1. BASIC MATHEMATICAL OBJECTS

1.1 Sets

- Ways to define a set:

  Enumerate its elements
  Give a property that characterizes its elements

- Examples of defining a set by enumerating its elements:

  $A = \{11, 12, 21, 22\}$
  $B = \{3, 5, 7, 9, \ldots\}$

- Examples of defining a set via a property:

  $A = \{x \mid x$ is a two-digit integer, each of whose digits is 1 or 2$\}$
  $B = \{x \mid x$ is an odd integer greater than 1$\}$

- The notation $x \in A$ indicates that x is an element of the set A.

- Example of using $\in$ in the definition of a set:

  $C = \{x \mid x \in B \text{ and } x \leq 11\}$
  $C = \{x \in B \mid x \leq 11\}$

- Set definitions are often abbreviated. The definition

  $D = \{x \mid$ there exist integers i and j, both ≥ 0, with $x = 3i + 7j\}$

  could be written as

  $D = \{3i + 7j \mid i, j$ are nonnegative integers$\}$

  or even

  $D = \{3i + 7j \mid i, j \in \mathcal{N}\}$

  where $\mathcal{N}$ is the set of nonnegative integers, or *natural numbers*.

Subsets

- If A and B are sets, A is a *subset* of B (written $A \subseteq B$) if every element of A is an element of B.

- Two sets are equal if they have the same elements, which is the same as saying that each is a subset of the other.

- One way to prove that $A = B$ is to prove that $A \subseteq B$ and $B \subseteq A$.

Operations on Sets

- The *complement* of a set A is the set A' of everything that is not an element of A:

  $A' = \{x \in U \mid x \notin A\}$

  U is the "universal" set of all possible elements. $\notin$ means "is not an element of."

- Definitions of the *union, intersection,* and *difference* of two sets:

  $$\begin{aligned} A \cup B &= \{x \mid x \in A \text{ or } x \in B\} \\ A \cap B &= \{x \mid x \in A \text{ and } x \in B\} \\ A - B &= \{x \mid x \in A \text{ and } x \notin B\} \\ &= \{x \mid x \in A\} \cap \{x \mid x \notin B\} \\ &= A \cap B' \end{aligned}$$

- Examples:

  $\{1, 2, 3, 4\} \cup \{2, 4, 6, 8\} = \{1, 2, 3, 4, 6, 8\}$
  $\{1, 2, 3, 4\} \cap \{2, 4, 6, 8\} = \{2, 4\}$
  $\{1, 2, 3, 4\} - \{2, 4, 6, 8\} = \{1, 3\}$

- The symbol $\varnothing$ is used to represent the *empty set:* the set containing no elements.

- If $A \cap B = \varnothing$, then the sets A and B are said to be *disjoint*.

- If $\mathcal{C}$ is a collection of subsets of a set, the elements of $\mathcal{C}$ are said to be *pairwise disjoint* if for any two distinct elements A and B of $\mathcal{C}$, $A \cap B = \varnothing$.

Set Identities

- There are many standard set identities:

  Commutative laws:

  $$A \cup B = B \cup A \tag{1.1}$$
  $$A \cap B = B \cap A \tag{1.2}$$

  Associative laws:

  $$A \cup (B \cup C) = (A \cup B) \cup C \tag{1.3}$$
  $$A \cap (B \cap C) = (A \cap B) \cap C \tag{1.4}$$

  Distributive laws:

  $$A \cup (B \cap C) = (A \cup B) \cap (A \cup C) \tag{1.5}$$
  $$A \cap (B \cup C) = (A \cap B) \cup (A \cap C) \tag{1.6}$$

  Idempotent laws:

  $$A \cup A = A \tag{1.7}$$
  $$A \cap A = A \tag{1.8}$$

  Absorptive laws:

  $$A \cup (A \cap B) = A \tag{1.9}$$
  $$A \cap (A \cup B) = A \tag{1.10}$$

  De Morgan laws:

  $$(A \cup B)' = A' \cap B' \tag{1.11}$$
  $$(A \cap B)' = A' \cup B' \tag{1.12}$$

