CS 341: Foundations of CS II

Marvin K. Nakayama
Computer Science Department
New Jersey Institute of Technology
Newark, NJ 07102

Chapter 0
Mathematical Background

Contents

Overview of Course

- Automata Theory:
  - What is a computer?
- Computability Theory
  - What can and cannot be computed?
- Complexity Theory
  - What can and cannot be computed efficiently?

Automata Theory

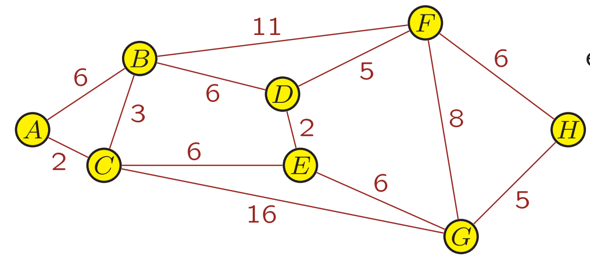
- Finite automata and regular expressions
  - String matching (`grep` in Unix)
  - Circuit design
  - Communication protocols
- Context-free grammars and pushdown automata
  - Compilers
  - Programming languages
- Turing machines
  - Computers
  - Algorithms
- Why study different models of computation?

Computability Theory

- There are algorithms to solve many problems.
- But there are some problems for which there is no algorithm.
- These are called **undecidable** problems:
  - Does a program run forever?
  - Is a program correct?
  - Are two programs equivalent?

Complexity Theory

- For a solvable problem, is there an **efficient** algorithm to solve it?
- Some problems can be solved efficiently:
  - Is there a path from A to H with total cost **at most** 20?



edge labels are costs

- Some problems have no known efficient algorithm:
  - Is there a path from A to H with total cost **at least** 50?

Alphabets, Strings, and Languages

Definition: A **set** is an unordered collection of **objects** or **elements**.

- Sets are written with curly braces $\{\}$.
- The elements in the set are written within the curly braces.

Definition:

- For any set S, "$x \in S$" denotes that x is an element of the set S.
- Also, "$y \notin S$" denotes that y is not an element of the set S.

Remark: We often specify a set using set notation, e.g.,

$$\{ x \mid x \in \mathcal{R},\ x^2 - 4 = 0 \}$$

- $\mathcal{R}$ denotes the set of real numbers.
- "$|$" means "such that"
- Comma means "and"

Sets

Examples:

- The set $\{a, b, c\}$ has elements a, b, and c.
- The sets $\{a, b, c\}$ and $\{b, c, b, a, a\}$ are the same.
  - Order and redundancy do not matter in a set.
- The set $\{a\}$ has element a.
  - $\{a\}$ and a are different things.
  - $\{a\}$ is a set with one element a.
- The set $\mathcal{Z}$ of **integers** is

$$\{\ldots, -2, -1, 0, 1, 2, \ldots\}.$$

- The set $\mathcal{Z}_+$ of **nonnegative integers** is

$$\{0, 1, 2, 3, \ldots\}.$$

Sets

Examples:

- The set of **even numbers** is
$$\{0, 2, 4, 6, 8, 10, 12, \ldots\},$$
which we can also write as $\{\, 2n \mid n = 0, 1, 2, \ldots \,\}$.
  - In particular, 0 is an even number.
- The set of **positive even numbers** is
$$\{2, 4, 6, 8, 10, 12, \ldots\}$$
- The set of **odd numbers**
$$\{1, 3, 5, 7, 9, 11, 13, \ldots\}$$
can also be written as $\{\, 2n + 1 \mid n = 0, 1, 2, \ldots \,\}$.

Example: If A is the set $\{\, 2n \mid n = 0, 1, 2, \ldots \,\}$, then $4 \in A$, but $5 \notin A$.

Alphabets

An **alphabet** is a *finite* set of fundamental units (called **letters** or **symbols**).

