Appendix A
Mathematical Background

In this appendix, we review the basic notions, concepts and facts of logic, set theory, algebra, analysis, and formal language theory that are used throughout this book. For further details see, for example, [105, 134, 138] for logic, [61, 56, 88] for set theory, [40, 74, 119] for algebra, [51] for formal languages, and [77, 142] for mathematical analysis.

Propositional Calculus P

Syntax

• An expression of $P$ is a finite sequence of symbols. Each symbol denotes either an individual constant or an individual variable, or it is a logic connective or a parenthesis. Individual constants are denoted by $a, b, c, \ldots$ (possibly indexed). Individual variables are denoted by $x, y, z, \ldots$ (possibly indexed). Logic connectives are $\lor, \land, \Rightarrow, \Leftrightarrow$, and $\neg$; they are called the disjunction, conjunction, implication, equivalence, and negation, respectively. Punctuation marks are parentheses.

• Not every expression of $P$ is well-formed. An expression of $P$ is well formed if it is either 1) an individual-constant or individual-variable symbol, or 2) one of the expressions $F \lor G$, $F \land G$, $F \Rightarrow G$, $F \Leftrightarrow G$, and $\neg F$, where $F$ and $G$ are well-formed expressions of $P$. A well-formed expression of $P$ is called a sentence.

Semantic

• The intended meanings of the logic connectives are: “or” ($\lor$), “and” ($\land$), “implies” ($\Rightarrow$), “if and only if” ($\Leftrightarrow$), “not” ($\neg$).

• Let $\{\top, \bot\}$ be a set. The elements $\top$ and $\bot$ are called logic values and stand for “true” and “false,” respectively. Often, 1 and 0 are used instead of $\top$ and $\bot$, respectively.

• Any sentence has either the truth value $\top$ or $\bot$. A sentence is said to be true if its truth value is $\top$, and false if its truth value is $\bot$. Individual constants and individual variables obtain their truth values by assignment. When logic connectives combine sentences into new sentences, the truth value of the new sentence is determined by truth values of its component sentences. Specifically, let $E$ and $F$ be sentences. Then:

  - $\neg E$ is true if $E$ is false, and $\neg E$ is false if $E$ is true.
  - $E \lor F$ is false if both $E$ and $F$ are false; otherwise $E \lor F$ is true.
  - $E \land F$ is true if both $E$ and $F$ are true; otherwise $E \land F$ is false.
  - $E \Rightarrow F$ is false if $E$ is true and $F$ is false; otherwise $E \Rightarrow F$ is true.
- $E \iff F$ is true if both $E$ and $F$ are either true or false; else, $E \iff F$ is false.
- The following hold: $\neg (E \lor F) \iff (\neg E) \land (\neg F)$ and $\neg (E \land F) \iff (\neg E) \lor (\neg F)$.

**First-Order Logic $L$**

**Syntax**
- An expression of $L$ is a finite sequence of symbols, where each symbol is an individual-constant symbol (e.g., $a, b, c$), an individual-variable symbol (e.g., $x, y, z$), a logic connective ($\lor, \land, \Rightarrow, \Leftrightarrow, \neg$), a function symbol (e.g., $f, g, h$), a predicate symbol (e.g., $P, Q, R$), a quantification symbol ($\forall, \exists$), or a punctuation mark (e.g., colon, parenthesis). (Predicates are also called relations.)
- We are only interested in the well-formed expressions of $L$. To define these, we need two definitions. First, a term is either 1) an individual-constant or individual-variable symbol, or 2) a function symbol applied to terms (e.g., $f(a, x)$). Second, an atomic formula is a predicate symbol applied to terms (e.g., $P(y, f(a, x))$). Finally, we say that an expression of $L$ is well formed if it is either 1) an atomic formula, or 2) one of the expressions $F \lor G, F \land G, F \Rightarrow G, F \Leftrightarrow G, \neg F, \forall \tau F$, and $\exists \tau F$, where $F$ and $G$ are well-formed expressions of $L$ and $\tau$ is an individual-variable symbol. A logic expression of $L$ that is well formed is called a formula.

**Semantic**
- The intended meanings of the quantification symbols are: “for all” ($\forall$), “exists” ($\exists$). For the meanings of logic connectives, see Propositional Calculus $P$ above.
- The truth value of a formula is determined as follows. Let $E$ and $F$ be formulas. Then:
  - $\forall \tau F$ is true if $F$ is true for every possible assignment of a value to $\tau$.
  - $\exists \tau F$ is true if $F$ is true for at least one possible assignment of a value to $\tau$.
  - For the truth values of $F \lor G, F \land G, F \Rightarrow G, F \Leftrightarrow G, \neg F$, see Propositional Calculus $P$ above.

**Sets**

**Basics**
- Given any objects $a_1, \ldots, a_n$, the set containing $a_1, \ldots, a_n$ as its only elements is denoted by \{ $a_1, \ldots, a_n$ \}. More generally, given a property $P$, the set of those elements having the property $P$ is written as $\{ x \mid P(x) \}$. If an element $x$ is in a set $A$, we say that $x$ is a member of $A$ and write $x \in A$; otherwise, we write $x \notin A$ and say that $x$ is not a member of $A$. The set with no members is called the empty set and denoted by $\emptyset$.
- For sets $A$ and $B$, we say that $A$ is a subset of $B$, written $A \subseteq B$, if each member of $A$ is also a member of $B$. A set $A$ is a proper subset of $B$, written $A \subset B$, if $A \subseteq B$ but there is a member of $B$ not in $A$. Instead of $\subseteq$ we also write $\subset$.
- Sets $A$ and $B$ are equal, written $A = B$, if $A \subseteq B$ and $B \subseteq A$.
- Given a set $A = \{ x_i \mid i \in I \}$, the set $I$ is called the index set of $A$.
- By $(a_1, \ldots, a_n)$, or also by $(a_1, \ldots, a_n)$, we denote the ordered $n$-tuple of objects $a_1, \ldots, a_n$. When $n = 2$, the $n$-tuple is called the ordered pair. Two ordered $n$-tuples $(a_1, \ldots, a_n)$ and $(b_1, \ldots, b_n)$ are equal, denoted by $(a_1, \ldots, a_n) = (b_1, \ldots, b_n)$, if $a_i = b_i$ for $i = 1, \ldots, n$. 
Operations on Sets

- The **union** of sets \( A \) and \( B \), written as \( A \cup B \), is the set of elements that are members of at least one of \( A \) and \( B \).
- The **intersection** of sets \( A \) and \( B \), written as \( A \cap B \), is the set of elements that are members of both \( A \) and \( B \). We say that \( A \) and \( B \) are **disjoint** if \( A \cap B = \emptyset \).
- The **difference** of sets \( A \) and \( B \), written as \( A - B \), is the set of those members of \( A \) which are not in \( B \).
- If \( A \subseteq B \), then the **complement of \( A \) with respect to \( B \)** is the set \( B - A \).
- The **power set** of a set \( A \) is the set of all subsets of \( A \) and is denoted by \( 2^A \).
- The **Cartesian product** of a finite sequence of sets \( A_1, \ldots, A_n \) is the set of all ordered \( n \)-tuples \((a_1, \ldots, a_n)\), where \( a_i \in A_i \) for each \( i \). In this case it is denoted by \( A_1 \times \cdots \times A_n \). If \( A_1 = \ldots = A_n = A \), the Cartesian product is denoted by \( A^n \). By convention, \( A^1 \) stands for \( A \).

Relations

**Basics**

- An \( n \)-ary **relation** on a set \( A \) is a subset of \( A^n \). When \( n = 2 \) we say that the relation is **binary**, or in short, a **relation**. If \( R \) is a relation, we write \( xRy \) to indicate that \((x, y) \in R \). A 1-ary relation on \( A \) is a subset of \( A \), and is called a **property** on \( A \).
- A relation \( R \) on \( A \) is:
  - **reflexive** if \( xRx \) for each \( x \in A \).
  - **irreflexive** if \( xRx \) for no \( x \in A \).
  - **symmetric** if \( xRy \) implies \( yRx \), for arbitrary \( x, y \in A \).
  - **asymmetric** if \( xRy \) implies that not \( yRx \), for arbitrary \( x, y \in A \).
  - **anti-symmetric** if \( xRy \) and \( yRx \) imply \( x = y \), for arbitrary \( x, y \in A \).
  - **transitive** if \( xRy \) and \( yRz \) imply \( xRz \), for arbitrary \( x, y, z \in A \).