  Other laws involving complements:

  $$(A')' = A \tag{1.13}$$
  $$A \cap A' = \varnothing \tag{1.14}$$
  $$A \cup A' = U \tag{1.15}$$

  Other laws involving the empty set:

  $$A \cup \varnothing = A \tag{1.16}$$
  $$A \cap \varnothing = \varnothing \tag{1.17}$$

  Other laws involving the universal set:

  $$A \cup U = U \tag{1.18}$$
  $$A \cap U = A \tag{1.19}$$

Set Identities (Continued)

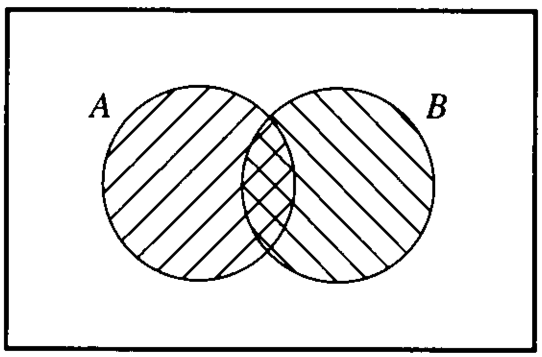
- A proof of the second De Morgan law:

  To show $(A \cap B)' \subseteq A' \cup B'$, let $x \in (A \cap B)'$. Then by definition of complement, $x \notin A \cap B$. By definition of intersection, x is not an element of both A and B; therefore, either $x \notin A$ or $x \notin B$. Thus, $x \in A'$ or $x \in B'$, and so $x \in A' \cup B'$.
  To show $A' \cup B' \subseteq (A \cap B)'$, let $x \in A' \cup B'$. Then $x \in A'$ or $x \in B'$. Therefore, either $x \notin A$ or $x \notin B$. Thus x is not an element of both A and B, and so $x \notin A \cap B$. Therefore, $x \in (A \cap B)'$.

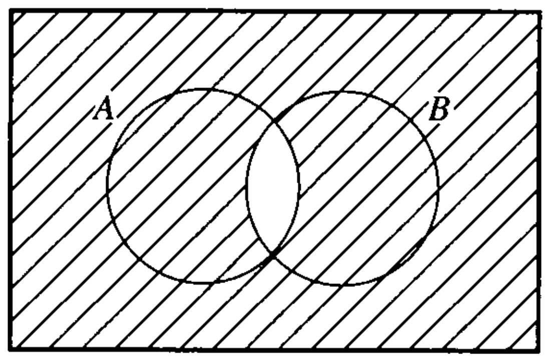
Venn Diagrams

- Set operations are often illustrated by *Venn diagrams.* A large region represents the universe; within this region, overlapping circles represent sets.

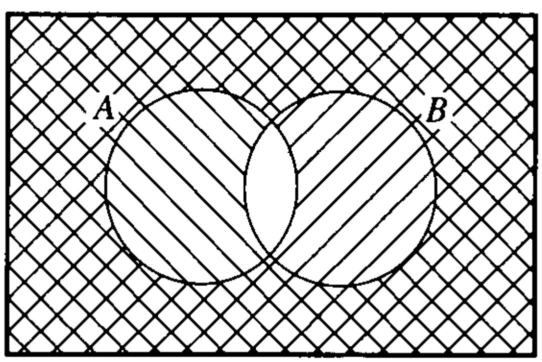
- Figure (a) shows a basic Venn diagram. Figures (b) and (c) illustrate the second De Morgan identity.