Remark: We typically denote an alphabet by a capital Greek letter

- e.g., Σ or Γ (i.e., Sigma or Gamma)

Examples:

- The alphabet of **lower-case Roman letters** is
$$\Sigma = \{\, \mathrm{a}, \mathrm{b}, \mathrm{c}, \ldots, \mathrm{z} \,\}.$$
There are 26 lower-case Roman letters.
- The alphabet of **upper-case Roman letters** is
$$\Gamma = \{\, \mathrm{A}, \mathrm{B}, \mathrm{C}, \ldots, \mathrm{Z} \,\}.$$
There are 26 upper-case Roman letters.

Alphabets

- The alphabet of **Arabic numerals** is
$$\Sigma = \{\, 0, 1, 2, \ldots, 9 \,\}.$$
There are 10 Arabic numerals.

- In this class we will often use the alphabets
$$\Sigma = \{a, b\},$$
$$\Sigma = \{0, 1\}.$$

Sequences and Strings

Definition: **Sequence** of objects is a list of these objects in some order.

- Order and redundancy matter in a sequence, unlike in a set.
- a, b, c and b, c, b, a, a are different sequences.
- $\{a, b, c\}$ and $\{b, c, b, a, a\}$ are the same set.

Definition: A **string over an alphabet** is a **finite** sequence of symbols from the alphabet (written without commas or spaces between the symbols).

Examples:

- x, $cromulent$, $embiggen$, and $kwyjibo$ are strings over the alphabet
$$\Sigma = \{a, b, c, \ldots, z\}.$$

- 0131 is a string over the alphabet $\Sigma = \{0, 1, 2, \ldots, 9\}$.

String Length

Definition: The **length** of a string w is the number of symbols in w.

- Sometimes denote length of w by length(w) or $|w|$.

Example: length$(mom) = |mom| = 3$.

Definition: The **empty string** or **null string**, denoted by ε (i.e., epsilon), is the string consisting of no symbols, i.e.,

$$|\varepsilon| = 0.$$

Kleene Star

Definition: For a given alphabet Σ, let Σ^* denote the set of all possible strings (including ε) over Σ.

Example: If $\Sigma = \{a, b\}$, then

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

String Ordering

Definition: A list of strings $w_1, w_2, \ldots$ over an alphabet Σ is in **string order** (also called **shortlex order**) if

1. shorter strings always appear before longer strings, and
2. strings of the same length appear in alphabetical order.

Example: If $\Sigma = \{0, 1\}$, the string ordering of the strings in Σ^* is

$$\varepsilon,\ 0,\ 1,\ 00,\ 01,\ 10,\ 11,\ 000,\ 001,\ 010,\ 011,\ \ldots$$

Remarks:

- Previous editions (before the 3rd) of Sipser's book instead called this **lexicographic order**.
- String ordering is not the same as dictionary ordering. Why?

Concatenation

Definition: The **concatenation** of strings x and y is the string xy.

Examples:

- If $x = cat$ and $y = dog$, then $xy = catdog$ and $yx = dogcat$.
- If $x = \varepsilon$ and $y = ab$, then $xy = ab = yx$.
- If $x = \varepsilon$ and $y = \varepsilon$, then $xy = \varepsilon = yx$; i.e., $\varepsilon\varepsilon = \varepsilon$.

Definition: For string w, we define w^n for $n \geq 0$ inductively as

- $w^0 = \varepsilon$;
- $w^n = w^{n-1}w$ for any $n \geq 1$.

Example: If $w = dog$, then

$$\begin{aligned} w^0 &= \varepsilon, \\ w^1 &= w^0 w \;=\; \varepsilon dog \;=\; dog, \\ w^2 &= w^1 w \;=\; dogdog, \\ w^3 &= w^2 w \;=\; dogdogdog, \\ &\;\;\vdots \end{aligned}$$

Example: Can also apply this to a single symbol

- $a^3 = aaa$
- $a^0 = \varepsilon$.

Substring

Definition: A **substring** of a string w is any contiguous part of w.

- i.e., y is a substring of w if there exist strings x and z (either or both possibly empty) such that $w = xyz$.