**Ordered Sets**

- A **preordered set** is a pair \( (A, R) \), where \( A \) is a set and \( R \) a binary relation on \( A \), such that (i) \( R \) is reflexive, and (ii) \( R \) is transitive. In this case, we say that \( R \) is a **preorder** on \( A \). Two elements \( x, y \in A \) are incomparable by \( R \) (in short, \( R \)-incomparable) if neither \( xRy \) nor \( yRx \).
- A **partially ordered set** is a pair \( (A, R) \), where \( A \) is a set and \( R \) a binary relation on \( A \), such that (i) \( R \) is reflexive, (ii) \( R \) is transitive, and (iii) \( R \) is anti-symmetric. In this case, we say that \( R \) is a **partial order** on \( A \). A partial order is often denoted by \( \leq, \leq, \preceq, \succeq \) or any other symbol indicating the properties of this order.
- Let \((A, \preceq)\) be a partially ordered set. The relation \( \preceq \) on \( A \) is the **strict partial order** corresponding to \( \preceq \) if \( a \preceq b \iff a < b \land a \neq b \), for arbitrary \( a, b \in A \). We say that \( \preceq \) is the **irreflexive reduction** of \( \preceq \). Conversely, \( \preceq \) is the reflexive closure of \( \preceq \), since \( a \preceq b \iff a \preceq b \vee a = b \).
- Let \((A, \preceq)\) be a partially ordered set and \( a, b \in A \). When \( a \preceq b \), we say that \( a \) is smaller than or equal to (or lower than or equal to) \( b \). Correspondingly, we say that \( b \) is larger than or equal to (or higher than or equal to) \( a \). When \( a \prec b \), we say that \( a \) is smaller than (or lower than, or below) \( b \). Correspondingly, we say that \( b \) is larger than (or higher than, or above) \( a \).
- Let \((A, \preceq)\) be a partially ordered set and \( a, b, c, d \in A \). Then we say:
  - \( a \) is \( \preceq \)-**minimal** if \( x \preceq a \) implies \( x = a \) for \( \forall x \in A \) (nothing in \( A \) is smaller than \( a \)).
  - \( b \) is \( \preceq \)-**least** if \( b \preceq x \) for \( \forall x \in A \) (\( b \) is smaller than any other in \( A \)).
  - \( c \) is \( \preceq \)-**maximal** if \( c \preceq x \) implies \( x = c \) for \( \forall x \in A \) (nothing in \( A \) is greater than \( c \)).
  - \( d \) is \( \preceq \)-**greatest** if \( x \preceq d \) for \( \forall x \in A \) (\( d \) is greater than any other in \( A \)).
When the relation $\leq$ is understood, we can drop the prefix “$\leq$”. The least and greatest elements are called the zero (0) and unit (1) element, respectively.

Let $(\mathcal{A}, \leq)$ be a partially ordered set, $\mathcal{B} \subseteq \mathcal{A}$, and $u, v, w, z \in \mathcal{A}$. Then we say:

- $u$ is a $\leq$-upper bound of $\mathcal{B}$ if $x \leq u$ for all $x \in \mathcal{B}$.
- $v$ is a $\leq$-least upper bound (or $\leq$-lub) of $\mathcal{B}$ if $v$ is a $\leq$-upper bound of $\mathcal{B}$ and $v \leq u$ for every $\leq$-upper bound $u$ of $\mathcal{B}$.
- $w$ is a $\leq$-lower bound of $\mathcal{B}$ if $w \leq x$ for all $x \in \mathcal{B}$.
- $z$ is a $\leq$-greatest lower bound (or $\leq$-glb) of $\mathcal{B}$ if $z$ is a $\leq$-lower bound of $\mathcal{B}$ and $w \leq z$ for every $\leq$-lower bound $w$ of $\mathcal{B}$.

When the relation $\leq$ is understood, we can drop the prefix “$\leq$”.

- A lattice is a partially ordered set $(\mathcal{A}, \preceq)$ in which any two elements have an lub and a glb. The lub of $a, b \in \mathcal{A}$ is denoted by $a \lor b$, and the glb by $a \land b$. An upper semi-lattice is a partially ordered set $(\mathcal{A}, \preceq)$ in which any two elements have a lub (but not necessarily a glb).
- A linearly (or totally) ordered set is a partially ordered set $(\mathcal{A}, \preceq)$ such that for all $x, y \in \mathcal{A}$ either $x \preceq y$ or $y \preceq x$. In this case, we say that $\preceq$ is a linear order on $\mathcal{A}$.
- A well-ordered set is a linearly ordered set $(\mathcal{A}, \preceq)$ such that every non-empty subset of $\mathcal{A}$ has a $\preceq$-least element. We say that such a $\preceq$ is a well-order on $\mathcal{A}$.
- Associated with every well-ordered set $(\mathcal{A}, \preceq)$ is the corresponding Principle of Complete Mathematical Induction: If $P$ is a property such that, for any $b \in \mathcal{A}$, $P(b)$, whenever $P(a)$ for all $a \in \mathcal{A}$ such that $a \preceq b$, then $P(x)$ for all $x \in \mathcal{A}$. When $\mathcal{A}$ is infinite, a proof using this principle is called a proof by transfinite induction.

Equivalence Relations

- A relation $R$ on $\mathcal{A}$ is an equivalence relation if (i) $R$ is reflexive, (ii) $R$ is symmetric, and (iii) $R$ is transitive. In this case, the $R$-equivalence class of $a \in \mathcal{A}$ is the set $\{x \in \mathcal{A} \mid xRa\}$. Elements of the $R$-equivalence class of $a$ are said to be $R$-equivalent to $a$. If $\mathcal{C}$ is an equivalence class, any element of $\mathcal{C}$ is called a representative of the class $\mathcal{C}$.
- A partition of $\mathcal{A}$ is any collection $\{\mathcal{A}_i \mid i \in \mathcal{I}\}$ of nonempty subsets of $\mathcal{A}$ such that (i) $\mathcal{A} = \bigcup_{i \in \mathcal{I}} \mathcal{A}_i$, and (ii) $\mathcal{A}_i \cap \mathcal{A}_j = \emptyset$, for all $i, j \in \mathcal{I}$ with $i \neq j$. So, $\mathcal{A}$ is the disjoint union of the sets in the partition.
- Any equivalence relation on $\mathcal{A}$ is associated with a partition of $\mathcal{A}$, and vice versa. If $R$ is an equivalence relation on $\mathcal{A}$, then the associated partition of $\mathcal{A}$ is called the quotient set of $\mathcal{A}$ relative to $R$ and is denoted by $\mathcal{A}/R$. The members of $\mathcal{A}/R$ are the $R$-equivalence classes of $\mathcal{A}$. The function $f : \mathcal{A} \to \mathcal{A}/R$ that associates with each element $a \in \mathcal{A}$ the $R$-equivalence class of $a$ is called the natural map of $\mathcal{A}$ relative to $R$.
- An equivalence relation is often denoted by $\sim, \simeq, \equiv$, or any other symbol indicating the properties of this relation.

Functions

Basics

- A total function $f$ from $\mathcal{A}$ into $\mathcal{B}$ is a triple $(\mathcal{A}, \mathcal{B}, f)$ where $\mathcal{A}$ and $\mathcal{B}$ are nonempty sets and for every $x \in \mathcal{A}$ there is a unique member, denoted by $f(x)$, of $\mathcal{B}$. We call $\mathcal{A}$ the domain of $f$ and denote it by $\text{dom}(f)$. The set $\mathcal{B}$ we call the co-domain of $f$ and denote it by $\text{codom}(f)$. We usually write $f : \mathcal{A} \to \mathcal{B}$ instead of $(\mathcal{A}, \mathcal{B}, f)$. A function is also called a mapping.
- In specifying a definition of $f : \mathcal{A} \to \mathcal{B}$ we say that $f$ is well-defined if we are assured that $f$ is single-valued, i.e., with each member of $\mathcal{A}$, $f$ associates a unique member of $\mathcal{B}$.
- When the domain of a function consists of ordered $n$-tuples, the function is said to be of $n$ arguments. A (total) function of $n$ arguments on a set $S$ is a function $f$ whose domain is $S^n$. We write $f(a_1, \ldots, a_n)$ instead of $f((a_1, \ldots, a_n))$. 

• Let \( f : \mathcal{A} \to \mathcal{B} \) and \( \mathcal{C} \subseteq \mathcal{A} \). The image of \( \mathcal{C} \) under \( f \) is a set denoted by \( f(\mathcal{C}) \) and defined by \( f(\mathcal{C}) = \{ f(x) \mid x \in \mathcal{C} \} \). In particular, \( f(\mathcal{A}) \) is called the range of \( f \) and denoted by \( \text{rng}(f) \).

• A function \( f : \mathcal{A} \to \mathcal{B} \) is:
  - injective if \( f(x) \neq f(y) \) whenever \( x \neq y \); we also say that such an \( f \) is an injection.
  - surjective if \( f(\mathcal{A}) = \mathcal{B} \); we also say that such an \( f \) is a surjection.
  - bijective if \( f \) is injective and surjective; we also say that such an \( f \) is a bijection.

• An element \( a \in \mathcal{A} \) is called the fixed point of a function \( f : \mathcal{A} \to \mathcal{A} \) if \( f(a) = a \).

• Let \( f : \mathcal{A} \to \mathcal{B} \) and \( \mathcal{C} \subseteq \mathcal{A} \). Then a function \( g : \mathcal{C} \to \mathcal{B} \) is the restriction of \( f \) to \( \mathcal{C} \) if \( g(x) = f(x) \) for each \( x \in \mathcal{C} \). The restriction of \( f \) to \( \mathcal{C} \) is denoted by \( f|_\mathcal{C} \). In that case \( f \) is the extension of \( g \) to \( \mathcal{A} \).

• An element \( \mathcal{A} \) is the characteristic function of \( \mathcal{A} \) is the function \( \chi_\mathcal{A} : \mathcal{U} \to \{0, 1\} \) such that \( \chi_\mathcal{A}(u) = 1 \) if \( u \in \mathcal{A} \) and \( \chi_\mathcal{A}(u) = 0 \) if \( u \notin \mathcal{A} \).

• Let \( \mathcal{A} \) and \( \mathcal{U} \) be sets, and let \( \mathcal{A} \subseteq \mathcal{U} \). The characteristic function of \( \mathcal{A} \) is the function \( \chi_\mathcal{A} : \mathcal{U} \to \{0, 1\} \) such that \( \chi_\mathcal{A}(u) = 1 \) if \( u \in \mathcal{A} \) and \( \chi_\mathcal{A}(u) = 0 \) if \( u \notin \mathcal{A} \).

• A set \( \mathcal{A} \) is:
  - finite if either \( \mathcal{A} = \emptyset \) or \( \mathcal{A} \simeq \{1, 2, \ldots, n\} \) for some natural \( n \);
  - infinite if it is not finite;
  - countable (or enumerable, or denumerable) if \( \mathcal{A} \simeq \mathcal{B} \) for some \( \mathcal{B} \subseteq \mathbb{N} \); when \( \mathcal{B} = \mathbb{N} \), the set \( \mathcal{A} \) is said to be countably infinite;
  - uncountable if it is not countable.