(a)



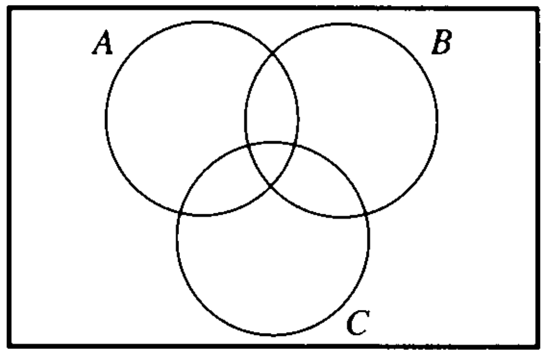
(b)



(c)

Venn Diagrams (Continued)

- Venn diagrams can be used with more than two sets. An unshaded Venn diagram for three sets:



- The *symmetric difference* of sets A and B is defined by

  $A \oplus B = (A - B) \cup (B - A)$

- Venn diagrams can be used to show that symmetric difference satisfies the associative law:

  $A \oplus (B \oplus C) = (A \oplus B) \oplus C$

- Venn diagrams have limited usefulness, so it is important to be able to work with identities as well.

- Example: Simplifying $A \cup (B - A)$:

$$\begin{aligned}
A \cup (B - A) &= A \cup (B \cap A') && \text{(by definition of } -\text{)} \\
&= (A \cup B) \cap (A \cup A') && \text{(by (1.5))} \\
&= (A \cup B) \cap U && \text{(by (1.15))} \\
&= A \cup B && \text{(by (1.19))}
\end{aligned}$$

Unions and Intersections of Multiple Sets

- Because union is an associative operation, it's not necessary to use parentheses when taking the union of more than two sets. In particular, the union of three sets can be defined as

$$A \cup B \cup C = \{x \mid x \in A \text{ or } x \in B \text{ or } x \in C\}$$
$$= \{x \mid x \text{ is an element of at least one of the sets } A, B, \text{ and } C\}$$

- In general:

$$\bigcup_{i=1}^{n} A_i = \{x \mid x \in A_i \text{ for at least one } i \text{ with } 1 \le i \le n\}$$

$$\bigcup_{i=1}^{\infty} A_i = \{x \mid x \in A_i \text{ for at least one } i \ge 1\}$$

$$\bigcap_{i=1}^{n} A_i = \{x \mid x \in A_i \text{ for every } i \text{ with } 1 \le i \le n\}$$

- If $P(i)$ is some condition involving i, then

$$\bigcup_{P(i)} A_i = \{x \mid x \in A_i \text{ for at least one } i \text{ satisfying } P(i)\}$$

Power Sets and Cartesian Products

- The elements of a set may themselves be sets.
- For any set A, the set of all subsets of A is referred to as the *power set* of A and is often written 2^A.
- If A has n elements, then 2^A has 2^n elements.
- If $A = \{1, 2, 3\}$, then

  $2^A = \{\varnothing, \{l\}, \{2\}, \{3\}, \{1, 2\}, \{2, 3\}, \{2, 3\}, \{1, 2, 3\}\}$
- The *Cartesian product* of two sets A and B is defined as follows:

  $A \times B = \{(a, b) \mid a \in A \text{ and } b \in B\}$

  Each element (a, b) is said to be an *ordered pair.*
- An example of a Cartesian product:

  $\{a, b\} \times \{b, c, d\} = \{(a, b), (a, c), (a, d), (b, b), (b, c), (b, d)\}$
- More generally, the set of all "ordered n-tuples" $(a_1, a_2, \ldots, a_n)$, where $a_i \in A_i$ for each i, is denoted by $A_1 \times A_2 \times \ldots \times A_n$.

1.2.1 Propositions and Logical Connectives

- A *proposition* is a declarative statement that is sufficiently objective, meaningful, and precise to have a *truth value* (true or false):

  Fourteen is an even integer.
  Winnipeg is the largest seaport in Canada.
  $0 = 0$.

- The following propositions involve one or more *free variables:*

  $x^2 < 4$.
  $a^2 + b^2 = 3$.
  He has never held public office.

  Once each free variable is given a specific value from an appropriate *domain*, or *universe*, each proposition will have a truth value.

Logical Connectives

- Compound propositions are created by applying *logical connectives* to one or more simpler propositions.

- The *conjunction* of two propositions p and q is the proposition $p \wedge q$ ("p and q").