Examples:

- $y = 47$ is a substring of $w = 472$ since letting $x = \varepsilon$ and $z = 2$ gives $w = xyz$.
- The string 472 has substrings ε, 4, 7, 2, 47, 72, and 472.
- 42 is not a substring of 472.

Languages

Definition: A **(formal) language** is a set of strings over an alphabet.

- Language typically denoted by capital Roman letter, e.g., A, B, or L.

Examples:

- Computer languages, e.g., C, C++, or Java, are languages with alphabet

$$\begin{aligned} \Sigma = \{\, & \texttt{a}, \texttt{b}, \ldots, \texttt{z}, \texttt{A}, \texttt{B}, \ldots, \texttt{Z}, \;\;, \texttt{0}, \texttt{1}, \texttt{2}, \ldots, \texttt{9}, \texttt{\_}, \\ & \texttt{>}, \texttt{<}, \texttt{=}, \texttt{+}, \texttt{-}, \texttt{*}, \texttt{/}, \texttt{(}, \texttt{)}, \texttt{.}, \texttt{,}, \texttt{\&}, \texttt{!}, \texttt{\%}, \texttt{|}, \texttt{'}, \texttt{"}, \\ & \texttt{:}, \texttt{;}, \texttt{\^{}}, \texttt{\{}, \texttt{\}}, \texttt{@}, \texttt{\#}, \backslash, \texttt{?}, \texttt{\$}, \texttt{\~{}}, \texttt{`}, \langle \mathsf{CR} \rangle, \langle \mathsf{FF} \rangle \,\}. \end{aligned}$$

  The rules of syntax define the rules for the language.
- The set of valid variable names in C++ is a language. What are the alphabet and rules defining valid variable names in C++?

Examples of Languages

Example: Alphabet $\Sigma = \{a\}$.
Language

$$\begin{aligned} L_0 &= \{\, \varepsilon, a, aa, aaa, aaaa, \ldots \} \\ &= \{\, a^n \mid n = 0, 1, 2, 3 \ldots \} \end{aligned}$$

Note that

- $a^0 = \varepsilon$, so $\varepsilon \in L_0$.
- there are different ways we can specify a language.

Another language

$$L_1 = \{\, a^n \mid n \geq 1 \,\}$$

has $\varepsilon \notin L_1$.

Examples of Languages

Example: Alphabet $\Sigma = \{a\}$.
Language

$$\begin{aligned} L_2 &= \{\, a,\, aaa,\, aaaaa,\, aaaaaaa,\, \ldots \} \\ &= \{\, a^{2n+1} \mid n = 0, 1, 2, 3, \ldots \} \end{aligned}$$

Example: Alphabet $\Sigma = \{0, 1, 2, \ldots, 9\}$.
Language

$$\begin{aligned} L_3 &= \{ \text{ any string of symbols that does not start with symbol "0"} \} \\ &= \{\varepsilon,\, 1,\, 2,\, 3,\, \ldots,\, 9,\, 10,\, 11, \ldots \} \end{aligned}$$

Examples of Languages

Example: Let $\Sigma = \{a, b\}$, and we can define a language L consisting of all strings that begin with a followed by zero or more b's; i.e.,

$$\begin{aligned} L &= \{\, a,\, ab,\, abb,\, abbb,\, \ldots \} \\ &= \{\, ab^n \mid n = 0, 1, 2, \ldots \}. \end{aligned}$$

Is L the language of strings beginning with a?

Definition: The set $\emptyset$, which is called the **empty set**, is the set consisting of no elements.

Remarks:

- $\varepsilon \notin \emptyset$ since $\emptyset$ has no elements.
- $\emptyset \neq \{\, \varepsilon \,\}$ since $\emptyset$ has no elements.

Set Relations and Operations

Definition: If S and T are sets, then $S \subseteq T$ (S is a **subset** of T) if $x \in S$ implies that $x \in T$.

- Each element of S is also an element of T.

