and ⅄, Stephen Cole Kleene [12] uses t and f, Alfred Tarski [28] uses T and F, and so forth.

(equal), $<$ (less than), $\leqslant$ (less than or equal), and so on. We can combine predicates using logical connectives $\vee$ (disjunction[4]), $\wedge$ (conjunction[5]), $\neg$ (negation[6]), $\Rightarrow$ (implication[7], with inverse $\Leftarrow$), and $\Leftrightarrow$ (if and only if)[8], as in $((x-1)+1=x) \wedge \neg(x > x+1)$. This is true for all possible values x so $\forall x \in \mathbb{Z} \,.\, (x-1)+1=x$ where $\forall$ is the universal quantification.[9] This property also holds on reals, but not on naturals because $\exists x \in \mathbb{N} \,.\, (x-1) \notin \mathbb{N}$ where $\exists$ is existential quantification[10] and $\notin$ means "does not belongs to" (by choosing $x=0$). Of course, $\forall x \in \mathbb{N} \,.\, (x+1)-1=x$ holds for naturals but not $\forall x \in \mathbb{N} \,.\, (x-1)+1=x$ (because 0 is an exception). We use $\exists!$ for unique existential quantification (i.e. $\exists! x \,.\, P(x)$ if and only if $\exists x \,.\, P(x) \wedge \forall x, y \,.\, (P(x) \wedge P(y)) \Rightarrow (x=y)$).[11]

If $P \Rightarrow Q$ then P is called the *premise* and Q the *conclusion*. We say that P is a *sufficient condition* for Q to hold (i.e. to be true). If $Q \Rightarrow P$ then we say that P is a *necessary condition* for Q to hold. If $P \Leftrightarrow Q$ then we say that P is a *necessary and sufficient condition* for Q. We say that $P \Rightarrow Q$ *holds vacuously* when P is $\mathtt{ff}$ or Q is $\mathtt{tt}$.

Exercise 2.1 Provide the truth tables of disjunction $\vee$, negation $\neg$, and implication $\Rightarrow$ (given all possible arguments $x, y \in \mathbb{B}$, such a truth table specifies the value $x \vee y \in \mathbb{B}$).

Exercise 2.2 Using the truth tables of exercise 2.1, prove that for all $x, y \in \mathbb{B}$, $x \Rightarrow y$ if and only if $\neg x \vee y$ if and only if $\neg(x \wedge \neg y)$.

4. George Boole [1] uses + (by analogy with addition).
5. George Boole [1] uses juxtaposition, so $x \wedge y$ is written xy (by analogy with a multiplication); Alfred Whitehead and Bertrand Russell use $\cdot$, which confusingly is also used as a delimiter [29, v. I, ch. I, 9–10].
6. George Boole [1, prop. III] writes $1-P$ for $\neg P$, Eliakim Hastings Moore writes $-P$ [18], Alfred Whitehead and Bertrand Russell [29] use $\sim$, Jacques Herbrand [8] refers to Alfred Whitehead and Bertrand Russell but uses ∞ instead. Following David Hilbert and Wilhelm Friedrich Ackermann [9], $\neg P$ is sometimes denoted $\overline{P}$.
7. Charles Sanders Peirce [24] uses $\prec$, Giuseppe Peano [22] uses $\supset$, Alfred Whitehead and Bertrand Russell [29], and Stephen Cole Kleene [12] uses $\supset$. This is unfortunate because if $P \Rightarrow Q$ then $\{x \mid P(x)\} \subseteq \{x \mid Q(x)\}$ (see section 2.3.2). David Hilbert and Wilhelm Friedrich Ackermann [9] and Alfred Tarski [28] use $\rightarrow$.
8. Alfred Whitehead and Bertrand Russell [29] use $\equiv$.
9. Introduced by Charles Sanders Peirce [24] as Any or $\prod$. David Hilbert and Wilhelm Friedrich Ackermann [9] as well as Alfred Whitehead and Bertrand Russell [29] write $(x)P$ for $\forall x \,.\, P$. Kazimierz Kuratowski and Andrzej Mostowski [14] use $\bigwedge_x P$. Alfred Tarski in [28] writes *for any*.
10. Introduced by Charles Sanders Peirce [24] as Some or Σ; for example $\forall P \,.\, \exists x \in P \,.\, \ldots$ is written $\prod_P \sum_{x_p} \ldots$. David Hilbert and Wilhelm Friedrich Ackermann [9] write $(Ex)P$ whereas Alfred Whitehead and Bertrand Russell [29] write $(\exists x)P$ for $\exists x \,.\, P$. Kazimierz Kuratowski and Andrzej Mostowski [14] use $\bigvee_x P$. Alfred Tarski [28] writes *there exists*.
11. i.e. stands for latin id est meaning "that is to say".

