Lectures on mathematical foundations of computing

Takunari Miyazaki

October 3, 2025

Contents

§1. Propositional logic

Our first topic is *propositional logic*: calculus of mathematical propositions.

Formally, a *proposition* is a statement that is either true or false.

Example 1.1. The following are all examples of such propositions.

(a) *Boston is the capital of Massachusetts.*

(b) *New Haven is the capital of Connecticut.*

(c) $1 + 1 = 2$.

(d) $2n + 1$ *is odd for any integer* $n > 0$.

On the other hand, none of the following is a proposition, as none of them is clearly neither true nor false.

(e) *Why am I here?*

(f) *Go Bantams!*

(g) $x = y$.

(h) *There are* 28 *days in February.*

In propositional calculus, it is customary to use lowercase letters, such as p, q and r, to denote arbitrary propositions, with the following notation for the five basic

Logical operators.

$\neg$ for "not"

$\wedge$ for "and"

$\vee$ for (inclusive) "or"

$\rightarrow$ for "implies"

$\leftrightarrow$ for "if and only if"

Example 1.2. If

$$
\begin{aligned}
p &= \text{"It is raining,"} \\
q &= \text{"The sun is shining," and} \\
r &= \text{"There are clouds in the sky,"}
\end{aligned}
$$

then

$$
\begin{aligned}
\neg p &= \text{"It is not raining,"} \\
\neg p \wedge q &= \text{"It is not raining, and the sun is shining,"} \\
p \vee q &= \text{"It is raining, or the sun is shining,"} \\
p \rightarrow r &= \text{"If it is raining, then there are clouds in the sky," and} \\
p \leftrightarrow r &= \text{"It is raining if and only if there are clouds in the sky."}
\end{aligned}
$$

In English, $p \rightarrow q$ is usually read "p implies q," or "if p, then q." Other equivalent expressions are "q if p," "q whenever p," and "p only if q."

Consider propositions p and q. Here, $q \rightarrow p$ is the *converse* of $p \rightarrow q$, and $\neg q \rightarrow \neg p$ is the *contrapositive* of $p \rightarrow q$ and "equivalent" to $p \rightarrow q$ (in a moment, we will define this notion of equivalence). For $p \rightarrow q$, we say p is a *sufficient condition* for p, and q is a *necessary condition* for p.

Example 1.3. Consider $p \rightarrow q$, where

$$
\begin{aligned}
p &= \text{"It is raining," and} \\
q &= \text{"Road is wet."}
\end{aligned}
$$

Here, the rainy weather is a sufficient condition for wet road. It is also necessary for road to be wet for the rainy weather.

p	$\neg p$
T	F
F	T

p	q	$p \wedge q$
T	T	T
T	F	F
F	T	F
F	F	F

p	q	$p \vee q$
T	T	T
T	F	T
F	T	T
F	F	F

p	q	$p \rightarrow q$
T	T	T
T	F	F
F	T	T
F	F	T

p	q	$p \leftrightarrow q$
T	T	T
T	F	F
F	T	F
F	F	T

Table 1.1: Truth values of $\neg p$, $p \wedge q$, $p \vee q$, $p \rightarrow q$ and $p \leftrightarrow q$

Given propositional variables p and q, what are the truth values of $\neg p$, $p \wedge q$, $p \vee q$, $p \rightarrow q$ and $p \leftrightarrow q$? The truth values are defined by truth tables for all combinations of truth values of the variables p and q (see Table 1.1).

Now, consider $(p \rightarrow q) \wedge (q \rightarrow p)$ and its truth table:

p	q	$(p \rightarrow q)$	$\wedge$	$(q \rightarrow p)$
T	T	T	T	T
T	F	F	F	T
F	T	T	F	F
F	F	T	T	T

We notice in particular that $(p \rightarrow q) \wedge (q \rightarrow p)$ has the same truth values as those of $p \leftrightarrow q$, and this shows that these propositions are "equivalent" in the following sense.

We formally say two propositions are *logically equivalent* if they have the same truth values under all combinations of truth values of their propositional variables. To indicate logical equivalence, we use $\equiv$.

On the next page, in Tables 1.2 and 1.3, we summarize some of the most important logical equivalences (cf. [2, §1.3, Tables 6, 7 and 8]).

$p \wedge \mathrm{T} \equiv p$ $p \vee \mathrm{F} \equiv p$	Identity laws
$p \vee \mathrm{T} \equiv \mathrm{T}$ $p \wedge \mathrm{F} \equiv \mathrm{F}$	Domination laws
$p \vee p \equiv p$ $p \wedge p \equiv p$	Idempotent laws
$\neg(\neg p) \equiv p$	Double negation law
$p \vee q \equiv q \vee p$ $p \wedge q \equiv q \wedge p$	Commutative laws
$(p \vee q) \vee r \equiv p \vee (q \vee r)$ $(p \wedge q) \wedge r \equiv p \wedge (q \wedge r)$	Associative laws
$p \vee (q \wedge r) \equiv (p \vee q) \wedge (p \vee r)$ $p \wedge (q \vee r) \equiv (p \wedge q) \vee (p \wedge r)$	Distributive laws
$\neg(p \wedge q) \equiv \neg p \vee \neg q$ $\neg(p \vee q) \equiv \neg p \wedge \neg q$	De Morgan's laws
$p \vee (p \wedge q) \equiv p$ $p \wedge (p \vee q) \equiv p$	Absorption laws
$p \vee \neg p \equiv \mathrm{T}$ $p \wedge \neg p \equiv \mathrm{F}$	Negation laws

Table 1.2: Logical equivalences

$p \rightarrow q \equiv \neg p \vee q$
$p \rightarrow q \equiv \neg q \rightarrow \neg p$
$p \vee q \equiv \neg p \rightarrow q$
$p \wedge q \equiv \neg(p \rightarrow \neg q)$
$\neg(p \rightarrow q) \equiv p \wedge \neg q$
$(p \rightarrow q) \wedge (p \rightarrow r) \equiv p \rightarrow (q \wedge r)$
$(p \rightarrow r) \wedge (q \rightarrow r) \equiv (p \vee q) \rightarrow r$
$(p \rightarrow q) \vee (p \rightarrow r) \equiv p \rightarrow (q \vee r)$
$(p \rightarrow r) \vee (q \rightarrow r) \equiv (p \wedge q) \rightarrow r$

$p \leftrightarrow q \equiv (p \rightarrow q) \wedge (q \rightarrow p)$
$p \leftrightarrow q \equiv \neg p \leftrightarrow \neg q$
$p \leftrightarrow q \equiv (p \wedge q) \vee (\neg p \wedge \neg q)$
$\neg(p \leftrightarrow q) \equiv p \leftrightarrow \neg q$

Table 1.3: Logical equivalences of conditional and biconditional statements

Logical equivalence is *transitive* in the sense that, for propositions p, q and r, if $p \equiv q$ and $q \equiv r$, then $p \equiv r$.