Examples:

- Suppose $S = \{\, ab,\, ba \,\}$ and $T = \{\, ab,\, ba,\, aaa \,\}$.
  - Then $S \subseteq T$.
  - But $T \not\subseteq S$.
- Suppose $S = \{\, ba,\, ab \,\}$ and $T = \{\, aa,\, ba \,\}$.
  - Then $S \not\subseteq T$ and $T \not\subseteq S$.

Equal Sets

Definition: Two sets S and T are **equal**, written $S = T$, if

$$S \subseteq T \text{ and } T \subseteq S.$$

Examples:

- Suppose $S = \{\, ab,\, ba \,\}$ and $T = \{\, ba,\, ab \,\}$.
  - Then $S \subseteq T$ and $T \subseteq S$.
  - So $S = T$.
- Suppose $S = \{\, ab,\, ba \,\}$ and $T = \{\, ba,\, ab,\, aaa \,\}$.
  - Then $S \subseteq T$, but $T \not\subseteq S$.
  - So $S \neq T$.

Union

Definition: The **union** of two sets S and T is

$$S \cup T = \{ x \mid x \in S \text{ or } x \in T \}$$

- $S \cup T$ consists of all elements in S or in T (or in both).

Examples:

- If $S = \{ ab, bb \}$ and $T = \{ aa, bb, a \}$,
  - then $S \cup T = \{ ab, bb, aa, a \}$.
- If $S = \{ a, ba \}$ and $T = \emptyset$,
  - then $S \cup T = S$.
- If $S = \{ a, ba \}$ and $T = \{ \varepsilon \}$,
  - then $S \cup T = \{ \varepsilon, a, ba \}$.

Intersection

Definition: The **intersection** of two sets S and T is

$$S \cap T = \{ x \mid x \in S \text{ and } x \in T \},$$

- $S \cap T$ consists of elements that are in both S and T.

Definition: Sets S and T are **disjoint** if $S \cap T = \emptyset$.

Examples:

- Suppose $S = \{ ab, bb \}$ and $T = \{ aa, bb, a \}$.
  - Then $S \cap T = \{ bb \}$.
- Suppose $S = \{ ab, bb \}$ and $T = \{ aa, ba, a \}$.
  - Then $S \cap T = \emptyset$, so S and T are disjoint.

Set Subtraction

Definition: The **difference** of two sets S and T is

$$S - T = \{ x \mid x \in S, x \notin T \}.$$

Examples:

- Suppose $S = \{ a, b, bb, bbb \}$ and $T = \{ a, bb, bab \}$.
  - Then $S - T = \{ b, bbb \}$.
  - What is $T - S$?
- Suppose $S = \{ ab, ba \}$ and $T = \{ ab, ba \}$.
  - Then $S - T = \emptyset$.

Complement

Definition: The **complement** of a set S is

$$\overline{S} = \{ x \mid x \notin S \}.$$

- $\overline{S}$ is the set of all elements under consideration that are *not* in S.

Example:

- Let S be set of strings over alphabet $\Sigma = \{a, b\}$ that begin with symbol b.
- Then $\overline{S}$ is set of strings over Σ that do not begin with symbol b, i.e.,
$$\overline{S} = \Sigma^* - S.$$
- $\overline{S}$ is **not** the set of strings over Σ that begin with the symbol a
  - $\varepsilon \in \overline{S}$ and ε does not begin with a.

Concatenation

Definition: The **concatenation** (or **product**) of sets S and T is

$$S \circ T = \{\, xy \mid x \in S,\ y \in T \,\}.$$

Remarks:

- $S \circ T$ is the set of strings that can be split into 2 parts
  - first part of string is in S, and
  - second part is in T.
- Sometimes write ST rather than $S \circ T$ to denote concatenation.

Concatenation

Recall

$$S \circ T = \{\, xy \mid x \in S,\ y \in T \,\}.$$

Examples:

- If $S = \{\, a,\ aa \,\}$ and $T = \{\, \varepsilon,\ a,\ ba \,\}$, then
$$\begin{aligned} S \circ T &= \{\, a,\ aa,\ aba,\ aaa,\ aaba \,\}, \\ T \circ S &= \{\, a,\ aa,\ aaa,\ baa,\ baaa \,\}. \end{aligned}$$
  - $aba \in S \circ T$, but $aba \notin T \circ S$.
  - Thus, $S \circ T \neq T \circ S$.
- If $S = \{\, ab,\ ba \,\}$ and $T = \emptyset$, then
$$S \circ T = T \circ S = \emptyset.$$

Cardinality

Definition: The **cardinality** $|S|$ of a set S is number of elements in S.