- The *disjunction* of p and q is the proposition $p \vee q$ ("p or q").

- The truth values of $p \wedge q$ and $p \vee q$ depend on the truth values of p and q, as shown by the following *truth table:*

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

- The *negation* of p is the proposition written as $\neg p$ (" not p"):

p	$\neg p$
T	F
F	T

Logical Connectives (Continued)

- Another logical connective is the *conditional.* The proposition $p \rightarrow q$ means "if p, then q."

- The truth table for conditional is not as obvious:

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

- The statements "if p then q" and "q if p" are equivalent.

- Another way to read $p \rightarrow q$ is "p only if q."

- The statements "p if q" ($q \rightarrow p$) and "p only if q" ($p \rightarrow q$) are not the same. Each is said to be the *converse* of the other.

- The proposition $(p \rightarrow q) \wedge (q \rightarrow p)$ is abbreviated $p \leftrightarrow q$, and the connective $\leftrightarrow$ is called the *biconditional.* $p \leftrightarrow q$ is read "p if and only if q."

Tautologies and Contradictions

- A compound proposition is composed of *propositional variables* such as p and q and logical connectives.

- The following truth table shows how to determine the truth value of the compound proposition $(p \vee q) \wedge \neg(p \rightarrow q)$ from the truth values of p and q:

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

- A compound proposition is called a *tautology* if it is true in every case. The proposition $p \vee \neg\, p$ is a tautology.

- A *contradiction* is a proposition that is false in every case, such as $p \wedge \neg\, p$.

1.2.2 Logical Implication and Equivalence

- Suppose that P and Q are compound propositions. If Q is true in each case that P is true, then P *logically implies* Q, which is written $P \Rightarrow Q$.

- If P and Q have the same truth value in every case, then P and Q are *logically equivalent*, written $P \Leftrightarrow Q$.

- Clearly, $P \Leftrightarrow Q$ is the same as $P \Rightarrow Q$ and $Q \Rightarrow P$.

- The $\rightarrow$ logical connective is related to the $\Rightarrow$ relation in the following way: $P \Rightarrow Q$ means that $P \rightarrow Q$ is a tautology.

- Similarly, $P \Leftrightarrow Q$ means that $P \leftrightarrow Q$ is a tautology.

- The set identities discussed earlier can be converted into logical equivalences, with $\vee$ and $\wedge$ replacing $\cup$ and $\cap$, respectively. The proposition *false* corresponds to the empty set $\varnothing$, and the proposition *true* corresponds to the universe U.

- Other useful logical equivalences:

$$(p \rightarrow q) \Leftrightarrow (\neg\, p \vee q)$$
$$(p \rightarrow q) \Leftrightarrow (\neg\, q \rightarrow \neg\, p)$$
$$\neg\,(p \rightarrow q) \Leftrightarrow (p \wedge \neg\, q)$$

  The second equivalence states that the conditional proposition $p \rightarrow q$ is logically equivalent to its *contrapositive*, $\neg\, q \rightarrow \neg\, p$.

1.2.3 Logical Quantifiers and Quantified Statements

- The statement “there exists an x such that $x^2 < 4$” is said to be a *quantified* statement.

  The phrase “there exists” is called the *existential quantifier.*
  The variable x is said to be a *bound* variable.

  This statement can be written more compactly as $\exists x(x^2 < 4)$.

- The other quantifier is the *universal quantifier,* written $\forall$.

  $\forall x(x^2 < 4)$ means “for every x, $x^2 < 4$.”

- The truth or falsity of a quantified statement depends on the domain of each bound variable.

- A proposition may contain more than one quantifier:

  $\forall x(\exists y((x - y)^2 < 4))$
  $\exists y(\forall x((x - y)^2 < 4))$

  The order of quantifiers is significant. If the domain of x and y is the set of real numbers, then the first proposition is true and the second is false.