Example 1.4. For propositions p and q, using the equivalence $p \to q \equiv \neg p \vee q$ from Table 1.3 and De Morgan's and double negation laws from Table 1.2, the following proves the equivalence $\neg(p \to q) \equiv p \wedge \neg q$:

$$\begin{array}{rcll} \neg(p \to q) & \equiv & \neg(\neg p \vee q) & \text{since } p \to q \equiv \neg p \vee q \\ & \equiv & \neg(\neg p) \wedge \neg q & \text{by De Morgan's law} \\ & \equiv & p \wedge \neg q & \text{by a double negation law} \end{array}$$

Example 1.5. For propositions p, q and r, the following proves the equivalence $(p \to r) \wedge (q \to r) \equiv (p \vee q) \to r$:

$$\begin{array}{rcll} (p \to r) \wedge (q \to r) & \equiv & (\neg p \vee r) \wedge (\neg q \vee r) & \text{since } p \to q \equiv \neg p \vee q \\ & \equiv & (r \vee \neg p) \wedge (r \vee \neg q) & \text{by a commutative law} \\ & \equiv & r \vee (\neg p \wedge \neg q) & \text{by a distributive law} \\ & \equiv & (\neg p \vee \neg q) \vee r & \text{by a commutative law} \\ & \equiv & \neg(p \vee q) \vee r & \text{by De Morgan's law} \\ & \equiv & (p \vee q) \to r & \text{since } p \to q \equiv \neg p \vee q \end{array}$$

Next, for propositions p and q, consider $(p \wedge (p \to q)) \to q$ and its truth table:

p	q	$(p$	$\wedge$	$(p \to q))$	$\to$	q
T	T	T	T	T	T	T
T	F	T	F	F	T	F
F	T	F	F	T	T	T
F	F	F	F	T	T	F

We notice that this proposition is always true regardless of the truth values of p and q. Using known logical equivalences, it is also easy to derive that $(p \wedge (p \to q)) \to q \equiv \mathrm{T}$.

In general, a proposition that is always true is called a *tautology*. On the other hand, a proposition that is always false is called a *contradiction*. Here is the most important

Example 1.6. For a proposition p, the following negation laws hold:

$$\begin{array}{rcl} p \vee \neg p & \equiv & \mathrm{T} \\ p \wedge \neg p & \equiv & \mathrm{F} \end{array}$$

Now, consider this statement:

> $2n+1$ *is an odd integer.*

With n unspecified, this is not a proposition; however, when n is specified, it becomes a proposition:

> $2n+1$ *is an odd integer for all integers n.*

This is an example of quantification of a *predicate*: a statement with variables that becomes a proposition with values of the variables specified.

There are two ways to quantify variables in predicates, using the universal and existential quantifiers. In the following, let $p(x)$ denote a predicate with a variable x in some domain.

With the *universal quantifier* $\forall x$, we write $\forall x\, p(x)$ to denote "for all x (in the domain), $p(x)$." Its truth values are defined by:

$$\begin{aligned}\forall x\, p(x) &\equiv \text{T} \quad \text{if } p(x) \text{ is true for every } x \text{ in the domain}\\ &\equiv \text{F} \quad \text{otherwise}\end{aligned}$$

That is, $\forall x\, p(x) \equiv \text{F}$ if there exists x_0 in the domain that makes $p(x_0)$ false. We call such an x_0 a *counterexample*. To prove $\forall x\, p(x)$, one must prove that $p(x)$ is true for all x in the domain. To prove it is false, it suffices to provide one counterexample.

With the *existential quantifier* $\exists x$, we write $\exists x\, p(x)$ to denote "there exists x (in the domain) such that $p(x)$." Its truth values are defined by:

$$\begin{aligned}\exists x\, p(x) &\equiv \text{T} \quad \text{if } p(x) \text{ is true for some } x \text{ in the domain}\\ &\equiv \text{F} \quad \text{otherwise}\end{aligned}$$

That is, $\exists x\, p(x) \equiv \text{F}$ if $p(x)$ is false for all x in the domain. To prove $\exists x\, p(x)$, it suffices to provide one example x_0 that makes $p(x_0)$ true. To prove it is false, one must prove that $p(x)$ is false for all x in the domain.

Example 1.7. Consider a predicate

$$q(n) \quad = \quad \text{“}2n+1 \text{ is prime,”}$$

where the set of all natural numbers is the underlying domain. Then $\forall n\, q(n)$ denotes "$2n+1$ is prime for all natural numbers n," and $\exists n\, q(n)$ denotes "there is a natural number n such that $q(n)$ is prime." Here, $\forall n\, q(n)$ is false with a counterexample $n_0 = 4$ since $2 \cdot 4 + 1 = 9$ is not prime; however, $\exists n\, q(n)$ is true with $n_0 = 1$ since $2 \cdot 1 + 1 = 3$ is clearly prime.

De Morgan's laws for a predicate $p(x)$ state:

$$\neg\forall x\, p(x) \equiv \exists x\, \neg p(x)$$
$$\neg\exists x\, p(x) \equiv \forall x\, \neg p(x)$$

Example 1.8. Consider a predicate

$$r(x) \quad = \quad \text{"}x \text{ is tall,"}$$

where the domain consists of all people. Then $\neg\forall x\, r(x)$ denotes, "Not everyone is tall, " and it is equivalent to $\exists x\, \neg p(x)$, which denotes, "There is someone who is not tall." On the other hand, $\neg\exists x\, r(x)$ denotes, "There is no one who is tall," and it is equivalent to $\forall x\, \neg p(x)$, which denotes, "Everyone is not tall."