Definition:

- A set S is **finite** if $|S| < \infty$.
- If S is not finite, then S is **infinite**.

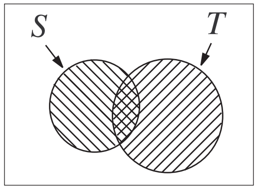
Examples:

- Suppose $S = \{\, \varepsilon,\ bba \,\}$ and $T = \{\, a^n \mid n \geq 1 \,\}$.
  - Then $|S| = 2$ and $|T| = \infty$.
- If $S = \emptyset$, then $|S| = 0$.

Cardinality of Union

Fact: If S and T are any 2 sets such that $|S \cap T| < \infty$, then

$$|S \cup T| = |S| + |T| - |S \cap T|.$$



In particular, if $S \cap T = \emptyset$, then $|S \cup T| = |S| + |T|$.

Sequences and Tuples

Definition: **Sequence** of objects is a list of these objects in some order.

- Sometimes sequences are written within parentheses.

Example: The sequence $7, 2, 7, 8$ may be written as $(7, 2, 7, 8)$.

Example: The sequence $(7, 2, 7, 8) \neq (2, 8, 7)$

- Order and redundancy matter in a sequence (but they don't in a set).

Definition: Finite sequences are called **tuples**.

- A k-**tuple** has k elements in the sequence.

Examples:

- $(43, 2, 7871)$ is a 3-tuple, which is also called a **triple**.
- $(9, 23)$ is a 2-tuple, which is also called a **pair**.

Cartesian Product

Definition: The **Cartesian product** (or **cross product**) of two sets S and T is the set of pairs

$$S \times T \;=\; \{\,(x, y) \mid x \in S,\, y \in T\,\}.$$

Examples: Suppose $S = \{\, a,\, ba,\, bb \,\}$ and $T = \{\, \varepsilon,\, ba \,\}$.

- $S \times T = \{\ (a, \varepsilon),\ (a, ba),\ (ba, \varepsilon),\ (ba, ba),\ (bb, \varepsilon),\ (bb, ba)\ \}$.
- For example, the pair $(a, ba) \in S \times T$.
- $T \times S = \{\ (\varepsilon, a),\ (\varepsilon, ba),\ (\varepsilon, bb),\ (ba, a),\ (ba, ba),\ (ba, bb)\ \}$.
- $(ba, a) \in T \times S$, but $(ba, a) \notin S \times T$, so $T \times S \neq S \times T$.
- Concatenation is not the same as Cartesian product:

$$S \circ T = \{\ a,\ aba,\ ba,\ baba,\ bb,\ bbba\ \} \neq S \times T.$$

Cartesian Product

Example: For $\mathcal{Z} = \{\ldots, -2, -1, 0, 1, 2, \ldots\}$,

$$\mathcal{Z} \times \mathcal{Z} = \{\,(x, y) \mid x \in \mathcal{Z},\, y \in \mathcal{Z}\,\}$$

Remark: $|S \times T| \;=\; |S| \cdot |T|$. Why?

Remark: Can also define Cartesian product of more than 2 sets.

Definition: The **Cartesian product** (or **cross product**) of k sets $S_1, S_2, \ldots, S_k$ is the set

$$\begin{aligned} & S_1 \times S_2 \times \cdots \times S_k \\ & \quad = \{\,(x_1,\, x_2,\, \ldots,\, x_k) \mid x_i \in S_i \text{ for } i = 1, 2, \ldots, k\,\} \end{aligned}$$

of k-tuples.