Logical Quantifiers and Quantified Statements (Continued)

- **Example 1.1:** Alternative Notation for Quantified Statements

  Sometimes it is important to specify the domain of a particular bound variable. The domain of an *existentially* bound variable can be specified by writing

  $\exists x \in A(P(x))$

  which means

  $\exists x(x \in A \wedge P(x))$

  The domain of a *universally* bound variable can be specified by writing

  $\forall x \in A(P(x))$

  which means

  $\forall x(x \in A \to P(x))$

  Sometimes the quantifier notation is relaxed even further. The proposition

  $\forall x > 0(P(x))$

  is an abbreviation for

  $\forall x(x > 0 \to P(x))$

- **Example 1.2:** A Quantified Statement Saying p is Prime

  The statement "p is prime" can be divided into two parts: "p is greater than 1" and "p is divisible only by 1 and itself," where the domain of p is the set of natural numbers.

  Formally, "p is prime" can be written as the conjunction of two propositions, the second of which uses a universal quantifier and an existential quantifier:

  $(p > 1) \wedge \forall k(\exists m(p = m * k) \to (k = 1 \vee k = p))$

Logical Quantifiers and Quantified Statements (Continued)

- The following rules can be used to simplify a quantified statement that is preceded by the negation symbol:

  $\neg\forall x(P(x))$ is the same as $\exists x(\neg P(x))$
  $\neg\exists x(P(x))$ is the same as $\forall x(\neg P(x))$

- These rules can be applied more than once if necessary:

$$\begin{aligned}\neg\forall x(\exists y(\forall z(P(x, y, z)))) &= \exists x(\neg\exists y(\forall z(P(x, y, z)))) \\ &= \exists x(\forall y(\neg\forall z(P(x, y, z)))) \\ &= \exists x(\forall y(\exists z(\neg P(x, y, z))))\end{aligned}$$

1.3 Functions

- A function assigns to each element of one set a single element of another set.

  The first set is the *domain* of the function.
  The second set is the *codomain*.

- The notation

  $f : A \rightarrow B$

  indicates that f is a function with domain A and codomain B.

- Examples of functions (H is the set of human beings):

  $f_1 : \mathcal{N} \rightarrow \mathcal{N}$, defined by the rule $f_1(x) = x^2$
  $f_2 : H \rightarrow H$, defined by the rule $f_2(x)$ = the mother of x
  $f_3 : H \rightarrow \mathcal{N}$, defined by the rule $f_3(x)$ = the number of siblings of x
  $f_4 : 2^H - \{\varnothing\} \rightarrow H$, defined by the rule $f_4(x)$ = the tallest person in the set x

  The last example assumes that no two humans are exactly the same height.

1.3.1 One-to-one and Onto Functions

- Suppose $f : A \to B$ and $S \subseteq A$. The effect of applying f to S is defined as follows:

  $f(S) = \{f(x) \mid x \in S\} = \{y \in B \mid y = f(x)$ for at least one $x \in S\}$

- The set $f(A)$ is called the *range* of f.

- If $f(A) = B$, f is said to be *onto*, or *surjective*, or a *surjection*.

- f is *one-to-one*, or *injective,* or an *injection,* if no single element y of B can be $f(x)$ for more than one x in A.

  In other words, f is one-to-one if, whenever $f(x_1) = f(x_2)$, then $x_1 = x_2$.

- A *bijection* is a function that is both one-to-one and onto.

- To a large extent, whether a function is one-to-one or onto depends on the domain and codomain, as the following examples show. ($\mathcal{R}$ is the set of all real numbers and $\mathcal{R}^+$ is the set of all nonnegative real numbers.)

  $f : \mathcal{R} \to \mathcal{R}$, defined by $f(x) = x^2$, is neither one-to-one nor onto.
  $f : \mathcal{R} \to \mathcal{R}^+$, defined by $f(x) = x^2$, is onto but not one-to-one.
  $f : \mathcal{R}^+ \to \mathcal{R}$, defined by $f(x) = x^2$, is one-to-one but not onto.
  $f : \mathcal{R}^+ \to \mathcal{R}^+$, defined by $f(x) = x^2$, is both one-to-one and onto.