We next consider formal arguments. In propositional logic, an *argument* is a sequence of propositions that begins with a number of given premises and ends with a conclusion.

Consider this argument:

If the demand rises, then companies expand.
If companies expand, then they hire workers.
Therefore, if the demand rises, then companies hire workers.

Here, with

$$\begin{aligned} p &= \text{"The demand rises,"} \\ q &= \text{"Companies expand," and} \\ r &= \text{"Companies hire workers,"} \end{aligned}$$

this argument is an example of

$$\begin{array}{rc} & p \to q \\ & q \to r \\ \hline \therefore & p \to r \end{array} \quad \text{Hypothetical syllogism}$$

The validity of hypothetical syllogism follows from this tautology:

$$((p \to q) \wedge (q \to r)) \to (p \to r).$$

Hypothetical syllogism is one of the eight basic rules of inference, which are summarized on the next page in Table 1.4.

$\begin{array}{rl} & p \\ & p \to q \\ \hline \therefore & q \end{array}$	Modus ponens
$\begin{array}{rl} & \neg q \\ & p \to q \\ \hline \therefore & \neg p \end{array}$	Modus tollens
$\begin{array}{rl} & p \to q \\ & q \to r \\ \hline \therefore & p \to r \end{array}$	Hypothetical syllogism
$\begin{array}{rl} & p \vee q \\ & \neg p \\ \hline \therefore & q \end{array}$	Disjunctive syllogism
$\begin{array}{rl} & p \\ \hline \therefore & p \vee q \end{array}$	Addition
$\begin{array}{rl} & p \wedge q \\ \hline \therefore & p \end{array}$	Simplification
$\begin{array}{rl} & p \\ & q \\ \hline \therefore & p \wedge q \end{array}$	Conjunction
$\begin{array}{rl} & p \vee q \\ & \neg p \vee r \\ \hline \therefore & q \vee r \end{array}$	Resolution

Table 1.4: Rules of inference

Consider this argument:

If the Red Sox win, Sam is happy.
If Sam is happy, then he sleeps well.
Sam does not sleep well.
Therefore, the Red Sox did not win.

Here, if

$$p \quad = \quad \text{"The Red Sox win,"}$$

$$\begin{aligned} q &= \text{“Sam is happy,” and} \\ r &= \text{“Sam sleeps well, ”} \end{aligned}$$

then this argument is formalized as

$$\begin{array}{rl} & p \to q \\ & q \to r \\ & \underline{\neg r \qquad} \\ \therefore & \neg p \end{array}$$

This is not found in Table 1.4, but is it a valid argument? Yes, and here is a formal proof that the conclusion indeed follows from the given premises, using some of the rules of inference from the table:

1.	$p \to q$	a premise
2.	$q \to r$	a premise
3.	$\neg r$	a premise
4.	$p \to r$	by hypothetical syllogism on 1 and 2
5.	$\neg p$	by modus tollens on 3 and 4

Another approach is to use modus tollens twice:

1.	$p \to q$	a premise
2.	$q \to r$	a premise
3.	$\neg r$	a premise
4.	$\neg q$	by modus tollens on 2 and 3
5.	$\neg p$	by modus tollens on 1 and 4

Example 1.9. The following is a proof that $\neg p$ follows from $(p \vee q) \to r$, $\neg s$ and $r \to s$.

1.	$(p \vee q) \to r$	a premise
2.	$\neg s$	a premise
3.	$r \to s$	a premise
4.	$\neg r$	by modus tollens on 2 and 3
5.	$\neg(p \vee q)$	by modus tollens on 1 and 4
6.	$\neg p \wedge \neg q$	by De Morgan's law on 5
7.	$\neg p$	by simplification on 6

In a *proof by contradiction*, we assume, in addition, the negation of the conclusion and derive a contradiction, a proposition that is always false.

Example 1.10. The following is a proof by contradiction that $\neg p$ follows from $(p \vee q) \to r$, $\neg s$ and $r \to s$.

1.	$(p \vee q) \to r$	a premise
2.	$\neg s$	a premise
3.	$r \to s$	a premise
4.	$\neg(\neg p)$	the negation of the conclusion
5.	p	by the double negation law on 4
6.	$p \vee q$	by addition on 5
7.	r	by modus ponens on 1 and 6
8.	s	by modus ponens on 3 and 7
9.	$s \wedge \neg s$	by conjunction on 2 and 8
10.	F	by a negation law on 9

We conclude this section with this simple argument:

If it is raining, then road is wet.
Road is wet.
Therefore, it is raining.

Here, if

$$\begin{aligned} p &= \text{"It is raining," and} \\ q &= \text{"Road is wet,"} \end{aligned}$$

then this argument is formalized as

$$\begin{array}{rl} & p \to q \\ & q \\ \hline \therefore & p \end{array}$$

This is also not found in Table 1.4. Is it a valid argument? It does not seem so. In fact, this is an example of a so-called *fallacy*. Indeed, it cannot be a valid argument because $((p \to q) \wedge q) \to p$ is *not* a tautology.

§2. Proof techniques

A *proof* is a valid argument that establishes the truth of a mathematical proposition. It begins with one or more premises and proceeds using rules of inference to reach a conclusion.

A *diret proof* shows that, for a given premise p and a conclusion q, the conditional statement $p \to q$ is true by showing that, if p is true, then q is also true.

Example 2.1. Consider the statement:

The product of two odd integers is odd.

What are premises? What is then the conclusion? Premises may be: We are given two arbitrary odd integers. The conclusion is that their product is also odd. Here, recall that an odd integer is defined as an integer of the form $2n + 1$ for an integer n.

To begin our proof, let a and b be odd integers. That is, by definition, $a = 2m + 1$ and $b = 2n + 1$ for some integers m and n. Then

$$ab = (2m + 1)(2n + 1) = 4mn + 2m + 2n + 1.$$

Here, $4mn + 2m + 2n + 1 = 2(2mn + m + n) + 1$; so, for $k = 2mn + m + n$, which is clearly an integer,

$$ab = 2k + 1.$$

Therefore, ab is also odd. □

A direct proof may not be always the best approach. For example, consider, for an integer n, the statement:

If n^2 is even, then so is n.