2.1.3 Set Notations

Naive set theory understands sets S as collections of elements x that belong to that set S, written $x \in S$.[12] The empty set $\varnothing$[13] has no elements so $\forall x \,.\, x \notin \varnothing$. A singleton is a set $\{x\}$ with only one element x, so $y \in \{x\} \Leftrightarrow y = x$. If $p(x)$ is a predicate on x, then the *set-builder notation* $\{x \mid p(x)\}$ denotes the set of all elements x satisfying the predicate p. Therefore, if $S \triangleq \{x \mid p(x)\}$ (i.e. S is defined in intension to be the set $\{x \mid p(x)\}$), then $x \in S$ if and only if $p(x)$ holds, that is, p is true of x. $S = \varnothing$ is empty if and only if $\forall x \,.\, \neg p(x)$.[14] For example, $2\mathbb{N} \triangleq \{x \in \mathbb{N} \mid \exists k \,.\, x = 2k\}$ is the set of all even natural numbers, and $2\mathbb{Z}+1 \triangleq \{x \in \mathbb{Z} \mid \exists k \,.\, x = 2k+1\}$ is the set of all odd integers. More generally, $d(a) \triangleq \{f(a,b,\ldots) \mid p(a,b,\ldots,x,\ldots)\}$ is a shorthand for the definition of function d as $\forall a \,.\, d(a) \triangleq \{f(a',b,\ldots) \mid \exists x \,.\, p(a',b,\ldots,x,\ldots) \wedge a' = a\}$. The free variables $a', b, \ldots, x, \ldots, a$ of the term $p(a',b,\ldots,x,\ldots) \wedge a' = a$ are existentially quantified unless they are free in $f(a',b,\ldots)$ or are a global variable a.[15] Contradictions such as the circular definition $S \triangleq \{x \mid x \notin S\}$ should be avoided. The cardinality $|S|$ of a finite set is the number of its elements (so, $|\varnothing| = 0$ and $|\{x\}| = 1$). For infinite sets, the cardinality is (informally) a measure of their size. For example the set $\mathbb{Z}$ of integers and the set $2\mathbb{Z}$ of even integers have the same cardinality (because there is a bijection $n \mapsto 2n$ between the two). We write $\{x \in X \mid p(x)\}$ for $\{x \mid x \in X \wedge p(x)\}$ and $\{x \in S\}$ for $\{x \mid x \in S\}$, which is S. We write $S \subseteq S'$ to mean that S is a subset of S', that is, $\forall x \in S \,.\, x \in S'$[16]. Therefore $\varnothing \subseteq S'$ is always true because there is no $x \in \varnothing$ so that all of them (and there is none) belong to any set. The union or join of sets is $\cup$ (defined as $x \in S \cup S' \Leftrightarrow x \in S \vee x \in S'$). The intersection or meet of sets is $\cap$. The difference of sets is $S \setminus S' \triangleq \{x \in S \mid x \notin S'\}$. The complement of a set S with respect to a set U (generally understood from the context) is $\neg S \triangleq U \setminus S$. Augustus De Morgan's laws state that $\neg(S \cup S') = (\neg S) \cap (\neg S')$ and $\neg(S \cap S') = (\neg S) \cup (\neg S')$. The powerset $\wp(S)$ of a set S is the set of all its subsets so $\wp(S) \triangleq \{S' \mid S' \subseteq S\}$[17]. $\wp_f(S)$ is the set of all finite subsets of S, so $\wp_f(S) \triangleq \{S' \mid S' \subseteq S \wedge |S| \in \mathbb{N}\}$. For example, $\wp(\{0,1\}) = \wp_f(\{0,1\}) = \{\varnothing, \{0\}, \{1\}, \{0,1\}\}$.

12. $\in$ can be remembered as the "e" in "is an element of". The symbol $\in$ was introduced by Giuseppe Peano [22]. Ernst Schröder uses the same notation $\notin$ both for $\in$ and $\subseteq$, [25] so formulæ must be parsed to sort between elements and sets! In fact $x \in X$ is $\{x\} \subseteq X$ so $\in$ is abstracted away in favor of $\subseteq$ in order theory, see chapter 10.

13. The denotation of the empty set by the letter $\varnothing$ from the Norwegian alphabet (similar to the German Ö) was introduced by the French mathematician André Weil.

14. Formally, [27, p. 34] defines $\{x \mid p(x)\} = y \Leftrightarrow (\forall x \,.\, (x \in y) \Leftrightarrow p(x) \wedge y \text{ is a set}) \vee (y = \varnothing \wedge \neg(\exists S \,.\, \forall x \,.\, x \in S \Leftrightarrow p(x)))$. "The point of the second member of the disjunction of the definiens is to put $\{x \mid p(x)\}$ equal to the empty set if there is no non-empty set having as members just those entities with property p."

15. A definition of the form $d(\vec{y}) \triangleq \{f(\vec{x}', \vec{y}) \mid P(\vec{x}', \vec{x}, \vec{y})\}$ has the global variables $\vec{y}$ of $f(\vec{x}', \vec{y})$ and $P(\vec{x}', \vec{x}, \vec{y})$ bound to the variables $\vec{y}$ in the definition of $d(\vec{y})$. The variables $\vec{x}'$ in $P(\vec{x}', \vec{x}, \vec{y})$ are bound to those of $f(\vec{x}', \vec{y})$ whereas $\vec{x}$ is free in $P(\vec{x}', \vec{x}, \vec{y})$ since it appears neither in $f(\vec{x}', \vec{y})$ nor in $d(\vec{y})$, nor (by assumption) under quantifiers in $P(\vec{x}', \vec{x}, \vec{y})$ so is implicitly existentially quantified. This is made more explicit by writing $d(\vec{y}) \triangleq \{f(\vec{x}', \vec{y}') \mid \exists \vec{x} \,.\, P(\vec{x}', \vec{x}, \vec{y}') \wedge y = y'\}$.