• If a set \( \mathcal{A} \) is infinite, then there is \( \mathcal{B} \subseteq \mathcal{A} \) such that \( \mathcal{B} \simeq \mathcal{A} \).

• Any subset of a countable set is countable. The union of countably many countable sets is countable. The Cartesian product of two countable sets is countable.

• Let \( n \) be a natural number, \( \aleph_0 = |\mathbb{N}| \), and \( c = |\mathbb{R}| \) the cardinality of continuum. Then: \( \aleph_0 + n = \aleph_0 \), \( \aleph_0 + \aleph_0 = \aleph_0 \), \( n \cdot \aleph_0 = \aleph_0 \), \( \aleph_0^c = \aleph_0 \), \( c + \aleph_0 = c \), and \( \aleph_0 \cdot c = c \).

• A sequence is a function \( f \) defined on \( \mathbb{N} \), the set of natural numbers. If we write \( f(n) = x_n \), for \( n \in \mathbb{N} \), we also denote the sequence \( f \) by \( \{x_n\} \) or by \( x_0, x_1, x_2, \ldots \). When \( x_n \in \mathcal{A} \) for all \( n \in \mathbb{N} \), we say that \( \{x_n\} \) is a sequence of elements of \( \mathcal{A} \). The elements of any at most countable set can be arranged in a sequence.

• The cardinality of \( \mathcal{B}^{\mathcal{A}} \), the set of all functions mapping \( \mathcal{A} \) into \( \mathcal{B} \), is \( |\mathcal{B}|^{|\mathcal{A}|} \).

### Operations and Algebraic Structures

• An \( n \)-ary operation on a set \( \mathcal{A} \) is a function \( * : \mathcal{A}^n \to \mathcal{A} \). When \( n = 2 \), we say that the operation is binary. In this case we write \( a \ast b \) instead of \( *(a, b) \) when \( n = 1 \), the operation is said to be unary and we write \( a^* \) instead of \( *(a) \).

• A binary operation on a set \( \mathcal{A} \) is:
  - associative if \( a \ast (b \ast c) = (a \ast b) \ast c \), for all \( a, b, c \in \mathcal{A} \).
  - commutative if \( a \ast b = b \ast a \), for all \( a, b \in \mathcal{A} \).
• A semigroup is a pair $(\mathcal{A}, \ast)$, where $\ast$ is an associative binary operation on $\mathcal{A}$.
• A group is a semigroup $(\mathcal{A}, \ast)$ satisfying the following requirements:
  – there exists an element $e \in \mathcal{A}$ such that $a \ast e = e \ast a = a$, for all $a \in \mathcal{A}$ ($e$ is called an identity of $\mathcal{A}$);
  – for each $a \in \mathcal{A}$ there exists an element $a^{-1} \in \mathcal{A}$ such that $a \ast a^{-1} = a^{-1} \ast a = e$ ($a^{-1}$ is called an inverse of $a$).

Natural Numbers

• Natural numbers are $0, 1, 2, \ldots$. The set of all natural numbers is denoted by $\mathbb{N}$. The cardinal number of $\mathbb{N}$ is denoted by $\aleph_0$ (aleph zero).
• A prime is a natural number greater than 1 that has no positive divisors other than 1 and itself. There are infinitely many primes. A natural number greater than 1 that is not a prime is called a composite.
• The Fundamental Theorem of Arithmetic states: Any positive integer ($\neq 1$) can be expressed as a product of primes; this expression is unique except for the order in which the primes occur. Thus, any positive integer $n(\neq 1)$ can be written as $p_1^{\alpha_1} p_2^{\alpha_2} \cdots p_r^{\alpha_r}$, where $p_1, p_2, \ldots, p_r$ are primes satisfying $p_1 < p_2 < \ldots < p_r$, and $\alpha_1, \alpha_2, \ldots, \alpha_r$ are positive integers.
• The Principle of Mathematical Induction is: Any subset of $\mathbb{N}$ that contains 0 and, for every natural $k$, contains $k + 1$ whenever it contains $k$, is equal to $\mathbb{N}$.
• The set $(\mathbb{N}, <)$ is well ordered. It is also denoted by $\omega$.
• The Principle of Complete Mathematical Induction: Any subset of $\omega$ that, for every natural $k$, contains $k + 1$ whenever it contains $k$, is equal to $\omega$.
• The set of all subsets of $\mathbb{N}$, i.e., the set $2^{\mathbb{N}}$, is uncountable. Its cardinality is $2^{\aleph_0}$. This is equal to $c = |\mathbb{R}|$, the cardinality of continuum.
• Functions $f : \mathbb{N} \rightarrow \mathbb{N}, k \geq 1$, are called numerical.
• The set $\mathbb{N}^\mathbb{N}$ of all functions $f : \mathbb{N} \rightarrow \mathbb{N}$ is uncountable: $|\mathbb{N}^\mathbb{N}| = 2^{\aleph_0} = c$. In particular, the set $\{0, 1\}^\mathbb{N}$ of all characteristic functions $\chi : \mathbb{N} \rightarrow \{0, 1\}$ is equinumerous to the set $2^\mathbb{N}$. Since each $\chi$ is identified with an infinite sequence of 0s and 1s, the set of all infinite binary sequences is also uncountable.
• The join of two sets $\mathcal{A}, \mathcal{B} \subseteq \mathbb{N}$ is the set denoted by $\mathcal{A} \oplus \mathcal{B}$ and defined by $\mathcal{A} \oplus \mathcal{B} \overset{\text{def}}{=} \{2x + 1 \mid x \in \mathcal{A}\} \cup \{2y \mid y \in \mathcal{B}\}$. Informally, $\mathcal{A} \oplus \mathcal{B}$ “remembers” every member of $\mathcal{A}$ and every member of $\mathcal{B}$.

Formal Languages

Basics

• An alphabet $\Sigma$ is a finite non-empty set of abstract symbols.
• A word of length $k \geq 0$ over the alphabet $\Sigma$ is a finite sequence $x_1, \ldots, x_k$ of symbols in $\Sigma$. A word $x_1, \ldots, x_k$ is usually written without commas, i.e., as $x_1 \ldots x_k$.
• The length of a word $w$ is denoted by $|w|$. The word of length zero is called the empty word and denoted by $\varepsilon$.
• If $w = x_1 \ldots x_k$ is a word, then the word $w^R = x_k \ldots x_1$ is called the reversal of $w$.
• Two words $x_1 \ldots x_r$ and $y_1 \ldots y_s$ over the alphabet $\Sigma$ are equal, written $x_1 \ldots x_r = y_1 \ldots y_s$, if $r = s$ and $x_i = y_i$ for each $i$.
• Let $x$ and $y$ be words over the alphabet $\Sigma$. The word $x$ is a subword of $y$ if $y = uxxv$ for some words $u$ and $v$. The word $x$ is a proper subword of $y$ if $x$ is a subword of $x$, but $x \neq y$. 
Let $x$ and $y$ be words over the alphabet $\Sigma$. The word $x$ is a prefix of $y$, written $x \subseteq y$, if $y = xv$ for some word $v$. The word $x$ is a proper prefix of $y$, written $x \subset y$, if $x$ is a prefix of $y$, but $x \neq y$.

- The set of all words, including $\epsilon$, over the alphabet $\Sigma$ is denoted by $\Sigma^*$.
- The set $\Sigma^*$ is countably infinite.
- Each subset $L \subseteq \Sigma^*$ is called a formal language (or language in short).

### Operations on Languages

- If $x = x_1 \ldots x_r$ and $y_1 \ldots y_s$ are words, then $xy$, called the concatenation of $x$ and $y$, is the word $x_1 \ldots x_ry_1 \ldots y_s$.
- For languages $L_1$ and $L_2$, the concatenation (or product) of $L_1$ and $L_2$ is a language denoted by $L_1L_2$ and defined by $L_1L_2 = \{xy | x \in L_1 \land y \in L_2\}$.
- For a language $L$ let $L^0 = \{\epsilon\}$ and, for each $n \geq 1$, let $L^n = L^{n-1}L$. The Kleene star of $L$ is the language denoted by $L^*$ and defined by $L^* = \bigcup_{i=0}^{\infty} L^i$. Similarly, Kleene plus of $L$ is the language denoted by $L^+$ and defined by $L^+ = \bigcup_{i=1}^{\infty} L^i$. In particular, for the alphabet $\Sigma$, the language $\Sigma^n$ contains all words of length $n$ over $\Sigma$, and $\Sigma^*$ contains all words over $\Sigma$.