We may begin with $n^2 = 2m$ for some integer $m \geq 0$, but then, how do we prove that $n = \sqrt{2m}$ is even? In this case, it turns out that it is just far easier to consider its contrapositive for an integer n:

If n is odd, then so is n^2.

In fact, this is a special case of what we just proved. In general, for propositions p and q, recall that $p \to q \equiv \neg q \to \neg p$; as such, a *proof by contraposition* shows that $p \to q$ is true by showing that its contrapositive $\neg q \to \neg p$ is true.

As considered earlier in the last section, a *proof by contradiction* begins with the negation of a conclusion and ends with a contradiction.

Example 2.2. Consider the statement:

The sum of rational and irrational numbers is irrational.

To prove this by contradiction, we suppose, conversely, that, given a rational number a and an irrational number α, the sum $b = a + \alpha$ is rational. Since a and b are both rational, there are integers k, ℓ, m and n such that

$$a = \frac{k}{\ell} \text{ and } b = \frac{m}{n},$$

where both ℓ and n are nonzero. Since $\alpha = b - a$,

$$\alpha = \frac{m}{n} - \frac{k}{\ell} = \frac{\ell m - kn}{\ell n},$$

where both $\ell m - kn$ and ℓn are integers, and $\ell n \neq 0$. Thus, α is rational. This is a contradiction. We conclude that the sum of rational and irrational numbers must be irrational. □

Example 2.3. Consider the statement:

$\sqrt{2}$ *is irrational.*

To prove this, we suppose, conversely, that $\sqrt{2}$ is rational. That is, there are integers m and n such that

$$\sqrt{2} = \frac{m}{n},$$

where $n \neq 0$. We may also assume that m and n have no common divisor other than 1. This then implies that

$$2 = \frac{m^2}{n^2},$$

or, equivalently, $m^2 = 2n^2$. That is, m^2 is even. For such an m, we have just proved that m must also be even. So, $m = 2\ell$ for some integer ℓ. Then, since $m^2 = 2n^2$, it follows that $4\ell^2 = 2n^2$. That is, $n^2 = 2\ell^2$. By the same argument as m's, since n^2 is even, n must also be even.

In sum, both m and n are even. This is a contradiction since m and n have no common divisor other than 1. Therefore, we conclude that $\sqrt{2}$ must be irrational. □

Some proofs require multiple arguments under a number of case-by-case scenarios. We call such a proof method a *proof by cases*.

Example 2.4. Consider, for an integer n, the statement:

n, $n+1$ or $n+2$ is divisible by 3.

To prove this, we consider all possible remainders of the integer division of an integer n by 3. By the so-called *division algorithm*, which we will study in depth later, this results in three possible remainders: 0, 1 and 2. That is, $n = 3k$, $n = 3k+1$ or $n = 3k+2$ for an integer k. We will consider each of the three cases separately.

We suppose first that $n = 3k$. In this case, evidently, n itself is divisible by 3.

Next, we suppose that $n = 3k+1$. In this case,

$$n + 2 = (3k + 1) + 2 = 3k + 3 = 3(k + 1),$$

proving that $n+2$ is divisible by 3.

Finally, the last case to consider is when $n = 3k+2$. In this case,

$$n + 1 = (3k + 2) + 1 = 3k + 3 = 3(k + 1).$$

That is, $n+1$ is divisible by 3.

As demonstrated in the three cases, we conclude that n, $n+1$ or $n+2$ is divisible by 3. □

Exact mathematical propositions often assert equivalence in biconditional *if-and-only-if* statements. In general, for propositions p and q, recall from §1 that $p \leftrightarrow q \equiv (p \to q) \wedge (q \to p)$; that is, proving $p \leftrightarrow q$ is equivalent to proving $p \to q$ and $q \to p$. In practice, to prove such an if-and-only-if statement, we most often take the latter approach and prove two conditional statements in both directions separately.

Example 2.5. Consider, for an integer n, the statement:

n is even if and only if $n + 1823$ is odd.

For this, we prove, separately, the following two conditional statements:

(i) *If n is even, then $n + 1823$ is odd.*

(ii) *If $n + 1823$ is odd, then n is even.*

To prove (i), we first assume that $n = 2k$ for an integer k. Then

$$n + 1823 = 2k + 1823 = 2(k + 911) + 1,$$

proving that $n + 1823$ is odd.

We leave a proof of (ii) as an exercise. □

§3. Sets

A *set* is an unordered collection of distinct objects called *elements* (or *members*). By "distinct", we mean that there can be no multiple occurances of the same element. A set is often described by listing its elements between a pair of curly braces, and it is unordered in the sense that the elements can be listed in any order, e.g., $\{0, 1, 2\} = \{1, 2, 0\}$.

In general, it is customary to use uppercase letters to denote sets and lowercase letters to denote elements. Some uppercase letters in the boldface are reserved to denote commonly-used sets of numbers; for example,

$\mathbf{Z} = \{\ldots, -3, -2, -1, 0, 1, 2, 3, \ldots\}$, the set of all integers,

$\mathbf{N} = \{0, 1, 2, 3, \ldots\}$, the set of all natural numbers,

$\mathbf{R}$ is the set of all real numbers, and

$\mathbf{Q}$ is the set of all rational numbers.

If a is an element of a set A, then we say *a belongs to A* and write $a \in A$; otherwise, we write $a \notin A$. For example, $\sqrt{2} \in \mathbf{R}$, but, as we have proven in Example 2.3, $\sqrt{2} \notin \mathbf{Q}$.

An important way to define sets is to a variable and its predicates that specify properties in the following format:

$$\{\text{a variable} : \text{its predicates specifying properties}\}$$

In this case, it consists of all elements that satisfy the given properties.

For example,

$$\{n : n \in \mathbf{Z} \text{ and } n \text{ is even}\}$$

defines the set of all even integers. This notation, often called the *set-builder* notation, is somewhat loosely used in that

$\{n \in \mathbf{Z} : n \text{ is even}\}$,

$\{n : n = 2k \text{ for some } k \in \mathbf{Z}\}$ and

$\{2n : n \in \mathbf{Z}\}$

also all define the same set of all even integers.