1.3.2 Compositions and Inverses of Functions

- Suppose that $f : A \rightarrow B$ and $g : B \rightarrow C$ are functions. The function $h : A \rightarrow C$ defined by $h(x) = g(f(x))$ is called the *composition* of g and f and is written $h = g \circ f$.

- Properties of composition:

  Composition is associative: $h \circ (g \circ f) = (h \circ g) \circ f$.
  If f and g are both one-to-one, then $g \circ f$ is also one-to-one.
  If f and g are both onto, then $g \circ f$ is also onto.
  If f and g are both bijections, then $g \circ f$ is also a bijection.

- If $f : A \rightarrow B$ is a bijection, then for any $y \in B$, it makes sense to speak of the element $x \in A$ for which $f(x) = y$, and we denote this x by $f^{-1}(y)$.

- The inverse function $f^{-1} : B \rightarrow A$ has the following properties:

  For every $x \in A$, $f^{-1}(f(x)) = x$.
  For every $y \in B$, $f(f^{-1}(y)) = y$.

- The f^{-1} notation is sometimes used with any function, regardless of whether it is one-to-one or onto:

  If $f : A \rightarrow B$ and S is any subset of B, we write $f^{-1}(S) = \{x \in A \mid f(x) \in S\}$

1.3.3 Operations on a Set

- A function of two or more variables can be viewed as one whose domain is a Cartesian product.

  For example, the function of x and y given by the formula $3x - xy$ could be considered to have domain $\mathcal{R} \times \mathcal{R}$ and codomain $\mathcal{R}$.

- A *binary* operation on a set S is a function from $S \times S$ to S.

- Where a binary operation a on a set is involved, it is common to use the "infix" notation $x \bullet y$ rather than the usual functional notation $\bullet(x, y)$.

  For example, we write $A \cup B$ instead of $\cup(A, B)$, and $x + y$ instead of $+(x, y)$.

- A *unary* operation on S is a function from S to S.

- If $\bullet$ is an arbitrary binary operation on a set S, and T is a subset of S, T is said to be *closed* under the operation $\bullet$ if $T \bullet T \subseteq T$.

- If u is a unary operation on S, and $T \subseteq S$, T is closed under u if $u(T) \subseteq T$.

- An n-ary operation on a set S is a function from the n-fold Cartesian product $S \times S \times \ldots \times S$ to S.

1.4 Relations

- A mathematical relation is a way of making more precise the intuitive idea of a relationship between objects.

- A function is a particular type of relation. The function $f : A \rightarrow B$ may be defined to be a subset f of $A \times B$ so that for each $a \in A$, there is exactly one element $b \in B$ for which $(a, b) \in f$.

- In general, a *relation* from A to B is simply a subset of $A \times B$.

- For an element $a \in A$, a corresponds to, or is *related* to, an element $b \in B$ if the pair (a, b) is in the subset.

- In many cases, A and B are the same set. In that case, the relation is said to be a *relation on A.*

- **Definition 1.1:** A *relation* on a set A is a subset of $A \times A$.

- If R is a relation on a set A, it is common practice to write aRb instead of $(a, b) \in R$.

Examples of Relations

- Examples of relations on $\mathcal{N}$:

  The = relation is defined by the set $\{(0, 0), (1, 1), (2, 2), \ldots\}$.
  The < relation is defined by the set $\{(0, 1), (0, 2), (1, 2), (0, 3), (1, 3), (2, 3), \ldots\}$.

- "Congruence mod n" is a less common relation on $\mathcal{N}$. If n is a fixed positive integer, a is congruent to b mod n, written $a \equiv_n b$, if $a - b$ is an integer multiple of n.

- The relation $\equiv_3$ contains the ordered pairs (0, 0), (1, 1), (1, 4), (4, 1), (7, 10), (3, 6), (8, 14), (76, 4), and every other pair (a, b) for which $a - b$ is divisible by 3.

- A relation on A can be an abitrary subset of $A \times A$, with no simple defining rule.