### Orders on Languages

- Let $\leq$ be a linear order on the alphabet $\Sigma$. A lexicographic order $\leq_{\text{lex}}$ on $\Sigma^n$, induced by $\leq$, is the order in which $x_1 \ldots x_n <_{\text{lex}} y_1 \ldots y_n$ if there is a $j$, $1 \leq j \leq n$, such that $x_i = y_i$ for each $i = 1, \ldots, j-1$, but $x_j < y_j$.
- A shortlex order on a language $L \subseteq \Sigma^*$ is the order in which words of $L$ are primarily ordered by their increasing length, and words of the same length are then lexicographically ordered. The shortlex order is a well-order on $\Sigma^*$ and, consequently, on $L$.
Appendix B
Notation Index

Frontmatter and Chapter 2

Box detour x
NB nota bene x
A, B, C, ... sets 13, 298
x ∈ A x is a member of A 13, 298
A ⊆ B A is a subset of B 13, 298
A the complement of A 14, 298
A ∪ B union of A and B 14, 298
A ∩ B intersection of A and B 14, 298
A − B set theoretic difference of A and B 14, 298
2^A power set of A 14, 298
(x, y) ordered pair where x is the first and y the second member 14, 298
A × B Cartesian product of A and B 14, 298
|A| cardinality of A 15, 300
N the set of all natural numbers 302
N0 |N|, the least transfinite cardinal 16, 302
N transfinite cardinal 16
≤ is less than or equal to (used for numbers) 14
ω well-ordered set (N, ≤), the least transfinite ordinal 17
R the set of all real numbers 16
c |R|, the cardinality of continuum 16
iff if and only if 17
Ω set of all ordinal numbers (paradoxical) 17
U set of all sets (paradoxical) 17
R Russell’s set of all sets not containing themselves (paradoxical) 17
Z the set of all integers 19
PM Principia Mathematica 25

Chapter 3

f.a.s. formal axiomatic system 31
F a particular f.a.s., the theory developed in this f.a.s. 31
F meta-theory, the theory about F 59
a, b, c, ... symbols for individual constants in f.a.s. 32
x, y, z, ... symbols for individual variables in f.a.s. 32
f, g, h, ... symbols for functions in f.a.s. 32
p, q, r, ... symbols for predicates in f.a.s. 32
F, G, H, ... symbols for formulas in f.a.s. 32
∧ and 32, 297
∨ or 32, 297
¬ not 32, 297
⇒ implies 32, 297
⇔ is equivalent to 32, 297
∃ there exists 32, 298
∀ for all 32, 298
 hẹ rule of inference 33
MP Modus Ponens 33
Gen Generalization 33
P ⊢ F F is derivable (formally provable) from the set of premises P 34
∗ F F F is a theorem of F, i.e., derivable (formally provable) in F 34
S mathematical structure 35
ι mapping that assigns meaning to F in S 36
P |=} F F is a logical consequence of the set of premises P 38
|=} F F is valid in F 39
P Propositional Calculus 36, 297
L First-order Logic 39, 298
A Formal Arithmetic 41
ZF Zermelo-Fraenkel axiomatic set theory 42, 44
ZFC ZF with Axiom of Choice 42
NBG Von Neumann-Bernays-Gödel’s axiomatic set theory 43, 45

Chapter 4

M formal axiomatic system for all mathematics 53, 54
D_Entsch decision procedure for M 53, 55
γ(Χ) Gödel number of syntactic object X 59
G Gödel’s formula 60
CH Continuum Hypothesis 59
GCH Generalized Continuum Hypothesis 59

Chapter 5

f, g, ... total functions from \( \mathbb{N}^k \) to \( \mathbb{N} \) 72
ζ zero function 73, 74
σ successor function 73, 74
π projection function 73, 74
[...x...] expression containing variable x 77
μX[...x...] μ-operation, the least x such that [...x...] = 0 and [...z...]↓ for z < x 73, 75
\( \delta(f) \) system of equations defining a function f 76
λx[...x...] partial function of x, defined by [...x...] 77, 78
\( \rightarrow^\alpha \) α-conversion, renaming of variables in a λ-term 78
\( \rightarrow^\beta \) β-contraction, application of a λ-term 78
\( \rightarrow^\beta \) sequence (composition) of β-reductions 78
β-nf β-normal form 78
TM Turing machine 80, 101
TP Turing program 80, 103
Σ input alphabet 80, 103
→ production 82, 84
\( \langle M \rangle \) code of an abstract computing machine M 86
### Chapter 6

<table>
<thead>
<tr>
<th>Symbol</th>
<th>Definition</th>
</tr>
</thead>
<tbody>
<tr>
<td>TM</td>
<td>Turing machine</td>
</tr>
<tr>
<td>(T)</td>
<td>a TM (basic model)</td>
</tr>
<tr>
<td>(T_n)</td>
<td>the TM with index (code number) (n)</td>
</tr>
<tr>
<td>(\Gamma)</td>
<td>tape alphabet</td>
</tr>
<tr>
<td>(\sqcup)</td>
<td>empty space</td>
</tr>
<tr>
<td>(\Sigma)</td>
<td>input alphabet</td>
</tr>
<tr>
<td>(\Sigma^*)</td>
<td>the set of all words over (\Sigma)</td>
</tr>
<tr>
<td>(Q)</td>
<td>set of states</td>
</tr>
<tr>
<td>(q_1)</td>
<td>initial state</td>
</tr>
<tr>
<td>(F)</td>
<td>set of final states</td>
</tr>
<tr>
<td>TP</td>
<td>Turing program</td>
</tr>
<tr>
<td>(\delta)</td>
<td>a Turing program</td>
</tr>
<tr>
<td>(\delta_n)</td>
<td>the TP with index (code number) (n)</td>
</tr>
<tr>
<td>(q_{yes})</td>
<td>a final state</td>
</tr>
<tr>
<td>(q_{no})</td>
<td>a non-final state</td>
</tr>
<tr>
<td>(\Delta)</td>
<td>matrix describing (\delta)</td>
</tr>
<tr>
<td>(V)</td>
<td>a Turing machine (generalized model)</td>
</tr>
<tr>
<td>(\langle T \rangle)</td>
<td>code of (T)</td>
</tr>
<tr>
<td>(U)</td>
<td>universal Turing machine</td>
</tr>
<tr>
<td>OS</td>
<td>operating system</td>
</tr>
<tr>
<td>RAM</td>
<td>random access machine</td>
</tr>
<tr>
<td>(\psi_f^{(k)}(x))</td>
<td>(k)-ary proper function of (T)</td>
</tr>
<tr>
<td>(\psi_{f_i}^{(k)}(x))</td>
<td>the (k)-ary proper function of (T_i)</td>
</tr>
<tr>
<td>(W_i)</td>
<td>domain of (\psi_{f_i}^{(k)}(x))</td>
</tr>
<tr>
<td>(\varepsilon)</td>
<td>empty word</td>
</tr>
<tr>
<td>(G_A)</td>
<td>generator of (A)</td>
</tr>
<tr>
<td>c.e.</td>
<td>computably enumerable</td>
</tr>
<tr>
<td>(L(T))</td>
<td>proper set of (T), i.e., language of (T)</td>
</tr>
<tr>
<td>(\mathcal{U})</td>
<td>universe, a large enough set</td>
</tr>
<tr>
<td>(\chi_A)</td>
<td>characteristic function of (A)</td>
</tr>
<tr>
<td>(D_A)</td>
<td>decider of (A)</td>
</tr>
<tr>
<td>(R_A)</td>
<td>recognizer of (A)</td>
</tr>
<tr>
<td>(\mathbb{P})</td>
<td>the set of prime numbers</td>
</tr>
</tbody>
</table>

### Chapter 7

<table>
<thead>
<tr>
<th>Symbol</th>
<th>Definition</th>
</tr>
</thead>
<tbody>
<tr>
<td>(\text{ind}(\varphi))</td>
<td>index set of a p.c. function (\varphi)</td>
</tr>
<tr>
<td>(\text{ind}(\mathcal{A}))</td>
<td>index set of a c.e. set (\mathcal{A})</td>
</tr>
</tbody>
</table>
Chapter 8

\(D\)  
decision problem 163

\(\langle d \rangle\)  
code of an instance \(d\) of a decision problem 163

\(L(\mathcal{D})\)  
language of decision problem \(D\) 164

\(\mathcal{D}_{\text{Halt}}\)  
Halting problem, “Does \(T\) halt on \(w\)?” 167

\(\mathcal{D}_{\text{H}}\)  
Halting problem, “Does \(T\) halt on \(\langle T \rangle\)?” 167

\(K_0\)  
universal language 167

\(K\)  
diagonal language 167

\(\overline{\mathcal{D}}\)  
complementary problem of a decision problem \(\mathcal{D}\) 171

\(\mathcal{D}_{\text{Halt}}\)  
non-Halting problem, “Does \(T\) never halt on \(w\)?” 170

\(\mathcal{D}_{\text{H}}\)  
non-Halting problem, “Does \(T\) never halt on \(\langle T \rangle\)?” 170

\(\mathcal{D}_{\text{Emp}}\)  
empty proper set problem, “Is \(L(T) = \emptyset\)?” 173

\(n\text{-BB}\)  
n-state busy beaver 173

\(\mathcal{D}_{\text{BB}}\)  
Busy Beaver Problem, “Is \(T\) a busy beaver?” 173

\(\text{PCP}\)  
Post’s Correspondence Problem 175

\(\mathcal{D}_{\text{PCP}}\)  
Post’s Correspondence Problem 175

\(\text{CFG}\)  
context-free grammar 177

\(\text{CFL}\)  
context-free language 177

\(\mathcal{D}_{K_1}\)  
“Is \(\text{dom}(\varphi)\) empty?” 179

\(\mathcal{D}_{\text{Fm}}\)  
“Is \(\text{dom}(\varphi)\) finite?” 179

\(\mathcal{D}_{\text{Inf}}\)  
“Is \(\text{dom}(\varphi)\) infinite?” 179

\(\mathcal{D}_{\text{Cof}}\)  
“Is \(A - \text{dom}(\varphi)\) finite?” (where \(\varphi: A \to B\)) 179

\(\mathcal{D}_{\text{Tot}}\)  
“Is \(\varphi\) total?” 179

\(\mathcal{D}_{\text{Ext}}\)  
“Can \(\varphi\) be extended to a total computable function?” 179