16. If $\forall x \,.\, p(x) \Rightarrow q(x)$ then $\{x \mid p(x)\} \subseteq \{x \mid a(x)\}$ so using $\supset$ for implication as [12] is confusing.

17. $\wp(S)$ is sometimes denoted 2^S. This is because a subset X of S can be represented by a boolean *characteristic function* $\mathbb{c}_{(x)} \triangleq x \in X$ characterizing which elements $x \in S$ belong to the subset X of S, the booleans are isomorphic to $2 = \{0,1\}$ (in John Von Neumann theory of ordinals [20]), and the set of functions from A to B which we denote as $A \to B$ is also denoted B^A.

Exercise 2.3 Formally define the property that a set S is a singleton.

Exercise 2.4 Formally define set intersection $\cap$.

Exercise 2.5 Prove De Morgan law $\neg(S \cup S') = (\neg S) \cap (\neg S')$.

More generally, by complement duality, if two predicates P and Q involving only variables X, $\vee/\exists$ (disjunction) or $\cup$ (union), $\wedge/\forall$ (conjunction) or $\cap$ (intersection), and $\neg$ (negation or complement) are logically equal $P \Leftrightarrow Q$ then their dual complements (replacing variables X by their negation $\neg X$, $\vee/\exists/\cup$ by $\wedge/\forall/\cap$, and $\wedge/\forall/\cap$ by $\vee/\exists/\cup$, and eliminating double negation $\neg\neg$) are also logically equal [9, ch I, § 6, and ch III, § 8]. This follows immediately from the fact that the dual complement of P is $\neg P$.

Exercise 2.6 Assume the set $\mathbb{R}$ of reals has already been defined (for example, by Tarski's axiomatization of the reals [28, Section 61 or 63]). Provide a definition of $\mathbb{N}$ and then $\mathbb{Z}$ as subsets of $\mathbb{R}$.

Remark 2.7 (on finiteness) Because programs run on finite machines with a finite lifetime, one may argue that everything concerning the programs and their executions must be finite [10]. However, infinite sets can be considered as the limit of finite sets for which cardinality is bounded but the bound is unknown. It is often easier to reason on infinite sets than on finite sets which cardinality has an unknown bound. A classical example is fairness or unbounded nondeterminism in parallelism [6, 15, 21], in which verification methods not explicitly considering the unknown bound are simpler. Another argument is that computations are currently performed on networks, in which a dying machine is replaced by another one without affecting the well-behaved operation of the network, hence of program executions that can, in principle, last forever. Distributed memory is also finite, but in contrast to a machine, its size bound is not known. Considering that the network has a finite lifetime is impossible, sibecausence it is necessary to give a lower bound to be able to prove anything. This lower bound is itself completely unknown. Therefore, considering an infinite lifetime, hence infinite sets, is a reasonable simplification to get rid of these minimal and maximal unknown resource and lifetime bounds.

2.1.4 Interval Notation

Given a lower and an upper integer bound $\ell, u \in \mathbb{Z}$, we let the closed interval be $[\ell, u] \triangleq \{z \in \mathbb{Z} \mid \ell \leqslant z \leqslant u\}$ while the opened intervals are $]\ell, u] \triangleq \{z \in \mathbb{Z} \mid \ell < z \leqslant u\}$, $[\ell, u[\triangleq \{z \in \mathbb{Z} \mid \ell \leqslant z < u\}$, and $]\ell, u[\triangleq \{z \in \mathbb{Z} \mid \ell < z < u\}$. Hence $[\ell, u] = \varnothing$ when $u < \ell$.

2.2 Mathematical Constructions

Mathematicians (followed by computer scientists) construct new objects and operations on these objects from previously defined ones (sets for mathematicians, and bits for computer scientists). For

example, Kazimierz Kuratowski [13, def. V, p. 171] defines a pair (sometimes called an ordered pair) as a set $\langle x,\ y\rangle \triangleq \{\{x\},\{x,y\}\}$. The first coordinate is $\langle x,\ y\rangle_1$ defined by $z = \langle x,\ y\rangle_1 \Leftrightarrow \forall S \in \langle x,\ y\rangle : z \in S$ (so that $\langle x,\ y\rangle_1 = x$). The second coordinate is $\langle x,\ y\rangle_2$ defined by $z = \langle x,\ y\rangle_2 \Leftrightarrow (\exists S \in \langle x,\ y\rangle : z \in S) \wedge (\forall S_1, S_2 \in \langle x,\ y\rangle : S_1 \neq S_2 \rightarrow (z \notin S_1 \vee z \notin S_2))$ (so that $\langle x,\ y\rangle_2 = y$).

Exercise 2.8 Show that $\langle x,\ x\rangle_1 = \langle x,\ x\rangle_2 = x$.