- Examples of relations on $A = \{1, 2, 3, 4\}$:

  $R_1 = \{(1, 3), (3, 1), (2, 4), (4, 2), (1, 4), (4, 1)\}$
  (for any a and b in A, aR_1b if and only if $|a - b|$ is prime)
  $R_2 = \{(1, 1), (1, 4), (3, 4), (4, 2)\}$

1.4.1 Equivalence Relations and Equivalence Classes

- **Definition 1.2:** Assume that R is a relation on a set A.

  1. R is *reflexive* if for every $a \in A$, aRa.
  2. R is *symmetric* if for every a and b in A, if aRb, then bRa.
  3. R is *transitive* if for every a, b, and c in A, if aRb and bRc, then aRc.
  4. R is an *equivalence relation* on A if R is reflexive, symmetric, and transitive.

- Examples:

  The relations $\leq$ and $\geq$ on the set $\mathcal{N}$ are reflexive.
  The $<$ relation and the $>$ relation are neither reflexive nor symmetric.
  The $\neq$ relation is neither reflexive nor transitive.

- The $=$ relation (on any set) is an equivalence relation.

- Another example of an equivalence relation is the $\equiv_n$ relation on $\mathcal{N}$.

  Reflexive: For every $a \in \mathcal{N}$, $a - a = 0 * n$.

  Symmetric: For every a and b, if $a \equiv_n b$, then for some k, $a - b = k * n$; it follows that $b - a = -k * n = (-k) * n$, and therefore $b \equiv_n a$.

  Transitive: If $a \equiv_n b$ and $b \equiv_n c$, then for some integers k and m, $a - b = kn$ and $b - c = mn$; therefore $a - c = (a - b) + (b - c) = (k + m)n$, and $a - c$ is a multiple of n.

Partitions

- A *partition* of a set A is a collection of pairwise disjoint subsets of A whose union is A.

- Example: The $\equiv_4$ relation partitions the elements of $\mathcal{N}$ into four subsets:

  {0, 4, 8, 12, …}
  {1, 5, 9, 13, …}
  {2, 6, 10, 14, …}
  {3, 7, 11, 15, …}

- If C is a partition of A (the sets in C are pairwise disjoint and $\cup_{S \in C} S = A$), then it defines an equivalence relation E on A as follows:

  aEb if and only if a and b belong to the same element of C

- Similarly, an equivalence relation defines a partition. Suppose R is an equivalence relation on A. For any element a of A, the notation $[a]_R$ (or simply $[a]$) denotes the equivalence class containing a:

  $[a]_R = \{x \in A \mid xRa\}$

- **Theorem 1.1.** For any partition C of a set A, the relation R on A defined by

  xRy if and only if x and y belong to the same element of C

  is an equivalence relation on A. Conversely, if R is any equivalence relation on A, the set of equivalence classes is a partition of A, and two elements of A are equivalent if and if they are in the same equivalence class.

Partitions (Continued)

- To show that a nonempty set S is an equivalence class, we must show two things:

  1. For any x and y in S, x and y are equivalent.
  2. For any $x \in S$ and any $y \notin S$, x and y are not equivalent.

- These two statements can be used to show that the following sets are the equivalence classes of the $\equiv_n$ relation:

$$
\begin{aligned}
[0] &= \{0, n, 2n, 3n, \ldots\} \\
[1] &= \{1, n + 1, 2n + 1, \ldots\} \\
[2] &= \{2, n + 2, 2n + 2, \ldots\} \\
&\ldots \\
[n - 1] &= \{n - 1, 2n - 1, 3n - 1, \ldots\}
\end{aligned}
$$

1.5 Languages

- A *language* is a set of strings involving symbols from some alphabet.

- Examples:

  The set of all valid English sentences.
  The set of all valid C programs.

- An *alphabet* for a particular language is the set of all the legal symbols that can be used to form strings in the language.

- Alphabets are always finite.

- A string over an alphabet Σ is obtained by placing some of the elements of Σ (possibly none) in order.

  Some strings over $\{a, b\}$: *a*, *baa*, *aba*, *aabba*

- The length of a string x is the number of symbols in x, written $|x|$.