\(\mathcal{D}_{\text{Sur}}\)  
“Is \(\varphi\) surjective?” 179

Chapter 9

\(sw\)  
switching function 191

\(\leq_m\)  
is \(m\)-reducible to (set, decision problem, function) 197

\(\leq_1\)  
is 1-reducible to (set, decision problem, function) 201

\(\mathcal{D}_{\text{Entsch}}\)  
Entscheidungsproblem 203

\(\mathcal{C}\)  
class of all c.e. sets 208

Chapter 10

\(o\text{-TM}\)  
oracle Turing machine 224

\(T^*\)  
an \(o\text{-TM}\) (with no oracle set attached) 226

\(\langle T^* \rangle\)  
the code of \(T^*\) 227

\(T^*_i\)  
the \(o\text{-TM}\) with index \(i\) (and no oracle set attached) 227

\(O\text{-TM}\)  
\(o\text{-TM}\) with oracle set \(O\) attached 224

\(T^O\)  
an \(O\text{-TM}\) 226

\(T^O_i\)  
the \(O\text{-TM}\) with index \(i\) 227

\(o\text{-TP}\)  
oracle Turing program 224

\(\tilde{\delta}\)  
an \(o\text{-TP}\) (i.e., a transition function of an \(o\text{-TM}\)) 224

\(\tilde{\delta}_i\)  
the \(o\text{-TP}\) with index \(i\) 227

w.l.g.  
without loss of generality 228

\(\Psi^i\)  
proper functional of the \(o\text{-TM}\) \(T^*_i\) (with arity understood) 228

\(\Psi^O_i\)  
proper functional of the \(O\text{-TM}\) \(T^O_i\) (with arity understood) 228

\(O\text{-p.c.}\)  
partial \(O\text{-computable} (function) 229
### B Notation Index

<table>
<thead>
<tr>
<th>Symbol/Notation</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>$O$-c.e.</td>
<td>$O$-semi-decidable (set)</td>
</tr>
<tr>
<td>$O^\text{ind}(\phi)$</td>
<td>index set of the $O$-p.c. function $\phi$</td>
</tr>
<tr>
<td>$O^\text{ind}(S)$</td>
<td>index set of the $O$-c.e. set $S$</td>
</tr>
</tbody>
</table>

**Chapter 11**

- $\leq_T$ \quad is $T$-reducible to (set, decision problem, function) \quad 236
- $\equiv_T$ \quad is $T$-equivalent to (set, decision problem, function) \quad 240
- $<_T$ \quad is $\leq_T$ but not $\equiv_T$ to (set, decision problem, function) \quad 236
- $\deg(S)$ \quad $T$-degree (degree of unsolvability) of the set $S$ \quad 241
- $<$ \quad is lower than ($T$-degree) \quad 242

**Chapter 12**

- $S'$ \quad the $T$-jump of the set $S$ \quad 247
- $K^S$ \quad the same as $S'$ \quad 247
- $S^{(n)}$ \quad the $n$-th $T$-jump of the set $S$ \quad 249

**Chapter 13**

- $\mathcal{D}$ \quad the class of all $T$-degrees \quad 255
- $a, b, \ldots$ \quad $T$-degrees \quad 255
- $\leq$ \quad is lower than or equal to ($T$-degree) \quad 256
- $d'$ \quad the $T$-jump of $d$ \quad 256
- $d^{(n)}$ \quad the $n$-th $T$-jump of $d$ \quad 256
- $0$ \quad the $T$-degree of the set $\emptyset$ \quad 256
- $0^{(n)}$ \quad the $n$-th $T$-jump of $0$ \quad 256
- $x \subseteq y$ \quad the word $x$ is a prefix of the word $y$ \quad 260
- $x \subset y$ \quad the word $x$ is a proper prefix of the word $y$ \quad 260
- $\leq\text{lub}$ \quad the least upper bound of $T$-degrees \quad 262
- $\leq\text{glb}$ \quad the greatest lower bound of $T$-degrees \quad 262
- $A \oplus B$ \quad join of sets $A$ and $B$ \quad 252
- $\text{ucone}(d)$ \quad upper cone of $d$ \quad 252
- $\text{lcone}(d)$ \quad lower cone of $d$ \quad 252

**Chapter 14**

- $\leq_{\text{bt}}$ \quad is bounded truth-table-reducible to (set, decision problem, function) \quad 273
- $\leq_{\text{t}}$ \quad is truth-table-reducible to (set, decision problem, function) \quad 273
- $x \in! A$ \quad add (enumerate) $x$ into the set $A$ \quad 275
- $x \notin! A$ \quad ban $x$ from the set $A$ \quad 275

**Chapter 15**

- $\Sigma_n$ \quad arithmetical class \quad 285
- $\Pi_n$ \quad arithmetical class \quad 285
- $\Delta_n$ \quad arithmetical class \quad 285
- $\text{graph}(\phi)$ \quad graph of $\phi$ \quad 292
References

References


72. Hopcroft, J.E., Ullman, J.D.: Introduction to Automata Theory, Languages, and Computation. Addison-Wesley (1979)


References

316 References

146. Sacks, G.E.: The Recursively Enumerable Degrees are Dense. Annals of Mathematics 80(2), 300–312 (1964)


<table>
<thead>
<tr>
<th>Term</th>
<th>Page(s)</th>
</tr>
</thead>
<tbody>
<tr>
<td>λ-calculus, λ-definable function, λ-term</td>
<td>77–79</td>
</tr>
<tr>
<td>α-conversion, β-contraction, β-normal form, β-redex, β-reduction</td>
<td>77–79</td>
</tr>
<tr>
<td>abstraction, application, Church’s numeral</td>
<td>77–79</td>
</tr>
<tr>
<td>defining a numerical function, final, initial</td>
<td>77–79</td>
</tr>
<tr>
<td>μ-operation, see Recursive function</td>
<td></td>
</tr>
<tr>
<td>Abstract computing machine, basic instruction, finite effect</td>
<td>86, 205, 209</td>
</tr>
<tr>
<td>behavior, global, local, capability, code</td>
<td>86</td>
</tr>
<tr>
<td>components, internal configuration, resources, unrestricted</td>
<td>86</td>
</tr>
<tr>
<td>Ackermann, Adleman, Al-Khwarizmi</td>
<td>26, 60, 89</td>
</tr>
<tr>
<td>Aleph zero, Algorithm, characterization of the concept</td>
<td>15, 59, 71, 3.51, 69</td>
</tr>
<tr>
<td>Church’s, convincing, Gödel-Kleene’s</td>
<td>78, 71, 74</td>
</tr>
<tr>
<td>Herbrand-Gödel’s, Markov’s, Post’s, Turing’s</td>
<td>77, 85, 84, 82</td>
</tr>
<tr>
<td>description, environment, execution, existence, formally</td>
<td>71, 71, 71, 90</td>
</tr>
<tr>
<td>intuitively, non-existence, offline, online</td>
<td>4, 70, 89, 233, 233</td>
</tr>
<tr>
<td>problems about, quantum</td>
<td>175, 70</td>
</tr>
<tr>
<td>property, shrewd</td>
<td>194–195, 195</td>
</tr>
<tr>
<td>Arithmetical class of sets, hierarchy, relation, set</td>
<td>286, 283–292, 284</td>
</tr>
<tr>
<td>Arithmetization of a theory, of man’s reflection</td>
<td>59, 6</td>
</tr>
<tr>
<td>Aspiration, Axiom</td>
<td>63</td>
</tr>
<tr>
<td>Axiom fertility, independence</td>
<td>12, 11</td>
</tr>
<tr>
<td>logical, of a theory, of Abstraction, of Choice, of Class Existence, of Extensionality</td>
<td>33, 38, 39, 52, 10, 56, 13, 15, 17, 18, 20, 45, 13, 24, 44, 45</td>
</tr>
<tr>
<td>© Springer-Verlag Berlin Heidelberg 2015</td>
<td></td>
</tr>
</tbody>
</table>
of Infinity, 44
of Mathematical Induction, 42, 47
of Pair, 44
of Power Set, 44
of Regularity, 44
of Separation (schema), 44
of Substitution (schema), 44
of Union, 44
proper (= non-logical), 33, 35, 38, 40, 52
schema, 39
Axiomatic
evident, 11
hypothetical, 11, 12
Axiomatic method
axioms, 10
basic notions, 10
Euclidean geometry, 9
initial theory, 10
limitations of, 58
Axiomatic set theory
\textbf{NBG}, 43, 45, 54
\textbf{ZFC}, 42, 44, 54
Axiomatic system, 9, 10
Babbage, 6, 70, 119
analytical engine, 6
Basic notions
of a theory, 10, 33, 34, 38, 43, 45, 46, 143
of computation, 72, 130, 143, 161
of generation, 127
Behavior
of a Turing program (machine)
global, 146, 147, 152, 209
local, 146, 147, 152, 209
of any mechanism, 209
Beltrami, 11
Bernays, 26, 43
Bolyai, 12
Boole, 23, 39
Boole’s
The Laws of Thought, 23
Boolean algebra, 23
Brouwer, 21
Burali-Forti, 17
Burali-Forti’s
Paradox, 17
Busy Beaver, 174
Busy Beaver function, 172, 174
c.e. Degree, 269–280
Post’s Problem, \textit{see} Post’s Problem
Canonical system, 83, 177, 231
Cantor, 13
Cantor’s
definition of the set, 20
diagonalization, 191
Paradox, 17
Theorem, 16
view of infinity, 21
Cardinal number, 15
\(\aleph_0\), the cardinality of \(\mathbb{N}\), 15
\(\aleph_1\), the cardinality of \(2^{\aleph_0}\), 16
finite, 16
transfinite, 16, 59
Cellular automaton, 89
Characteristic function, 131, 144, 147, 172, 204
Characterization of the algorithm, 71
Church’s, 78
convincing, 88
Gödel-Kleene’s, 74
Herbrand-Gödel’s, 77
Markov’s, 85
Post’s, 84
Turing’s, 82
Chomsky, 179
Church, 73, 77–79, 87, 129, 161, 166, 202
Church Thesis, 87
Church-Rosser Theorem, 79
Church-Turing Thesis, \textit{see} Computability Thesis
Class, 43, 45, 286
arithmetical, 286
non-proper, 45
proper, 45
Class \(\mathcal{D}\) of degrees of unsolvability, 255–267
basic properties, 257–267
cardinality, 257–258
greatest lower bound, 262–263
incomparable Turing degree, 259–261
intermediate Turing degree, 263–264
least Turing degree, 262
least upper bound, 262
minimal Turing degree, 266–267
structure of, 258–263
upper and lower cone, 264–265
initial clues of, 169–171
Class of all decision problems
basic structure, 169–171
Coding function, 163
Cohen, 58
Compactness Theorem, 47
Completeness, 49, 203
Completeness Problem
of Principia Mathematica, 29
Completeness Requirement, 73, 77, 78, 82, 87, 97
rules of, 33
Diagonal language $K$, 167
Diagonalization, 16, 91, 129, 161, 191–195, 199
direct, 191–193
switched diagonal, 192, 193
switching function, 192, 193
indirect, 194–195
shrewd Turing machine (algorithm), 194
Diophantine equations, 180
DNA-calculus, 89
Domain, 26
of function, 93, 125, 158
of interpretation, 26, 35
dovetailing, 134