2.2.1 Cartesian Product

The Cartesian product of two sets S_1 and S_2 is the set $S_1 \times S_2 \triangleq \{\langle x,\ y\rangle \mid x \in S_1 \wedge y \in S_2\}$ of pairs of elements of S_1 and S_2. The set $S \times S$ is sometimes written S^2. This generalizes to n-ary tuples, $n \in \mathbb{N}^+$, as $\langle x_1,\ x_2,\ \ldots,\ x_n\rangle \in S_1 \times S_2 \times \ldots \times S_n$ and S^n when $S_1 = S_2 \ldots = S_n = S$. The selection of the ith element of the tuple $\langle x_1,\ x_2,\ \ldots,\ x_n\rangle$ uses the index subscript notation $\langle x_1,\ x_2,\ \ldots,\ x_n\rangle_i \triangleq x_i$ or the function notation $\langle x_1,\ x_2,\ \ldots,\ x_n\rangle(i) \triangleq x_i$. When the index is not an integer, we write $\prod_{i\in\Delta} x_i$, which we understand as a function from the set Δ of indexes to S.

2.2.2 Relations

A binary relation r on sets S_1 and S_2 is a set of pairs $\langle x,\ y\rangle \in S_1 \times S_2$ of related elements $x \in S_1$ and $y \in S_2$ so $r \in \wp(S_1 \times S_2)$. y is said to be an image of x by r. We write $x\ r\ y, x \xrightarrow{r} y$, or $r(x,y)$ for $\langle x, y\rangle \in r$. For example, equality on $\mathbb{Z}$ is $= \triangleq \{\langle x,\ x\rangle \mid x \in \mathbb{Z}\}$ whereas "less than or equal" is $\leqslant \triangleq \{\langle x, y\rangle \in \mathbb{Z} \times \mathbb{Z} \mid \exists z \in \mathbb{N} \,.\, x + z = y\}$, and the identity relation on a set S is $\mathbb{1}_S \triangleq \{\langle x,\ x\rangle \mid x \in S\}$. The domain of a relation r on sets S_1 and S_2 is $\mathsf{dom}(r) \triangleq \{x \in S_1 \mid \exists y \in S_2 \,.\, \langle x,\ y\rangle \in r\}$; the codomain is $\mathsf{cod}(r) \triangleq \{y \in S_2 \mid \exists x \in S_1 \,.\, \langle x,\ y\rangle \in r\}$; and the field is $\mathsf{fld}(r) \triangleq \mathsf{dom}(r) \cup \mathsf{cod}(r)$. The left (respectively right) restriction $r \restriction S$ (respectively $r \lceil S$) of a relation $r \in \wp(S_1 \times S_2)$ to a set S is $\{\langle x, y\rangle \in r \mid x \in S\}$ (respectively $\{\langle x,\ y\rangle \in r \mid y \in S\}$), which is equal to $\{\langle x,\ y\rangle \in r \mid x \in S_1 \cap S\}$ (respectively $\{\langle x,\ y\rangle \in r \mid y \in S_2 \cap S\}$). The composition of relations is $r_1 \fatsemi r_2 \triangleq \{\langle x,\ z\rangle \mid \exists y \,.\, \langle x, y\rangle \in r_1 \wedge \langle y,\ z\rangle \in r_2\}$. The inverse of a relation $r \in \wp(S_1 \times S_2)$ is $r^{-1} \triangleq \{\langle y,\ x\rangle \mid \langle x,\ y\rangle \in r\}$. $\langle \wp(S \times S),\ \fatsemi,\ \mathbb{1}_S\rangle$ is an example of *monoide* that is a mathematical structure $\langle \mathcal{S},\ \oplus,\ 1\rangle$, where $\oplus$ is a binary relation on the set $\mathcal{S}$, which is associative (i.e. $(x \oplus y) \oplus z = x \oplus (y \oplus z)$ with neutral element 1 (i.e. $1 \oplus x = x \oplus 1 = x$).

Exercise 2.9 Show that $\langle \wp(S \times S),\ \fatsemi,\ \mathbb{1}_S,\ ^{-1}\rangle$ is not a *group* that is a mathematical structure $\langle \mathcal{S},\ \oplus,\ 1,\ ^{-1}\rangle$ where $\oplus$ is a binary relation on the set $\mathcal{S}$, with neutral element 1, and inverse $^{-1}$ (i.e. $x \oplus x^{-1} = x^{-1} \oplus x = 1$).

2.2.3 Equivalence and Partial Order Relations

An *equivalence relation* $\equiv$ on a set S is *reflexive* ($\forall x \in S \,.\, x \equiv x$), *symmetric* ($\forall x, y \in S \,.\, (x \equiv y) \Leftrightarrow (y \equiv x)$), and *transitive* ($\forall x, y, z \in S \,.\, (x \equiv y \wedge y \equiv z) \Rightarrow (x \equiv z)$). The equivalence class of an element $x \in S$ is the set $[x]_{\equiv} \triangleq \{y \in S \mid y \equiv x\}$ of all elements of S that are equivalent to x. The quotient $S|_{\equiv} \triangleq \{[x]_{\equiv} \mid x \in S\}$ is the set of all equivalence classes.

A *partial order* $\leqslant$ on a set S is reflexive, *antisymmetric* ($\forall x, y \in S \,.\, (x \leqslant y \wedge y \leqslant x) \Rightarrow (x = y)$) and transitive. The *strict* partial order is $x < y \triangleq (x \leqslant y) \wedge (x \neq y)$. An order is *total* if and only

if any two elements of S are comparable ($\forall a, b \in S \;.\; (a \leqslant b) \vee (b \leqslant a)$). A set S equipped with a partial order $\leqslant$ is called a *poset* $\langle S, \leqslant \rangle$.