- The *null string* (the string of length 0) is a string over any alphabet. The null string is denoted by Λ.

Languages (Continued)

- For any alphabet Σ, the set of all strings over Σ is denoted by Σ^*.

- Every language over Σ is a subset of Σ^*.

- Suppose that $\Sigma = \{a, b\}$. Then

  $\Sigma^* = \{a, b\}^* = \{\Lambda, a, b, aa, ab, ba, bb, aaa, aab, aba, abb, baa, \ldots\}$

- Examples of languages over Σ:

  $\{\Lambda, a, aa, aab\}$
  $\{x \in \{a, b\}^* \mid |x| \leq 8\}$
  $\{x \in \{a, b\}^* \mid |x| \text{ is odd}\}$
  $\{x \in \{a, b\}^* \mid n_a(x) \geq n_b(x)\}$
  $\{x \in \{a, b\}^* \mid |x| \geq 2 \text{ and } x \text{ begins and ends with } b\}$

  The notation $n_a(x)$ means the number of a's in the string x.

Operations on Languages

- Because languages are sets of strings, new languages can be constructed using set operations.

- For any two languages over an alphabet Σ, their union, intersection, and difference are also languages over Σ.

- The complement of a language over Σ is defined as follows:

  $L' = \Sigma^* - L$

- The *concatenation* of strings x and y is the string xy formed by writing the symbols of x and the symbols of y consecutively.

  For any string x, $x\Lambda = \Lambda x = x$.
  Concatenation is associative: for any strings x, y, and z, $(xy)z = x(yz)$.

- A string x is a *substring* of another string y if there are strings w and z, either or both of which may be null, so that $y = wxz$.

  A *prefix* of a string is an initial substring.
  A *suffix* of a string is a final substring.

Operations on Languages (Continued)

- The concatenation operation can be applied to languages as well as to strings:

  $L_1L_2 = \{xy \mid x \in L_1 \text{ and } y \in L_2\}$

- Concatenating any language L with $\{\Lambda\}$ produces L:

  $L\{\Lambda\} = \{\Lambda\}L = L$

- Exponential notation can be used to indicate the number of items (symbols, strings, or languages) being concatenated:

  $a^k = aa \cdot\cdot\cdot a$
  $x^k = xx \cdot\cdot\cdot x$
  $\Sigma^k = \Sigma\Sigma \cdot\cdot\cdot \Sigma = \{x \in \Sigma^* \mid |x| = k\}$
  $L^k = LL \cdot\cdot\cdot L$

- An important special case is the one in which $k = 0$:

  $a^0 = \Lambda \quad x^0 = \Lambda \quad \Sigma^0 = \{\Lambda\} \quad L^0 = \{\Lambda\}$

- Applying the Kleene star operation to a language L yields the set of all strings that can be obtained by concatenating any number of elements of L:

  $$L^* = \bigcup_{i=0}^{\infty} L^i$$

  Note that Λ is always an element of L^*, since $L^0 = \{\Lambda\}$.

- L^+ is the set of all strings obtainable by concatenating one or more elements of L:

  $$L^+ = \bigcup_{i=1}^{\infty} L^i$$

  Note that $L^+ = L^*L = LL^*$.

Defining Languages

- There are two primary ways to define an infinite language: by specifying how to *generate* strings in the language or by specifying how to *recognize* strings in the language.

- A language description based on generation:

  $L_1 = \{ab, bab\}^* \cup \{b\}\{bb\}^*$

- A language description based on recognition:

  $L_2 = \{x \in \{a, b\}^* \mid n_a(x) \geq n_b(x)\}$

- Some language descriptions fall into both categories:

  $L_3 = \{byb \mid y \in \{a, b\}^*\}$

- This course deals with different ways of generating languages as well as various ways of recogizing languages.

- An algorithm for recognizing a language can be thought of as an abstract machine, or automaton.

- Some languages are more complicated than others, requiring the use of more powerful generation techniques or more sophisticated automata.