Eckert, 120
EDVAC, see General-purpose computing machine
Effective procedure, 73
Effectiveness Requirement, 73, 77, 78, 82, 87, 97
Elgot, 122
ENIAC, see General-purpose computing machine
Entscheidungsproblem, 53, 55, 56, 69, 97, 202–204
Enumeration function, 116, 129
Equation
Diophantine, 180
system, 76
Euclid, 5, 11
Euclid’s
algorithm, 5
axiomatic method, 9
fifth axiom, 11
gometry, 9
Execution
of algorithm, 70
of instruction, see Instruction
Existence
and construction, 21, 172
intuitionistic view, 21, 172
of a computable non-recursive function, 91, 191, 193
of a general-purpose computing machine, 119
of a Universal Turing machine, 117–118
of equivalent Turing programs, 151, 153
of fixed-points of computable functions, 152, 154
of incomputable functions, 172, 174
of Turing machines, 114
of undecidable problems, 166
of undecidable semi-decidable problems, 169
of undecidable semi-decidable sets, 169, 170
of undecidable sets, 168
Platonic view, 21
Expression, 27, 28, 32, 82
Finite Extension Method, 275
Finitism, 28, 54, 61, 70, 90
First Incompleteness Theorem, 57–61, 69
First-Order Logic $L$, 24, 39, 50, 53, 54, 60, 298
Fixed-Point Theorem, see Recursion Theorem
Formal Arithmetic $A$, 38, 40, 54, 57, 60, 203, 204
Formal axiomatic system, 27, 31, 33, 35, 204
$M$, of the whole of mathematics, 54
consistent axiomatic extension of, 57, 60
extension of $L$, 40
of the first order, 40, 46
of the second order, 46
Formalism, 20, 26, 28, 31
its goals, 29
Formalization
of algorithm’s environment, 71
of arithmetic, 40
of computation, 71
of logic, 39
of set theory, 42
of the whole of mathematics, 54
Formula, 27, 32, 59, 203
closed (= sentence), 32, 37
independent of a theory, 50, 57, 58, 60
its truth-value, 37, 52
logically valid, 38
of the second order, 47, 48
open, 32, 37
satisfiable under the interpretation, 38
undecidable, 69, 90
valid in a theory, 38, 52, 58
valid under the interpretation, 38
Foundations of mathematics, 49
problems of, 49
Fraenkel, 42
Frege, 23, 39
Frege’s
Begriffsschrift, 24
Friedberg, 274–278
Function, 14
$\lambda$-definable, 78, 87
Ackermann, 73, 99
bijective, 14, 137
Busy Beaver, 172, 174
characteristic, see Characteristic function
coding (for a decision problem), 163
computable, see Computable function defined, 94
domain of, 93, 158
elementary, 182
enumeration, see Enumeration function equal, 93
extension of, 93
fixed point of, 152, 154, 204, 207, 212
general recursive, 76
graph of, 292
homeomorphism, 183
identity, 202, 203
incomputable, see Incomputable function injective, 14, 93, 163, 201
Markov-computable, 85
numerical, 72, 90, 91, 193
oracle-computable, see Oracle-computable function p.c., see Partial computable (p.c.) function pairing, 140
partial, 74, 92–97, 103
Post-computable, 84
proper (of a Turing machine), 125, 146, 246
property of, see Property range of, 93, 129, 130
recursive, see Recursive function surjective, 14, 93
switched, see Diagonalization total, 72, 90, 97, 205, 207
transition, see Turing machine Turing-computable, 81, 87
zeros of, 182
Function computation, 125–126
Functional, 228, 247
proper (of an oracle Turing machine), 228, 247, 249
Generator, 129, 135, 138, 203
Gentzen, 61
Geometry
Euclidean, 11
fifth axiom (=Parallel Postulate), 11
non-Euclidean elliptic, 12
hyperbolic, 12
model of, 12
Goldbach’s Conjecture, 22, 63, 90
Grammar
in Markov algorithms, 84
in Post’s canonical systems, 127
of a formal language, 177
ambiguous, 177
context-free, 177
context-sensitive, 177
equivalent, 178
regular, 177
problems about, 177
Graph Theorem, 292
Gödel number, 59, 64, 71
Gödel’s Completeness Theorem, 52
First Incompleteness Theorem, 46, 57–61, 69
formula, 60
Second Incompleteness Theorem, 46, 60–62
Hartmanis, 122
Heisenberg, 88
Herbrand, 54, 73, 76, 129
Heyting, 21
Hierarchy
arithmetical, 283–292
jump, 245–252, 288–291
Hilbert, 12, 26, 49, 202
Hilbert’s elementary geometry, 12
Program, 49, 53, 58, 204
finitism, 54, 61, 70
intention, 53
its fate, 54, 204
legacy, 63
IAS, see General-purpose computing machine
Incomputability proving, 191–215
by diagonalization, see Diagonalization
by Recursion Theorem, see Recursion Theorem
by reduction, see Reduction
General recursive function, 76, 87
General-purpose computing machine, 119–124, 138
architecture, see RAM
compiler, see Procedure
existence of, 119
operating system, see Operating system
Generalization, 33, 39, 40, 52
Generating (=enumeration) problem, 162
Generation
formalization by
Church and Kleene, 129
Post, 127
Turing machine, 129
of pairs of natural numbers, 135
of sequences of symbols, 55, 203
Index

Function computation, 125–126
Functional, 228, 247
proper (of an oracle Turing machine), 228, 247, 249
Generator, 129, 135, 138, 203
Gentzen, 61
Geometry
Euclidean, 11
fifth axiom (=Parallel Postulate), 11
non-Euclidean elliptic, 12
hyperbolic, 12
model of, 12
Goldbach’s Conjecture, 22, 63, 90
Grammar
in Markov algorithms, 84
in Post’s canonical systems, 127
of a formal language, 177
ambiguous, 177
context-free, 177
context-sensitive, 177
equivalent, 178
regular, 177
problems about, 177
Graph Theorem, 292
Gödel number, 59, 64, 71
Gödel’s Completeness Theorem, 52
First Incompleteness Theorem, 46, 57–61, 69
formula, 60
Second Incompleteness Theorem, 46, 60–62
Hartmanis, 122
Heisenberg, 88
Herbrand, 54, 73, 76, 129
Heyting, 21
Hierarchy
arithmetical, 283–292
jump, 245–252, 288–291
Hilbert, 12, 26, 49, 202
Hilbert’s elementary geometry, 12
Program, 49, 53, 58, 204
finitism, 54, 61, 70
intention, 53
its fate, 54, 204
legacy, 63
IAS, see General-purpose computing machine
Incomputability proving, 191–215
by diagonalization, see Diagonalization
by Recursion Theorem, see Recursion Theorem
by reduction, see Reduction
General recursive function, 76, 87
General-purpose computing machine, 119–124, 138
architecture, see RAM
compiler, see Procedure
existence of, 119
operating system, see Operating system
Generalization, 33, 39, 40, 52
Generating (=enumeration) problem, 162
Generation
formalization by
Church and Kleene, 129
Post, 127
Turing machine, 129
of pairs of natural numbers, 135
of sequences of symbols, 55, 203

