Preliminaries

We establish here a few notational conventions used throughout the text.

Arithmetic with ∞

We shall sometimes use the symbols "∞" and "$-\infty$" in simple arithmetic expressions involving real numbers. The interpretation given to such expressions is the usual, natural one; for example, for all real numbers x, we have $-\infty < x < \infty$, $x + \infty = \infty$, $x - \infty = -\infty$, $\infty + \infty = \infty$, and $(-\infty) + (-\infty) = -\infty$. Some such expressions have no sensible interpretation (e.g., $\infty - \infty$).

Logarithms and exponentials

We denote by $\log x$ the natural logarithm of x. The logarithm of x to the base b is denoted $\log_b x$.

We denote by e^x the usual exponential function, where $e \approx 2.71828$ is the base of the natural logarithm. We may also write $\exp[x]$ instead of e^x.

Sets and relations

We use the symbol $\emptyset$ to denote the empty set. For two sets A, B, we use the notation $A \subseteq B$ to mean that A is a subset of B (with A possibly equal to B), and the notation $A \subsetneq B$ to mean that A is a proper subset of B (i.e., $A \subseteq B$ but $A \neq B$); further, $A \cup B$ denotes the union of A and B, $A \cap B$ the intersection of A and B, and $A \setminus B$ the set of all elements of A that are not in B.

For sets $S_1, \ldots, S_n$, we denote by $S_1 \times \cdots \times S_n$ the **Cartesian product** of $S_1, \ldots, S_n$, that is, the set of all n-tuples $(a_1, \ldots, a_n)$, where $a_i \in S_i$ for $i = 1, \ldots, n$.

We use the notation $S^{\times n}$ to denote the Cartesian product of n copies of a set S, and for $x \in S$, we denote by $x^{\times n}$ the element of $S^{\times n}$ consisting of n copies of x. (We shall reserve the notation S^n to denote the set of all nth powers of S, assuming a multiplication operation on S is defined.)

Two sets A and B are **disjoint** if $A \cap B = \emptyset$. A collection $\{C_i\}$ of sets is called **pairwise disjoint** if $C_i \cap C_j = \emptyset$ for all i, j with $i \neq j$.

A **partition** of a set S is a pairwise disjoint collection of non-empty subsets of S whose union is S. In other words, each element of S appears in exactly one subset.

A **binary relation** on a set S is a subset R of $S \times S$. Usually, one writes $a \sim b$ to mean that $(a, b) \in R$, where $\sim$ is some appropriate symbol, and rather than refer to the relation as R, one refers to it as $\sim$.

A binary relation $\sim$ on a set S is called an **equivalence relation** if for all $x, y, z \in S$, we have

- $x \sim x$ (reflexive property),
- $x \sim y$ implies $y \sim x$ (symmetric property), and
- $x \sim y$ and $y \sim z$ implies $x \sim z$ (transitive property).

If $\sim$ is an equivalence relation on S, then for $x \in S$ one defines the set $[x] := \{y \in S : x \sim y\}$. Such a set $[x]$ is an **equivalence class**. It follows from the definition of an equivalence relation that for all $x, y \in S$, we have

- $x \in [x]$, and
- either $[x] \cap [y] = \emptyset$ or $[x] = [y]$.

In particular, the collection of all distinct equivalence classes partitions the set S. For any $x \in S$, the set $[x]$ is called the the **equivalence class containing** x, and x is called a **representative** of $[x]$.

Functions

For any function f from a set A into a set B, if $A' \subseteq A$, then $f(A') := \{f(a) \in B : a \in A'\}$ is the **image** of A' under f, and $f(A)$ is simply referred to as the **image** of f; if $B' \subseteq B$, then $f^{-1}(B') := \{a \in A : f(a) \in B'\}$ is the **pre-image** of B' under f.

A function $f : A \to B$ is called **one-to-one** or **injective** if $f(a) = f(b)$ implies $a = b$. The function f is called **onto** or **surjective** if $f(A) = B$. The function f is called **bijective** if it is both injective and surjective; in this case, f is called a **bijection**. If f is bijective, then we may define the

inverse function $f^{-1} : B \to A$, where for $b \in B$, $f^{-1}(b)$ is defined to be the unique $a \in A$ such that $f(a) = b$.

If $f : A \to B$ and $g : B \to C$ are functions, we denote by $g \circ f$ their composition, that is, the function that sends $a \in A$ to $g(f(a)) \in C$. Function composition is associative; that is, for functions $f : A \to B$, $g : B \to C$, and $h : C \to D$, we have $(h \circ g) \circ f = h \circ (g \circ f)$. Thus, we can simply write $h \circ g \circ f$ without any ambiguity. More generally, if we have functions $f_i : A_i \to A_{i+1}$ for $i = 1, \ldots, n$, where $n \geq 2$, then we may write their composition as $f_n \circ \cdots \circ f_1$ without any ambiguity. As a special case of this, if $A_i = A$ and $f_i = f$ for $i = 1, \ldots, n$, then we may write $f_n \circ \cdots \circ f_1$ as f^n. It is understood that $f^1 = f$, and that f^0 is the identity function on A. If f is a bijection, then so is f^n for any non-negative integer n, the inverse function of f^n being $(f^{-1})^n$, which one may simply write as f^{-n}.

Binary operations

A **binary operation** $\star$ on a set S is a function from $S \times S$ to S, where the value of the function at $(a, b) \in S \times S$ is denoted $a \star b$.

A binary operation $\star$ on S is called **associative** if for all $a, b, c \in S$, we have $(a \star b) \star c = a \star (b \star c)$. In this case, we can simply write $a \star b \star c$ without any ambiguity. More generally, for $a_1, \ldots, a_n \in S$, where $n \geq 2$, we can write $a_1 \star \cdots \star a_n$ without any ambiguity.

A binary operation $\star$ on S is called **commutative** if for all $a, b \in S$, we have $a \star b = b \star a$. If the binary operation $\star$ is both associative and commutative, then not only is the expression $a_1 \star \cdots \star a_n$ unambiguous, but its value remains unchanged even if we re-order the a_i.