A *pointwise* definition of a relation is $\dot{r} \triangleq f, g \mapsto \forall x \;.\; r(f(x), g(x))$. The functional pointwise definition is $\ddot{r} \triangleq f, g \mapsto \forall X \;.\; \dot{r}(f(X), g(X)) = f, g \mapsto \forall X \;.\; \forall x \;.\; r(f(X)x, g(X)x)$, and so forth. For example, $f \mathrel{\dot{\sqsubseteq}} g$ is $\forall x \;.\; f(x) \sqsubseteq g(x)$. If $\langle \langle L_i, \sqsubseteq_i \rangle, i \in \Delta \rangle$ is a family of posets (see section 2.2.5), then the *componentwise* order $\dot{\sqsubseteq}$ on the Cartesian product $\prod_{i\in\Delta} L_i$ is $\prod_{i\in\Delta} x_i \mathrel{\dot{\sqsubseteq}} \prod_{i\in\Delta} y_i \triangleq \forall i \in \Delta \;.\; x_i \sqsubseteq_i y_i$. The componentwise order $\dot{\sqsubseteq}$ is sometimes denoted $\prod_{i\in\Delta} \sqsubseteq_i$ or $\sqsubseteq_1 \times \sqsubseteq_2$ when $\Delta = \{1, 2\}$.

Exercise 2.10 Prove that equality (=) on a non-empty set S is the only relation on that set S that is both an equivalence and a partial order.

Exercise 2.11 (Lexicographic Order) Let $\langle A, \leqslant \rangle$ and $\langle B, \sqsubseteq \rangle$ be two posets. Define $\langle a, b \rangle \leqslant \times \sqsubseteq \langle a', b' \rangle \triangleq (a < a') \vee (a = a' \wedge b \sqsubseteq b')$. Show that the Cartesian product $\langle A \times B, \leqslant \times \sqsubseteq \rangle$ is a poset. Show that if $\leqslant$ and $\sqsubseteq$ are total orders then $\leqslant \times \sqsubseteq$ is a total order.

More details on equivalence and partial order relations appear in chapter 10.

2.2.4 Partial and Total Functions

A relation $r \in \wp(S_1 \times S_2)$ is functional when any element of S_1 is in relation with at most one element of S_2 by r that is $\forall x \in S_1 \;.\; \forall y, y' \in S_2 \;.\; (\langle x,\ y \rangle \in r \wedge \langle x,\ y' \rangle \in r) \Rightarrow (y = y')$. In that case, we write $r \in \wp_{\mathrm{F}}(S_1 \times S_2)$. A relation $r \in \wp(S_1 \times S_2)$ is total when any element of S_1 has at least one image by r that is $\forall x \in S_1 \;.\; \exists y \in S_2 \;.\; \langle x,\ y \rangle \in r$.

A partial function $f \in S_1 \nrightarrow S_2$ of S_1 into S_2 is a functional relation r on sets S_1 and S_2 such that any element $x \in S_1$ has at most an image, written $f(x)$ when it exists (so $f(x)$ is the unique y such that $\langle x,\ y \rangle \in r$). We write $f \triangleq x \mapsto\!\!\!\!\!\rightarrow e(x)$ when $\forall x \in \mathsf{dom}(f) \;.\; f(x) \triangleq e(x)$ and $f \triangleq x \in S \mapsto\!\!\!\!\!\rightarrow e(x)$ when $S = \mathsf{dom}(f)$. We sometimes use the subscript notation f_x for $f(x)$. The composition of partial functions is $f \circ g = x \mapsto\!\!\!\!\!\rightarrow f(g(x))$. Considered as relations, this is $g \fatsemi f$.

A total function $f \in S_1 \to S_2$ has $\mathsf{dom}(f) = S_1$, that is, is everywhere defined on S_1, which we write $x \in S_1 \mapsto f(x)$. If $S_1 = S_2 = S$ then $f \in S \to S$ is often called an *operator* on S or an *S-transformer*. A function $F \in (S_1 \to S_2) \to (S_1' \to S_2')$ taking functions as parameters is called a *functional*.

The right image of a relation $r \in \wp(S_1 \times S_2)$ is the function $x \in S_1 \mapsto \{y \in S_2 \mid \langle x,\ y \rangle \in r\} \in S_1 \to \wp(S_2)$.

A total function $f \in S_1 \to S_2$ is injective/one-to-one when $\forall x_1 \in S_1 \;.\; \forall x_2 \in S_2 \;.\; x_1 \neq x_2 \Rightarrow f(x_1) \neq f(x_2)$ (written $f \in S_1 \rightarrowtail S_2$). It is surjective/onto if and only if $\forall y \in S_2 \;.\; \exists x \in S_1 \;.\; f(x) = y$ (written $f \in S_1 \twoheadrightarrow S_2$). It is bijective if and only if both injective and surjective (written $f \in S_1 \rightarrowtail\!\!\!\!\!\twoheadrightarrow S_2$). Sets S_1 and S_2 are isomorphic when there exists a bijection of S_1 onto S_2.