by Rice’s Theorem, see Rice’s Theorem
Incomputable function, 95, 126, 138, 172, 174, 204
  on a set, 96, 126
Incomputable problem, 161
  Ambiguity of CFG grammars, 177
  Busy Beaver function, 174
  Busy Beaver Problem, 174
  Classification of manifolds, 183
  Correctness of algorithms (programs), 176
  Decidability of first-order theories, 184
  Domino snake problem, 186
  Domino tiling problem, 186
  Equality of words, 182
  Equivalence of CFG grammars, 178
  Existence of shorter equiv. programs, 176
  Existence of zeros of functions, 182
  Halting of Turing machines, 173
  Mortal matrix problem, 180
  Other properties of CFGs and CFLs, 178
  Post’s Correspondence Problem, 175
  Properties of computable functions, 179
  Properties of TM languages, 173
  Satisfiability of first-order formulas, 185
  Shortest equivalent program, 210
  Solvability of Diophantine equations, 180
  Termination of algorithms (programs), 175
  Validity of first-order formulas, 185
  Word problem for groups, 181
  Word problem for semi-groups, 181
Incomputable set, see Undecidable set
Independent formula, 50, 57, 58, 60
Index
  of a partial computable (p.c.) function, 145–147, 151, 153
  of a proper function (of a TM), 125
  of a Turing machine, 115, 145, 149
  of a Turing program, 145, 149
Index set, 146–147, 205–207, 211
  of a partial computable (p.c.) function, 146
  of a partial oracle-computable function, 230
  of a semi-decidable (=c.e.) set, 147
  of an oracle-semi-decidable set, 230
  property of, 206–207
Induction
  mathematical, 42, 47
  transfinite, 54, 61, 62
Inference, 10, 27
  rules of, 23, 28, 33, 35, 52, 56
Infinity
  Aristotle’s view of, 21
  Cantor’s view of, 21
  in intuitionism, 21
  Inspiration, ingenuity, 63, 205
Instruction
  basic, 70, 86
  execution
    continuous, 70
    discrete, 70
    predictable, 70
    random, 70
    intuitively, 4
  Internal configuration, 86, 203
Interpretation
  domain of, 26, 35, 36
  of a theory, 12, 26, 28, 35, 36
  standard, 38, 40, 42
Intuition and experience, 11, 12
Intuitionism, 20
  and existence of objects, 21
  and the Law of Excluded Middle, 21
Jump hierarchy, 245–252, 288–291
Kleene, 74, 87, 129, 149, 231
Kučera, 280
Kuratowski, 14
Lachlan, 279
Language
  diagonal \( K \), 167
  expression of, 47
  formal, 28, 137
  natural, 24, 28, 82
  of a Turing machine, see Proper set of TM problems about, 177
  programming, 177
  symbolic, 24, 27, 32, 35
    of the first order, 32, 46
    of the second order, 32, 46
  universal \( K_0 \), 167, 169
Law of Excluded Middle, 13, 18, 21, 54, 163
Leibniz, 6, 59
  arithmetization of man’s reflection, 6
  calculus ratiocinator, 6
  lingua characteristica universalis, 6
Lexicographical order, 91
Lobachevsky, 12
Logical validity (of formulas), 38
Logicism, 20, 23
Löwenheim-Skolem Theorem, 47
Machine, 101
Manifold
  classification of, 183
  homeomorphic, 183
  topological, 183
Markov, 82, 84–85
Markov algorithm, 84–85
Markov-computable function, 85
Mathematical induction, 42
Mathematical structure, 35
Matiyasevič, 180
Matrix mechanics, 88
Mauchly, 120
Mechanical manipulation
of symbols, 24, 34
benefits, 34
Melzak, 122
Metamathematics, 11, 28, 38
Metatheory, see Metamathematics
Method, 191
Minsky, 122
Model, 69
Model of a theory, 35, 38, 52, 58
standard, 40, 42, 57, 60
Model of computation, 72–86, 89, 97, 101, 205
after functions, 72
after humans, 80
after languages, 82
cellular automaton, 89
Church’s λ-calculus, 78
convincing, 88
data-flow graph, 83
DNA-calculus, 89
equivalent, 87, 88, 97, 122, 124, 149, 153, 195
Gödel-Kleene’s recursive functions, 74, 90
Herbrand-Gödel’s general recursive
functions, 77
instance of, see Abstract computing machine
Markov algorithm, 85
Post machine, 84
RAM, 89, 122
RASP, 89, 122
reasonable, 86, 97
register machine, 89, 122
Turing machine, 82, see Turing machine
unrestricted, 86
Model theory, 46
Modus Ponens, 33, 39, 40, 52
Muchnik, 274–278
Natural numbers \( \mathbb{N} \), 54, 57, 71, 116, 137, 193,
204, 207
\( \mathbb{N}_0 \), the cardinality of \( \mathbb{N} \), 15, 193
\( \mathbb{N}_1 \), the cardinality of \( 2^{\mathbb{N}} \), 16, 191
von Neumann’s construction, 15
Nature
at macroscopic level, 70
at microscopic level, 70, 88
continuous, 70
discrete, 70
predictable, 70
random, 70
objectively, 70
subjectively, 70
Non-finitism, 61
Nondiamond theorem, 279
Notation
standard formal, 24
Number system
decimal positional, 5
Roman, 5
Operating system, 121, 138, 156
activation record, 156
bootstrap loader, 121
data region, 121
dynamic allocation of space, 121
executable program, 158
file system, 121
global variables, 121
heap, 121
input-output management, 121
loading, 121
local variables, 121, 156
memory management, 121
multiprogramming, 121
networking, 121
procedure-linkage, 121
program initiation, 121
protection, 121
residency, 121
runtime stack, 121, 156
security, 121
system call, 158
task, 156
Optimization problem, see Search problem
Oracle, 219–222
Oracle Turing machine, 220–224
\( o \)-TM, 223–231
coding, 226–227, 246–247, 249
enumeration of, 226–227
index of, 227
operational semantics, 223
oracle tape, 223–224
program of, 224–226
with a database, 232
with a network of computers, 232
with oracle set, 224
\( o \)-machine, 220–222
nondeterminism, 221
oracle set, 221
special instructions, 220–221
special states, 220–221
Oracle-computable function, 229
computable, 229
on a set, 229
incomputable, 229
partial, 229
on a set, 229
Oracle-decidable set, 230
in a set, 230
Oracle-semi-decidable set, 230
in a set, 230
Oracle-undecidable set, 230
Order
lexicographical, 91
shortlex, 91
Ordered pair, 14
Ordinal number, 17, 18, 63
finite, 17
limit, 63
transfinite, 17
Ordinal type, 18
OS, see Operating system
Outwitting an incomputable problem, 187

Padding Lemma, 143, 145–146, 158
generalized, 230
Paradox, 8, 17, 21, 27, 38, 63
Burali-Forti’s, 17, 18
Cantor’s, 17
fear of, 18
Russell’s, 18, 24, 45
safety from, 63
Parallel Postulate, see Geometry
Parameter, 147, 153
Parametrization Theorem, 143, 147–149, 153, 154, 158, 247
Partial computable (p.c.) function, 95, 126, 138, 146, 147, 205–208
on a set, 96, 126
property of, 205–208
Partial recursive function, 74, 153
also see Recursive function, 93
relative to a set, 231
Pascal, 6
Peano, 12, 23, 24, 39
Peano Arithmetic PA, 40, see Formal Arithmetic A
Peano’s
axioms of arithmetic, 12, 40
Formulario Mathematico, 24
symbolic language, 24
Peirce, 23
Penrose, 118
Platonic view, 10, 20
Post, 51, 82–84, 87, 88, 127, 167, 220, 231
Post machine, 83–84, 128
computation on, 84
halting, 84
starting, 84
step, 84
control unit, 83
input word, 83
Post program, 83
as a graph, 83
queue, 83
tape, 83
window, 83
Post Thesis, 128, 130
Post’s Problem, 270
Friedberg-Muchnik’s solution, 278
Post’s Program, 271
priority-free solution
Kučera, 280
Post’s Theorem, 144
Post-computable function, 84
Predicate Calculus, see First-Order Logic L
Premise, 10, 33, 56
Presburger, 60, 204
Presburger Arithmetic, 60, 204
Principia Mathematica, 25, 29, 63, 128
Completeness Problem of, 25, 29
Consistency Problem of, 25, 29
decidability of, 127
imperfections in, 25
importance of, 25
Priority Method, 269, 274–280
candidate, 275
Finite-Injury Priority Method, 277
Infinite-Injury Priority Method, 277
priority, 276
priority argument, 279
requirement, 275
fulfilling, 275
injury, 277
receiving attention, 275
Problem kinds, 162, 165, 210
computational, see Computational problem
computable, see Computable problem
incomputable, see Incomputable problem
counting, see Counting problem
decision
decidable, see Decidable problem
semi-decidable, see Semi-decidable problem
undecidable, see Undecidable problem
decision (=yes/no), see Decision problem
general
solvable, 165
unsolvable, 165
Index