Intervals are infinite sets of all real numbers between given upper and lower bounds. In particular, given $a, b \in \mathbf{R}$ such that $a \leq b$,

$[a, b] = \{x \in \mathbf{R} : a \leq x \leq b\}$,

$$(a,b] = \{x \in \mathbf{R} : a < x \leq b\},$$
$$[a,b) = \{x \in \mathbf{R} : a \leq x < b\} \text{ and}$$
$$(a,b) = \{x \in \mathbf{R} : a < x < b\};$$

here, we often call $[a,b]$ the *closed* interval, $(a,b]$ and $[a,b)$ *half-open* intervals and (a,b) the *open* interval (or *segment*) between a and b.

Let S and T be sets. We say S is a *subset* of T, denoted by $S \subseteq T$, if each element of S is an element of T. The sets S and T are *equal*, denoted by $S = T$, if $S \subseteq T$ and $T \subseteq S$. If $S \subseteq T$ but $S \neq T$, then S is a *proper subset* of T, denoted by $S \subset T$.

The *empty set*, denoted by $\emptyset$, is the unique set that has no element.

3.1. *For any set S, the empty set $\emptyset \subseteq S$.* □

The *power set* of a set S, denoted by $\mathcal{P}(S)$ (or 2^S), is the set of all subsets of S. For any set S, clearly, $\emptyset \in \mathcal{P}(S)$ as well as $S \in \mathcal{P}(S)$.

Example 3.2. If $S = \{0, 1\}$, then

$$\mathcal{P}(S) = \{\emptyset, \{0\}, \{1\}, \{0, 1\}\}.$$

The *cartesian product* of sets S and T is the set of all ordered pairs (s,t), where $s \in S$ and $t \in T$, and denoted by $S \times T$.

Example 3.3. If $S = \{0, 1\}$ and $T = \{a, b, c\}$, then

$$S \times T = \{(0,a), (0,b), (0,c), (1,a), (1,b), (1,c)\}.$$

If a set S has a finite number of elements, then the *size* (or *order*) of S, denoted by $|S|$, is the number of elements of S.

3.4. *If S and T are finite sets, then*

(i) $|\emptyset| = 0$,
(ii) $|S \times T| = |S| \times |T|$ *and*
(iii) $|\mathcal{P}(S)| = 2^{|S|}$. □

For sets A and B,

(i) the *union* of A and B is $A \cup B = \{x : x \in A \text{ or } x \in B\}$,

$A \cap U = A$ $A \cup \emptyset = A$	Identity laws
$A \cup U = U$ $A \cap \emptyset = \emptyset$	Domination laws
$A \cup A = A$ $A \cap A = A$	Idempotent laws
$\overline{\overline{A}} = A$	Complementation law
$A \cup B = B \cup A$ $A \cap B = B \cap A$	Commutative laws
$A \cup (B \cup C) = (A \cup B) \cup C$ $A \cap (B \cap C) = (A \cap B) \cap C$	Associative laws
$A \cup (B \cap C) = (A \cup B) \cap (A \cup C)$ $A \cap (B \cup C) = (A \cap B) \cup (A \cap C)$	Distributive laws
$\overline{A \cap B} = \overline{A} \cup \overline{B}$ $\overline{A \cup B} = \overline{A} \cap \overline{B}$	De Morgan's laws
$A \cup (A \cap B) = A$ $A \cap (A \cup B) = A$	Absorption laws
$A \cup \overline{A} = U$ $A \cap \overline{A} = \emptyset$	Complement laws

Table 3.1: Set identities

(ii) the *intersection* of A and B is $A \cap B = \{x : x \in A \text{ and } x \in B\}$, and

(iii) the *difference* of A and B is $A - B = \{x : x \in A \text{ but } x \notin B\}$.

The *universal set* is the set of all elements under consideration and often denoted by U. The *complement* of a set A is $\overline{A} = \{x : x \in U \text{ but } x \notin A\}$; that is, $\overline{A} = U - A$.

Table 3.1 above summarizes set identities analogous to the logical equivalences in Table 1.2. Indeed, these identities all follow from the corresponding logical equivalences, as we demonstrate some of them below.

Proof of the distributive law $A \cup (B \cap C) = (A \cup B) \cap (A \cup C)$. We will prove $A \cup (B \cap C) \subseteq (A \cup B) \cap (A \cup C)$ and then $(A \cup B) \cap (A \cup C) \subseteq A \cup (B \cap C)$.

First, let $x \in A \cup (B \cap C)$. By definition, x satisfies

$$x \in A \vee (x \in B \wedge x \in C),$$

which is logically equivalent to, by the distributive law from Table 1.2,

$$(x \in A \vee x \in B) \wedge (x \in A \vee x \in C).$$

The latter implies that $x \in A \cup B$ and $x \in A \cup C$, which in turn means that $x \in (A \cup B) \cap (A \cup C)$. Thus, $A \cup (B \cap C) \subseteq (A \cup B) \cap (A \cup C)$.

Next, let $y \in (A \cup B) \cap (A \cup C)$. Here, y satisfies

$$(y \in A \vee y \in B) \wedge (y \in A \vee y \in C).$$

By the same distributive law from Table 1.2, $y \in A$ or $y \in B \cap C$, which in turn means that $y \in A \cup (B \cap C)$. Thus, $(A \cup B) \cap (A \cup C) \subseteq A \cup (B \cap C)$. We can now conclude that $A \cup (B \cap C) = (A \cup B) \cap (A \cup C)$. $\square$

De Morgan's laws can also be proved using the same approach, but it can be shortened in the following way.