It may be that the returned value of a function always belong to a set that depends upon one of its parameters (this is called a dependent type in computer science [2]). For example $f \in n \in \mathbb{N} \to \{k \in \mathbb{N} \mid k \geqslant n\}$ specifies a function $f \in \mathbb{N} \to \mathbb{N}$ such that $\forall n \in \mathbb{N} \;.\; f(n) \geqslant n$. More generally,

$f \in x \in S_1 \to S_2(x)$ is the function $f \in x \in S_1 \to \bigcup_{x \in S_1} S_2(x)$ such that $\forall x \in S_1 \,.\, f(x) \in S_2(x)$ where S_2 maps each $x \in S_1$ to a set $S_2(x)$. Up to an isomorphism $f \in \prod_{x \in S_1} S_2(x)$.

Exercise 2.12 Show that $\mathbb{N}$, $2\mathbb{N}$, and $2\mathbb{N} + 1$ are isomorphic.

A set S is *enumerable* if and only if there exists a bijection $\iota \in S \rightarrowtail\!\!\!\!\!\rightarrow \mathbb{N}$ between S and the naturals.

A function may return a function, and $f \in A \to B \to C$ stands for $f \in A \to (B \to C)$, the same for partial functions $\nrightarrow$.

A *pointwise* definition of a function is $\dot{f} \triangleq x \mapsto f(x)$, $\ddot{f} \triangleq X \mapsto \dot{f}(X) = X \mapsto x \mapsto f(X)x$ for functionals, and so forth.

2.2.5 Families

A family $F \in \Delta \to S$ of elements of S indexed by Δ is a map from a set Δ (called the domain or index set, which may be infinite) into a set S. Such a family defines a set $\{F(i) \mid i \in \Delta\}$ (where $F(i)$ is often denoted F_i with an index $i \in \Delta$). It also defines a Cartesian product $\prod_{i \in \Delta} F_i$ as well as a sequence $\langle F_i,\ i \in \Delta \rangle$ when Δ is totally ordered.

2.2.6 Recursive Definitions

An example of recursive definition of a function $f \in \mathbb{N} \to S$ where S is a set has the form $f(0) \triangleq c$ where $c \in S$ and $f(n) \triangleq F(n, f(n-1))$ where $F \in \mathbb{N} \times S \to S$. For example the factorial is $!0 \triangleq 1$ and $!n \triangleq n \times !(n-1)$. Recursive programming was promoted, among others, by [5, 16].

Recursive definitions may be ill defined, such as $f(0) \triangleq 0$ and $f(n) \triangleq f(n+1)$ when $n \neq 0$. We have $f(n) = 0$ for $n \leqslant 0$, whereas $f(n)$ is undefined when $n > 0$. For programs, *undefined* means "does not terminate" or "terminates with a runtime error" (such as `Stack overflow` or `Segmentation fault`, etc.). So recursive definitions must be proved to be well-defined. (e.g. $! \in \mathbb{N} \to \mathbb{N}$).[18] Recursive definitions are generalized to inductive definitions in section 16.4.

2.3 Properties

2.3.1 Properties Are Sets

Properties (e.g. "to be an even integer" and "to be an odd natural") can be understood as the set of mathematical objects that have this property (e.g. $2\mathbb{Z} \triangleq \{x \in \mathbb{Z} \mid \exists k \in \mathbb{Z} \,.\, x = 2k\}$ and $2\mathbb{N} + 1 = \{x \in \mathbb{N} \mid \exists k \in \mathbb{N} \,.\, x = 2k + 1\}$). Hence if P is a property then $x \in P$ means "x has property P", and $x \notin P$ means "x does not have property P." For example, $42 \in 2\mathbb{Z}$[19] but $43 \notin 2\mathbb{Z}$ and $! \in \mathbb{N} \to \mathbb{N}$ and $! \notin \mathbb{Z} \to \mathbb{Z}$. The concept of property is fully developed in chapter 8, "Program Properties".

18. e.g. stands for Latin *exempli gratia* or example given.

19. 42, in the *Hitchhiker's Guide to the Galaxy* by Douglas Adams, is the "Answer to the Ultimate Question of Life, the Universe, and Everything", calculated by a supercomputer named Deep Thought over a period of 7.5M years.

2.3.2 Implication, Weaker and Stronger Properties

In considering properties as sets, the logical implication is subset inclusion $\subseteq$. For example, "to be greater than 42 implies to be positive" is $\{x \in \mathbb{Z} \mid x > 42\} \subseteq \{x \in \mathbb{Z} \mid x \geqslant 0\}$. If $P \subseteq Q$ then P is said to be stronger/more precise than Q, and Q is said to be weaker/less precise than P. Stronger/more precise properties are satisfied by fewer elements, and weaker/less precise properties are satisfied by more elements. $\mathbb{ff}$, that is, $\varnothing$, is the strongest property, and $\mathbb{tt}$, that is, $\mathbb{Z}$, is the weakest property of integers.

2.4 Proofs

2.4.1 Proof by Contraposition

A proof of $P \Rightarrow Q$ by contraposition consists in proving the contrapositive $\neg Q \Rightarrow \neg P$. If P is true then $\neg P$ is false, so $\neg Q$ cannot be true because then $\neg P$ would be true and therefore Q is true.