generating (=enumeration), 162
non-computational, 161
search, see Search problem
Procedure, 156–158
activation, 157
activation record, 157
actual parameter, 157
addressability information, 157
local variables, 157
register contents, 157
return address, 157
return-values, 157
call, 157
callee, 157
caller, 157
recursive, 157
runtime algorithms and data structures, 157
stack, 157
Processor
capability
limited, 70
unlimited, 70
intuitively, 4, 70
Production, 82, 84
Program, 101
problems about, 175
recursive, 153, 154
with procedures, see Procedure
Proof, 10
constructive, 172
formal (=derivation), 27, 34, 51, 55
informal, 10, 28
of consistency of a theory, 60
absolute, 60
relative, 60
Property
decidable and undecidable, 205–210
intrinsic, 205–210
of abstract computing machines, 209
of algorithms, 194–195
of c.e. sets, 205, 208–209
of index sets, 205–207
of p.c. functions, 205–208
of sets, 137
of Turing programs (machines), 209
trivial and non-trivial, 205–210
Proposition, 10, 127
Propositional Calculus \( \mathbf{P} \), 23, 50, 51, 60, 127, 297
Putnam, 180
Quantification symbol
existential \( \exists \), 32, 46
universal \( \forall \), 32, 46
Quantum
algorithm, 70
phenomena, 70
theory, 88
Radó, 174
RAM, 89, 122–124, 138
accumulator, 122
equivalence with Turing machine, 122, 124
halting, 123
input memory, 122
instructions, 123
main memory, 122
output memory, 122
processor, 122
program, 122
program counter, 122
random access to data, 122
registers, 122
starting, 123
von Neumann’s architecture, 122
Randomness
objective, 70
subjective, 70
Range of a function, 93, 129, 130
RASP, 89
Real numbers \( \mathbb{R} \)
c, the cardinality of \( \mathbb{R} \), 16, 58, 172, 193
completeness of \( \mathbb{R} \), 48
Rechow, 122
Recipe, 3, 4, 70, 71, 77
Recognition
of arithmetical theorems, 204
of mathematical truths, 69
of sets, 130–133
Recognizer, 131, 132, 138, 144, 147, 165, 203, 248
Recursion (=self-reference), 142, 143, 149, 161
in a general-purpose computer, 156–158
in a partial recursive function, 73–75
in definition of a Turing program, 153–154, 158
in execution of a Turing program, 154–156, 158
primitive, 73–75
Recursion Theorem, 143, 149–158, 204, 212
Recursive function, 73–75, 90, 91, 149, 193
Ackermann, 73, 99
initial, 73–75
projection \( \pi \), 73–75
successor \( \sigma \), 73–75
zero \( \zeta \), 73–75
partial, 74
primitive, 73
rule of construction
\(\mu\)-operation, 74–75, 90
composition, 73–75
primitive recursion, 73–75, 153
Recursive set, see Computable set
Recursively enumerable (r.e.) set, see Computationally enumerable (c.e.) set
Reduction, 196–204, 271–273
of a computational problem, 196–197
additional conditions, 197
basic conditions, 196
informally, 197
of a decision problem
to checking the (non)triviality, 207
to the set-membership problem, 164
of a decision problem or a set
1-reduction \(\leq_1\), 201–204
\(T\)-reduction \(\leq_T\), 235–237
\(m\)-reduction \(\leq_m\), 197–201
\(btt\)-reduction \(\leq_{btt}\), 273
\(tt\)-reduction \(\leq_{tt}\), 273
resource-bounded, 197
strong, 197, 273
Register machine, 89
Rice, 205
Rice’s Theorem, 204–210
Riemann, 12
Robinson, 122, 180
Rogozhin, 118
Rosser, 60, 78–79, 87
Rule
of construction, 24, 28, 32, 35
of inference, 23, 28, 33, 35, 52, 56
Generalization, 33
Modus Ponens, 33
Russell, 18, 24, 39, 42
Russell’s
Paradox, 18, 24, 45
Theory of Types, 24
s-m-n Theorem, 147–149, 158
Sacks, 266, 279
Satisfiability (of formulas), 37, 184
Scenario, see Turing program
Schickard, 6
Schrödinger, 88
Search problem, 162, 204, 210
Second Incompleteness Theorem, 60–62
Self-reference, see Self-reference or Recursion
Semantic Completeness, 63
Semantic Completeness Problem, 52
Semantics, 26, 28
intended, 35
Semi-decidable problem, 165, 199, 203
Semi-decidable set, 132, 138, 143–144, 147, 158, 165, 199, 205, 208–209
existence of undecidable one, 169
is computably enumerable (c.e.) set, 136
property of, 205, 208–209
Semi-Thue system, 181
Sentence, 32
Set
\(m\)-complete, 213
1-complete, 213
arithmetical, 284
Burali-Forti’s paradoxical \(\Omega\), 18, 20, 24
c.e., see Computationally enumerable (c.e.) set
Cantor’s definition of, 13, 20
Cantor’s paradoxical \(\mathcal{U}\), 18, 20, 24, 42
complement, 169
computable, see Decidable set
creative, 273
decidable, see Decidable set
equinumerous, 15
homeomorphic, 183
hyper-simple, 273
hyperhyper-simple, 273
incomputable, see Undecidable set
inductive, 45
infinite
as actuality, 21
as potentiality, 21
intersection, 144, 158
linearly ordered, 18
locally Euclidean, 183
normal, 127–129
property of, 137
recognized with oracle, see Oracle-decidable set
Russell’s paradoxical \(\mathcal{R}\), 18, 20, 24, 42
semi-decidable, see Semi-decidable set
similar, 17, 18
simple, 272
undecidable, see Undecidable set
union, 144, 158
well-ordered, 17, 62
Set generation, 127–130, 133–135
Set recognition, 130–135, 164
Set theory
axiomatic
\(\text{NBG}\), 43, 45, 54
\(\text{ZFC}\), 42, 44, 54
Cantor’s, 13, 42
as a foundation of all mathematics, 15
Shannon, 118
Shepherdson, 122
Shortlex order, 91
Index

Skolem, 42
Soare, 95
Space, see Abstract computing machine
Spector, Clifford, 266
Statement, 10, 23, 37, 203
metamathematical, 59
Stopper, 174
Storage
capacity
limited, 70
unlimited, 70
intuitively, 70
Sturgis, 122
Symbol, 24, 27
canonical, 127
equality, 32
function, 32
function-variable, 32, 46
individual-variable, 32
logical connective, 32
mechanical manipulation, 24, 34
proper, 40
punctuation mark, 32
quantifier, 32
relation, 32
relation-variable, 32, 46
Syntactic Completeness Problem, 50
Syntactic object, 59
Syntactic relation, 59
Syntax, 26
System
canonical, 127
normal, 127
of equations, 76, 129
Terminology (old vs. new), 142
Theorem
of a theory, 10, 27, 34, 51
Theory
$\omega$-consistent, 60
axioms of, 10
basic notions of, 10
cognitive value of, 18, 50, 52
complete, 39
conservative extension, 44
consistent, 38, 50, 56, 57, 60
decidable, 51, 56, 127, 204
development of, 10
in a formal axiomatic system, 27, 33, 35
especially incomplete, 59
initial, 10, 33
model, 46
of the first order, 40
of the second order, 46
paraconsistent, 62
recursively axiomatizable, 56
semantically complete, 52
semantically incomplete, 57, 63
sound, 52
syntactically complete, 50, 56
Thue, 181
system, 181
Tiling
a path, 185
a polygon, 185
Time, see Abstract computing machine
TM, see Turing machine
Topological invariant, 183
Torsion group, 48
Total extension, see Function
TP, see Turing program
Transfinite Induction, 54, 61, 62
Truth-table, 127
Truth-value, 21
dubious, 21
of a formula, 37, 52
Turing, 80–82, 87, 107, 115, 117, 161, 166,
202, 220
Turing degree, 240–243, 245–252
c.e., see c.e. degree
greatest lower bound, 262–263
incomparable, 259–261
intermediate, 263–264
least, 262
least upper bound, 262
minimal, 266–267
upper and lower cone of, 264–265
Turing jump
of a set, 246–252
of a Turing degree, 255–267
Turing machine, x, 97, 101–138, 161
accepting a word, 130
as a computer of a function, 126, 138
as a decider of a set, see Decider
as a generator of a set, see Generator
as a recognizer of a set, see Recognizer
as an acceptor of a set, 131
basic model, 80–82, 101–106, 138
cell, 81, 102
computation on, 81, 103, 104
cell, 103, 104, 166
control unit, 81, 103
empty, 116
final state, 103
generalization, see Turing machine
variants
halting, 103, 104, 166
initial configuration, 104
initial state, 103
ingput alphabet $\Sigma$, 81, 103, 119
input word, 81, 103
instruction, 103, 104
internal configuration, 104
memorizing, 82
non-final state, 103
starting, 103
state, 81, 103
tape, 81, 102
tape alphabet $\Gamma$, 102
tape symbol, 102
transition function, 103
Turing program, 81, 103
window, 81, 103
Busy Beaver, 174
coding of TMs, 115, 147, 168, 246
enumeration of TMs, 115
equivalence with RAM, 124
generalized models, 101, 107
external configuration, 107
Finite storage TM, 107
Multi-tape TM, 108
Multi-track TM, 107
Multidimensional TM, 108
Nondeterministic TM, 108
Two-way unbounded TM, 107
use of, 114
index of, 115, 145
not recognizing a word, 130
proper function of, 125, 146
proper set of, 130, 169, 202
reduced model, 113
redundant instruction, 145
rejecting a word, 130
shrewd, 168, 194
simulation with the basic model, 109
of Finite storage TM, 109
of Multi-tape TM, 110
of Multi-track TM, 110
of Multidimensional TM, 111
of Nondeterministic TM, 112
of Two-way unbounded TM, 110
simulation with the reduced model
of basic model, 114
stopper, 174
universal, 115–124, 138, 169
existence of, 117–118
small, 118
use of a TM, 125–133
for function computation, 125
for set generation, 127
for set recognition, 130
variants, 115, 138
Turing program, 147
behavior
global, 146, 147, 152, 209
local, 146, 147, 152, 209
coding of TPs, 115, 147
enumeration of TPs, 115
equivalent, 151–153, 211
index of, 115, 145
length, 210, 211
nondeterministic, 108
decision tree, 113
indeterminacy of, 112
instruction indeterminacy, 112
scenario of execution, 112
property of, 209
recursive definition, 153, 158
recursive execution, 154, 158
activation, 154
activation record, 155
actual parameter, 154
callee, 155
caller, 155
stack, 156
shortest equivalent, 210
transformation of, 151, 153
Turing reduction, 235–237
properties of, 237–239
Turing Thesis, 88
Turing-computable function, 81, 87
Undecidability proving, see Incomputability proving
Undecidable formula, 69, 90
Undecidable problem, 165, 197–210, 245–252
examples, see Incomputable problem
existence of, 166
Halting Problem, 166
Undecidable set, 132, 138, 165, 199, 206, 207
existence of, 168
Universal language $K_0$, 167, 169
Universe, 131
standard $\Sigma^*$ and $\mathbb{N}$, 136, 170
Validity (of formulas), 37, 184
Variable
bound, 32, 77–79
free, 32, 37, 77–79
quantified, 24
do Neumann, 15, 43, 120
do Neumann’s architecture, 120
accumulator, 120
main memory, 120
memory address, 120
memory location, 120
Index

processor, 120
program, 120
program counter, 120
random access to data, 120
registers, 120

Wang, 122
Wave mechanics, 88

Whitehead, 24, 25, 39
Word, 27, 32
World of infinities, 17

Yates, 273, 279
Yes/No problem, see Decision problem

Zermelo, 42