Proof of De Morgan's law $\overline{A \cap B} = \overline{A} \cup \overline{B}$. We first note that, by definition, for $x \in U$, $x \in \overline{A \cap B}$ if and only if $x \notin A \cap B$, or, equivalently, x satisfies

$$p(x) = \neg((x \in A) \wedge (x \in B)).$$

On the other hand, for $x \in U$, $x \in \overline{A} \cup \overline{B}$ if and only if x satisfies

$$q(x) = \neg(x \in A) \vee \neg(x \in B).$$

By De Morgan's law from Table 1.2, $p(x)$ and $q(x)$ are logically equivalent. Therefore, for $x \in U$, $x \in \overline{A \cap B}$ if and only if $x \in \overline{A} \cup \overline{B}$. We have just shown that $\overline{A \cap B}$ and $\overline{A} \cup \overline{B}$ are subsets of one another. We conclude that $\overline{A \cap B} = \overline{A} \cup \overline{B}$. $\square$

§4. Functions and sequences

Given nonempty sets S and T, a *function* $f : S \to T$ is an assignment of exactly one element of T to every element of S; to emphasize this clear and umambiguous property, we also say f is *well defined*. For such an $f : S \to T$, we say f *maps* S into T and call S the *domain* and T the *codomain* of f.

Let $f : S \to T$ be a function. The unique element of T assigned to $s \in S$ by f, called the *image* of s under f, is denoted by $f(s)$. For a subset $S' \subseteq S$, the set $\{f(s') : s' \in S'\}$, called the *image* of S' under f, is denoted by $f(S')$. The subset $f(S) \subseteq T$, the image of S under f, is also called the *range* of f.

Definition. Let $f : S \to T$ be a function.

(i) f is *one-to-one* (or *injective*) if, for all $s, s' \in S$, whenever $s \neq s'$, $f(s) \neq f(s')$.

(ii) f is *onto* (or *surjective*) if $f(S) = T$.

(iii) f is *one-to-one correspondence* (or *bijective*) if f is both one-to-one and onto.

Let $f : S \to T$ be a function. To prove that f is one-to-one, it suffices to prove that, for all $s, s' \in S$, whenever $f(s) = f(s')$, $s = s'$. To prove that f is onto, it suffices to prove that, for any $t \in T$, there is $s \in S$ such that $f(s) = t$.

In the following, as usual, $\mathbf{Z}$ is the set of all integers, $\mathbf{N}$ is the set of all natural numbers, and $\mathbf{R}$ is the set of all real numbers.

Example 4.1. Consider a function $f : \mathbf{Z} \to \mathbf{Z}$ defined by $f(n) = n + 1$ for $n \in \mathbf{Z}$. This f is both one-to-one and onto.

First, to see that it is one-to-one, consider $m, n \in \mathbf{Z}$ such that $f(m) = f(n)$. That is, $m + 1 = n + 1$ and thus $m = n$. Therefore, f is one-to-one.

To prove that it is onto, consider $\ell \in \mathbf{Z}$ and let $k = \ell - 1$. Here, $f(k) = k + 1 = (\ell - 1) + 1 = \ell$. Since $k \in \mathbf{Z}$, this proves that f is onto.

Example 4.2. Consider a function $g : \mathbf{R} \to \mathbf{R}$ defined by $g(x) = x^2$ for $x \in \mathbf{R}$. This g is neither one-to-one nor onto.

To see that it is not one-to-one, we notice that, while $1 \neq -1$, $g(1) = 1 = g(-1)$. To prove that it is not onto, it suffices to note that there is no real number whose square is a negative number, say, -1. For both properties, giving such counterexamples is sufficient.

For $x \in \mathbf{R}$, the *floor of* x, denoted by $\lfloor x \rfloor$, is the largest integer $\leq x$, and the *ceiling of* x, denoted by $\lceil x \rceil$, is the smallest integer $\geq x$. For example, $\lfloor \sqrt{2} \rfloor = 1$, whereas $\lceil \sqrt{2}\, \rceil = 2$. Clearly, for any $n \in \mathbf{Z}$, $\lfloor n \rfloor = \lceil n \rceil = n$.

Example 4.3. Consider a function $h : \mathbf{R} \to \mathbf{Z}$ defined by $h(x) = \lfloor x \rfloor$ for $x \in \mathbf{R}$. This h is onto but not one-to-one.

To see that it is onto, consider $n \in \mathbf{Z}$. Then $h(n) = \lfloor n \rfloor = n$. This proves that h is onto.

On the other hand, while $\sqrt{2} \neq 1$, $h(\sqrt{2}) = 1 = h(1)$. This proves that h is not one-to-one.

Given functions $f : S \to T$ and $g : T \to U$, the *composition* (or *product*) of f and g is the function $g \circ f : S \to U$ defined by

$$(g \circ f)(x) = g(f(x))$$

for $x \in S$.

Example 4.4. Consider functions $f : \mathbf{Z} \to \mathbf{N}$ defined by $f(n) = n^2$ for $n \in \mathbf{Z}$ and $g : \mathbf{N} \to \mathbf{R}$ defined by $g(n) = \sqrt[3]{n}$ for $n \in \mathbf{N}$. The composition of f and g is the function $g \circ f : \mathbf{Z} \to \mathbf{R}$ defined by

$$(g \circ f)(n) = g(f(n)) = g(n^2) = \sqrt[3]{n^2} = n^{\frac{2}{3}}$$

for $n \in \mathbf{Z}$.

Function composition is associative in the following sense.

Proposition 4.5. *If $f : S \to T$, $g : T \to U$ and $h : U \to V$, are functions, then $h \circ (g \circ f) = (h \circ g) \circ f$.*

Proof. We first note that, by definition, since $g \circ f$ is a function from S to U, $h \circ (g \circ f)$ is a function from S to V. Similarly, $(h \circ g) \circ f$ is also a function from S to V. Thus, it remains to prove that $h \circ (g \circ f) = (h \circ g) \circ f$.

It suffices to show that, for any $s \in S$, $(h \circ (g \circ f))(s) = ((h \circ g) \circ f)(s)$. By definition, for any $s \in S$, $(h \circ (g \circ f))(s) = h((g \circ f)(s)) = h(g(f(s)))$, whereas $((h \circ g) \circ f)(s) = (h \circ g)(f(s)) = h(g(f(s)))$. That is, for any $s \in S$, the elements $(h \circ (g \circ f))(s)$ and $((h \circ g) \circ f)(s)$ are equal. We conclude that the two compositions are equal. $\square$

Definition. An *inverse* of a function $f : S \to T$ is a function $f^{-1} : T \to S$ such that

(i) $f^{-1}(f(s)) = s$ for all $s \in S$ and
(ii) $f(f^{-1}(t)) = t$ for all $t \in T$.