2.4.2 Proof by Reductio Ad Absurdum or by Contradiction

A proof of P by *reductio ad absurdum* consists in finding a property Q that is known to be true and proving $\neg P \Rightarrow \neg Q$. By contraposition $Q \Rightarrow P$ that is $\mathbb{tt} \Rightarrow P$ and so P is true.

2.4.3 Proof by Recurrence

Theorem 2.13 (proof by recurrence) To prove that a property P holds for all natural numbers i.e. $\mathbb{N} \subseteq P$ equivalently $\forall n \in \mathbb{N} \,.\, n \in P$, the proof by recurrence consists in proving $0 \in P$ and $\forall n \in \mathbb{N} \,.\, (n \in P) \Rightarrow (n + 1 \in P)$.

$n \in P$ is called the induction hypothesis (abbreviated ind. hyp., and also called recurrence hypothesis) so $n + 1 \in P$ must be proved assuming this induction hypothesis.

Proof (soundness of the proof by recurrence) Assume that we have made the proof by recurrence and $\mathbb{N} \nsubseteq P$. Then $\exists n \in \mathbb{N} \,.\, n \notin P$. The case $n = 0$ is impossible because we proved $0 \in P$. Therefore $n > 0$ hence $n = (n - 1) + 1$. We proved that $\forall m \in \mathbb{N} \,.\, (m \in P) \Rightarrow (m + 1 \in P)$ so $\neg(m+1 \in P) \Rightarrow \neg(m \in P)$. For $m = n-1$ we have $n-1 \notin P$. Going on this way, $n-2 \notin P$, $n-3 \notin P$, …, $0 \notin P$ in contradiction with the proof that $0 \in P$. By reductio ad absurdum $\neg(\exists n \in \mathbb{N} \,.\, n \notin P)$, that is, $\forall n \in \mathbb{N} \,.\, n \in P$.

Proof (completeness of the proof by recurrence) If P holds (that is $\mathbb{N} \subseteq P$) then this can always be proved by recurrence. Let $Q \triangleq P \cap \mathbb{N}$, so that $\mathbb{N} \subseteq P$ implies $Q = \mathbb{N}$. So trivially, $0 \in Q$ and $\forall n \in Q \,.\, n + 1 \in Q$. Therefore we have $\mathbb{N} \subseteq Q = P \cap \mathbb{N} \subseteq P$.

So $\mathbb{N} \subseteq P$ can be proved by recurrence (maybe with a stronger recurrence hypothesis Q and an additional implication $Q \subseteq P$).

Proof by recurrence dates back to Pierre de Fermat infinite descent method [7] (originally formulated contrapositively as $\forall n \in \mathbb{N} \,.\, (n \notin P) \Rightarrow (\exists m < n \,.\, m \notin P)$ then $\forall n \in \mathbb{N} \,.\, n \in P$). The term mathematical induction is credited to Augustus De Morgan [19].

Exercise 2.14 Prove that factorial $! \in \mathbb{N} \to \mathbb{N}$ by recurrence.

2.5 Conclusion

Set theory is the logical basis for all mathematics and computer science. Additional topics in set theory will be covered subsequently, as needed. The reader may enjoy studying more advanced introductions to abstract set theory, such as [17].

2.6 Solutions to Selected Exercises

Solution to Exercise 2.3 S is a singleton if and only if $\exists x \,.\, x \in S \wedge \forall x, y \in S \,.\, x = y$.

Solution to Exercise 2.6 $\mathbb{N}$ is the smallest subset of $\mathbb{R}$ containing 0 and the successor of every natural that is $\mathbb{N} = \bigcap\{S \in \wp(\mathbb{R}) \mid 0 \in S \wedge \forall n \in S \,.\, n+1 \in S\}$. $\mathbb{N}^+ = \mathbb{N} \setminus \{0\}$, $\mathbb{Z} = \mathbb{N} \cup \{-n \mid n \in \mathbb{N}^+\}$.

Solution to Exercise 2.9 Take $S = \{a, b, c\}$, $r = \{\langle a,\ c\rangle, \langle b,\ c\rangle\}$ so $r^{-1} = \{\langle c,\ a\rangle, \langle c,\ b\rangle\}$ and $r \fatsemi r^{-1} = \{\langle a,\ a\rangle, \langle a,\ b\rangle, \langle b,\ b\rangle, \langle b,\ a\rangle\} \neq \mathbb{1}_S = \{\langle a,\ a\rangle, \langle b,\ b\rangle\}$.

Solution to Exercise 2.14 We have $!0 = 1$ by definition, so $!0 \in \mathbb{N}$. Assume, by induction hypothesis, that $!m \in \mathbb{N}$ for all $m < n+1$. Then $n < n+1$ so $!n \in \mathbb{N}$ by induction hypothesis and therefore $!(n+1) = (n+1) \times !n \in \mathbb{N}$ by definition of the factorial and $\times \in \mathbb{N}^2 \to \mathbb{N}$. By recurrence, $\forall n \in \mathbb{N} \,.\, !n \in \mathbb{N}$ so $! \in \mathbb{N} \to \mathbb{N}$.

2.7 Bibliography

[1] George Boole. *An Investigation of the Laws of Thought on Which are Founded the Mathematical Theories of Logic and Probabilities.* Walton and Maberly, 1884 (1, 2).