Not all functions have inverses. We call those that do *invertible.*

Theorem 4.6. *A function f is invertible if and only if f is one-to-one correspondence.*

Proof. We will first prove that, if a function f is invertible, then f is one-to-one correspondence. Suppose that a function $f : S \to T$ has an inverse

$f^{-1} : T \to S$. If, for some $s, s' \in S$, $f(s) = f(s')$, then

$$s = f^{-1}(f(s)) = f^{-1}(f(s')) = s',$$

which proves that f is one-to-one. To see that f is onto, let $t \in T$ and $s = f^{-1}(t)$. Then

$$f(s) = f(f^{-1}(t)) = t,$$

which proves that f is onto.

We will next prove that, if a function f is one-to-one correspondence, then f is invertible. To begin, consider a one-to-one correspondence function $f : S \to T$. Since f is onto, for any $t \in T$, there is $s \in S$ such that $f(s) = t$; in fact, since f is also one-to-one, there is only one such s. Here, for given $t \in T$, we define, with t's unique $s \in S$ such that $f(s) = t$,

$$g(t) = s.$$

This g defines a function from T to S, and it is indeed uniquely defined. We now claim that $g = f^{-1}$, the inverse of f. For this, we must prove that $g(f(s)) = s$ for all $s \in S$ and that $f(g(t)) = t$ for all $t \in T$. First, for any given $s \in S$, there is $t \in T$ such that $f(s) = t$ and thus $g(t) = s$ so that $g(f(s)) = g(t) = s$. Conversely, for any given $t \in T$, there is $s \in S$ such that $f(s) = t$ and thus $g(t) = s$ so that $f(g(t)) = f(s) = t$. We have shown that $g = f^{-1}$. □

For the set of all natural numbers $\mathbf{N}$, a *sequence* is a function $s : \mathbf{N} \to S$ for some set S, where, for $n \in \mathbf{N}$, each image $s(n)$ is called the *nth term* of the sequence. For such a sequence, it is also customary to write, for $n \in \mathbf{N}$, $s_n = s(n)$ and $\{s_n\}_{n \in \mathbf{N}} = s(\mathbf{N})$.

An *arithmetic progression* is a sequence $\{a_n\}_{n \in \mathbf{N}}$ defined by

$$a_n = a_0 + dn$$

for $n \in \mathbf{N}$, where a_0 is the *initial term*, and d is the *common difference*.

A *geometric progression* is a sequence $\{g_n\}_{n \in \mathbf{N}}$ defined by

$$g_n = g_0 r^n$$

for $n \in \mathbf{N}$, where g_0 is the initial term, and r is the *common ratio*.

Example 4.7.

(a) Let $\{a_n\}_{n \in \mathbf{N}}$ be an arithmetic progression defined by $a_0 = 0$ and $d = 2$, i.e., $a_n = 0 + 2n$ for $n \in \mathbf{N}$. Here,

$$a_0 = 0, a_1 = 2, a_2 = 4, a_3 = 6, \ldots$$

That is, this defines the sequence of all nonnegative multiples of 2.

(b) Let $\{g_n\}_{n \in \mathbf{N}}$ be a geometric progression defined by $g_0 = 1$ and $r = 2$, i.e., $g_n = 1 \cdot 2^n$ for $n \in \mathbf{N}$. Here,

$$g_0 = 1, g_1 = 2, g_2 = 4, g_3 = 8, \ldots$$

That is, this defines the sequence of all powers of 2. The terms of a sequence need not be always increasing. For example, if $r = -1$, then the terms of this sequence become

$$g_0 = 1, g_1 = -1, g_2 = 1, g_3 = -1, \ldots,$$

alternating with only 1 and -1; in particular, for $n \in \mathbf{N}$, $g_n = 1$ if n is even, and $g_n = -1$ if n is odd.

§5. Algorithms and complexity

The notion of *algorithms* is fundamental to all of computer science. Simply put, an algorithm is a finite sequence of precise and simple instructions. Often, we also consider algorithms with respect to these general features (cf. [1, §1.1]):

(i) *Finiteness.* An algorithm is a finite sequence of finite statements. It must also terminate after a finite number, however large, of steps.

(ii) *Definiteness.* Each step of an algorithm must be precisely defined, in rigorous and unambiguous terms.

(iii) *Input.* Zero or more inputs are initially given before execution.

(iv) *Output.* An algorithms generates one or more outputs.

(v) *Effectiveness.* Each operation defined must be sufficiently simple that, in principle, it can be completed exactly by human manually, given sufficient time.

We now consider a very simple problem that admits two very different approaches.

Searching.
Given: *a sequence* $S = (s_1, \ldots, s_n)$ *and a target* x*.*
Find: *an index* i *such that* $x = s_i$ *if such an* i *exists, or* 0 *otherwise.*

A natural approach to solve this problem is so-called *linear* (or *sequential*) *search*, which simply compares x against $s_i \in S$ for $i = 1, \ldots, n$:

```
function linear-search(S, x)
    begin
        i := 1;
        while (i ≤ n and x ≠ s_i) do
            i := i + 1;
        if (i ≤ n) then return i;
        else return 0;
    end.
```

If, in addition, it is known that S is *sorted* such that $s_1 \leq \cdots \leq s_n$, then *binary search*, which repeatedly compares x against the middle of a reduced search range and bisects it to reduce it further, is more effective:

```
function binary-search(S, x)
    begin
        ℓ := 1;
        r := n;
        while (ℓ < r) do
            begin
                m := ⌊(ℓ + r)/2⌋;
                if (x > s_m) then ℓ := m + 1;
                else r := m;
            end;
        if (x = s_ℓ) then return ℓ;
        else return 0;
    end.
```

When analyzing efficiency of an algorithm formally, we most often focus on the time it takes in the worst case. We express time in a function of the input size n, where the time corresponds to the number of "unit operations" executed. We call such a function the (*time*) *complexity* of the algorithm. In case of searching, we measure time by the number of element comparisons performed.