[2] Simon Castellan, Pierre Clairambault, and Peter Dybjer. "Categories with Families: Unityped, Simply Typed, and Dependently Typed." *CoRR.* abs/1904.00827 (2019) (6).

[3] Richard Courant and Herbert Robbins (revised by Ian Stewart). *What is Mathematics? An Elementary Approach to Ideas and Methods.* 2nd ed. Oxford University Press, 1941, 1969, 1996 (1).

[4] Richard Dedekind. *Was sind und was sollen die Zahlen?.* 2nd ed. Friedrich Bieweg und Sohn, 1893 (1, 8).

[5] Edsger W. Dijkstra. "Recursive Programming." *Numerische Mathematik.* 2.1 (1960), pp. 312–318 (7, 12).

[6] Edsger W. Dijkstra and Carel S. Scholten. *The Strongest Postcondition.* Texts and Monographs in Computer Science. Springer, 1990, pp. 209–215 (4, 10).

[7] Pierre de Fermat. "Relation des nouvelles découvertes en la science des nombres." Lettre à Pierre de Carcavi. Aug. 1659 (9).

[8] Jacques Herbrand. "Recherches sur la théorie de la démonstration." Ch. V of "Écrits logiques", Jean Van Heijenoort (Ed.), Presses Universitaires de France, 1968, pp. 35–143. Thèse. Université de Paris, June 11, 1930 (2, 1, 18, 45).

[9] David Hilbert and Wilhelm Ackermann. *Grundzüge der Theoretischen Logik*. 6th ed. Engl. trans. "Principles of mathematical logic", Lewis M. Hammond, George G. Leckie, F. Steinhardt, AMS Chelsea, 1958, reprinted 2008. Springer, 1928, 1949, reprinted 1959 (1, 2, 4).

[10] Gerard J. Holzmann. "Does Not Compute." *IEEE Software*. 36.3 (2019), pp. 14–16 (4).

[11] Irving Kaplansky. *Set Theory and Metric Spaces*. 2nd ed. American Mathematical Society, 1977 (1).

[12] Stephen Cole Kleene. *Introduction to Meta-Mathematics*. Elsevier North-Holland, 1952 (1–3, 8).

[13] Kazimierz Kuratowski. "Sur la notion de l'ordre dans la Théorie des Ensembles." *Fundamenta Mathematicæ*. 2.1 (1921), pp. 161–171 (5).

[14] Kazimierz Kuratowski and Andrzej Mostowski. *Set Theory*. North-Holland, Jan. 1968 (2).

[15] Daniel J. Lehmann, Amir Pnueli, and Jonathan Stavi. "Impartiality, Justice and Fairness: The Ethics of Concurrent Termination." In *ICALP*. Vol. 115. Lecture Notes in Computer Science. Springer, 1981, pp. 264–277 (4).

[16] John McCarthy. "Recursive Functions of Symbolic Expressions and Their Computation by Machine, Part I." *Commun. ACM*. 3.4 (1960), pp. 184–195 (7, 1, 24).

[17] James Donald Monk. *Introduction to Set Theory*. McGraw-Hill, 1969. `http://euclid.colorado.edu/~monkd/monk11.pdf` (9).

[18] Eliakim Hastings Moore. *Introduction to a Form of General Analysis*. Yale University Press, 1910 (2, 21, 1).

[19] Augustus De Morgan. "Mathematical Induction." *The Penny Cyclopedia of the Society for the Diffusion of Useful Knowledge*. 12 (1838). `http://education.lms.ac.uk/wp-content/uploads/2011/10/De_Morgan_Mathematical_Induction.pdf` (9).

[20] John Von Neumann. "Zur Einführung der transfiniten Zahlen." *Acta Scientiarum Mathematicarum (Szeged)*. 1.4 (1923), pp. 199–208 (3).

[21] David Michael Ritchie Park. "On the Semantics of Fair Parallelism." In *Abstract Software Specifications*. Vol. 86. Lecture Notes in Computer Science. Springer, 1979, pp. 504–526 (4, 7).

[22] Giuseppe Peano. *Arithmetices Principia, Nova Methodo Exposita*. Fratres Bocca, 1889. `https://archive.org/details/arithmeticespri00peangoog/` (1–3, 11).

[23] Giuseppe Peano. *Formulaire de Mathématiques*. Bocca frères, 1895. `https://archive.org/details/formulairedemat03peangoog/page/n9` (1).

[24] Charles Sanders Peirce. "On the Algebra of Logic: A Contribution to the Philosophy of Notation." *American Journal of Mathematics*. 7.2 (Jan. 1885), pp. 180–202 (2).

[25] Ernst Schröder. *Vorlesungen über die Algebra der Logik (Exakte Logik)*. Vol. 1. B.G. Teubner, 1890 (3).

[26] Raymond Smullyan. *A Beginner's Guide to Mathematical Logic*. Dover Books on Mathematics, 2014 (1).

[27] Patrick Suppes. *Axiomatic Set Theory*. Dover, 1952 (1, 3).

[28] Alfred Tarski. *Introduction to Logic and to the Methodology of Deductive Sciences*. 4th ed. Oxford University Press, Mar. 1994 (1, 2, 4).

[29] Alfred North Whitehead and Bertrand Russell. *Principia Mathematica, Volume I, II, III*. 2nd ed. Cambridge University Press, 1927 (2).