For linear search, it is easy to see that the number of element comparisons performed is just one in the best case (when $x = s_1$) and n in the worst case (when $x = s_n$ or x does not exist in S).

For binary search, for simplicity, we assume that $n = 2^k$ (i.e., $k = \log_2 n$) for some integer $k \geq 0$. The length of the search range is halved each time. In particular, the length is $r - \ell + 1$. Initially, this is $n = 2^k$. In the next round, this becomes 2^{k-1}. It is halved until it becomes 1. That is, the loop runs k times, with just one comparison each time. With one more comparison at the end, the total number of comparisons is $k + 1 = \log_2 n + 1$.

When analyzing algorithms, it is often useful to consider efficiency with respect to broad families of functions, such as the linear and quadratic families. To focus on growth rates, as represented by familiar expressions

$$1,\ \log_2 n,\ n,\ n \log_2 n,\ n^2,\ n^3,\ \ldots,\ 2^n,\ n!,\ \ldots,$$

we will use the so-called *O notation*.

As usual, let $\mathbf{N}$ denote the set of all natural numbers and $\mathbf{R}^{\geq 0}$ denote the set of all nonnegative real numbers. For simplicity, in what follows, we will focus on functions mapping $\mathbf{N}$ into $\mathbf{R}^{\geq 0}$.

Definition. For functions f and g mapping $\mathbf{N}$ into $\mathbf{R}^{\geq 0}$, we say $f(n)$ *is* $O(g(n))$ for $n \in \mathbf{N}$ if there are constants $c > 0$ and $n_0 \geq 0$ such that

$$f(n) \leq cg(n)$$

for all integers $n \geq n_0$.

The general goal is to categorize a given function f under another function g that represents a well-known growth rate by approximating f from above with g.

Example 5.1.

(a) $f(n) = 2n + 1$ *is* $O(n)$ *for* $n \in \mathbf{N}$. To see this, we notice that

$$f(n) = 2n + 1 \leq 2n + n = 3n$$

for all integers $n \geq 1$ (here, we must exclude zero as it would otherwise imply $1 \leq 0$). That is, with $g(n) = n$, $c = 3$ and $n_0 = 1$, $f(n) \leq cg(n)$ for all integers $n \geq n_0$.

(b) $f(n) = 3n^2 + 2n \log_2 n + 1$ *is* $O(n^2)$ *for* $n \in \mathbf{N}$. For this,

$$f(n) = 3n^2 + 2n \log_2 n + 1 \leq 3n^2 + 2n^2 + n^2 = 6n^2$$

for all integers $n \geq 1$ (here, we exclude zero again for the same reason). That is, with $g(n) = n^2$, $c = 6$ and $n_0 = 1$, $f(n) \leq cg(n)$ for all integers $n \geq n_0$.

(c) $f(n) = 1 + 2 + \cdots + n$ *is* $O(n^2)$ *for* $n \in \mathbf{N}$. For this, we recall that

$$f(n) = 1 + 2 + \cdots + n = \frac{n(n+1)}{2},$$

where

$$\frac{n(n+1)}{2} = \frac{n^2}{2} + \frac{n}{2} \leq \frac{n^2}{2} + \frac{n^2}{2} = n^2,$$

for all integers $n \geq 0$. That is, with $g(n) = n^2$, $c = 1$ and $n_0 = 0$, $f(n) \leq cn^2$ for all integers $n \geq n_0$.

The O notation naturally subsumes constants. It also preserves addition and multiplication and is transitive in the following sense:

Theorem 5.2. *Let f, g, f', g' and h be functions mapping $\mathbf{N}$ into $\mathbf{R}^{\geq 0}$.*

(i) *For $n \in \mathbf{N}$, if $f(n)$ is $O(g(n))$, then $kf(n)$ is also $O(g(n))$ for any constant $k > 0$.*

(ii) *For $n \in \mathbf{N}$, if $f(n)$ is $O(g(n))$, and $f'(n)$ is $O(g'(n))$, then*

  (a) *$f(n) + f'(n)$ is $O(g(n) + g'(n))$, and*

  (b) *$f(n)f'(n)$ is $O(g(n)g'(n))$.*

(iii) *For $n \in \mathbf{N}$, if $f(n)$ is $O(g(n))$, and $g(n)$ is $O(h(n))$, then $f(n)$ is $O(h(n))$.*

Proof. To prove (i), for given f and g, consider constants $c > 0$ and $n_0 \geq 0$ such that $f(n) \leq cg(n)$ for all integers $n \geq n_0$. Let k be a constant > 0. If $c' = kc$, then $kf(n) \leq c'g(n)$ for all integers $n \geq n_0$. Thus, $kf(n)$ is $O(g(n))$ for $n \in \mathbf{N}$.

To prove (ii), for given f, g, f' and g', consider positive constants c and c' and nonnegative constants n_0 and n'_0 such that $f(n) \leq cg(n)$ for all integers

$n \geq n_0$ and $f'(n) \leq c'g'(n)$ for all integers $n \geq n'_0$. If $c_1 = \max(c, c')$, $c_2 = cc'$ and $n_1 = \max(n_0, n'_0)$, then

$$f(n) + f'(n) \leq cg(n) + c'g(n) \leq c_1(g(n) + g'(n))$$

and

$$f(n)f'(n) \leq cg(n)c'(g) = c_2 g(n)g'(n)$$

for all integers $n \geq n_1$, proving (ii).

To prove (iii), for given f, g and h, consider positive constants c and c' and nonnegative constants n_0 and n'_0 such that $f(n) \leq cg(n)$ for all integers $n \geq n_0$ and $g(n) \leq c'h(n)$ for all integers $n \geq n'_0$. If $c_1 = cc'$ and $n_1 = \max(n_0, n'_0)$, then

$$f(n) \leq cg(n) \leq cc'h(n) = c_1 h(n)$$

for all integers $n \geq n_1$, proving (iii). □

References

[1] D. E. Knuth, *Fundamental algorithms*, 3rd ed., The Art of Computer Programming, vol. 1, Addison-Wesley, Reading, Mass., 1997.

[2] K. H. Rosen, *Discrete mathematics and its applications*, 8th ed., McGraw-Hill, New York, 2019.

Trinity College
Hartford, Connecticut