Definition: $S^k = \underbrace{S \times S \times \cdots \times S}_{k \text{ times}}$

Cartesian Product

Example:

- Suppose

$$\begin{aligned} S_1 &= \{\, ab,\, ba,\, bbb \,\}, \\ S_2 &= \{\, a,\, bb \,\}, \\ S_3 &= \{\, ab,\, b \,\}. \end{aligned}$$

- Then

$$\begin{aligned} S_1 \times S_2 \times S_3 = \{\ & (ab, a, ab),\ (ab, a, b),\ (ab, bb, ab), \\ & (ab, bb, b),\ (ba, a, ab),\ (ba, a, b), \\ & (ba, bb, ab),\ (ba, bb, b),\ (bbb, a, ab), \\ & (bbb, a, b),\ (bbb, bb, ab),\ (bbb, bb, b)\ \}. \end{aligned}$$

- Note that the 3-tuple $(ab, a, ab) \in S_1 \times S_2 \times S_3$.

Power Set

Definition: The **power set** $\mathcal{P}(S)$ of a set S is

$$\mathcal{P}(S) = \{ A \mid A \subseteq S \}.$$

- $\mathcal{P}(S)$ is the set of all possible **subsets** of S.

Example: If $S = \{ a, bb \}$, then

$$\mathcal{P}(S) = \{ \emptyset, \{a\}, \{bb\}, \{a, bb\} \}.$$

Fact: If $|S| < \infty$, then

$$|\mathcal{P}(S)| = 2^{|S|},$$

i.e., there are $2^{|S|}$ different subsets of S. Why?

Repeated Concatenations of a Set

Definition: Given a set S of strings, we define $S^{(k)}$ for $k \geq 0$ as

$$\begin{aligned} S^{(0)} &= \{ \varepsilon \} \text{ and} \\ S^{(k)} &= S^{(k-1)} \circ S \text{ for } k \geq 1. \end{aligned}$$

Remarks:

- Can show (by induction) that for $k \geq 1$,

$$\begin{aligned} S^{(k)} &= \underbrace{S \circ S \circ \cdots \circ S}_{k \text{ times}} \\ &= \{ w_1 w_2 \cdots w_k \mid w_i \in S, \ \forall\, i = 1, 2, \ldots, k \}. \end{aligned}$$

- $S^{(k)}$ is the set of strings formed by concatenating k strings from S, where we allow repetition.
- Note that $S^{(1)} = S$.

Example

Example: If $S = \{a, bb\}$, then

$$S^{(0)} = \{ \varepsilon \},$$

$$S^{(1)} = \{ a, bb \},$$

$$S^{(2)} = \{ aa, abb, bba, bbbb \},$$

$$S^{(3)} = \{ aaa, aabb, abba, abbbb, bbaa, bbabb, bbbba, bbbbbb \}.$$

Example: If $S = \emptyset$, then

$$\begin{aligned} S^{(0)} &= \{\varepsilon\}, \\ S^{(k)} &= \emptyset, \text{ for all } k \geq 1. \end{aligned}$$

Kleene Star Closure S^*

Definition: The **(Kleene star) closure** of a set of strings S is

$$S^* = \bigcup_{k=0}^{\infty} S^{(k)} = S^{(0)} \cup S^{(1)} \cup S^{(2)} \cup S^{(3)} \cup \cdots.$$

Remarks:

- S^* is the set of all strings formed by concatenating zero or more strings from S, where we may use the same string more than once.
- In set notation,

  $$S^* = \{ w_1 w_2 \cdots w_k \mid k \geq 0 \text{ and } w_i \in S \text{ for all } i = 1, 2, \ldots, k \},$$

  where the concatenation of $k = 0$ strings is the empty string ε.
- $S \subseteq S^*$.

Examples of Kleene Star Closure

Example: If $S = \{ ba,\ a \}$, then

$$S^* = \{ \varepsilon,\ a,\ aa,\ ba,\ aaa,\ aba,\ baa,\ aaaa,\ aaba,\ \ldots \}.$$

If $x \in S^*$, can bb ever be a substring of x?

Example: If $\Sigma = \{ a,\ b \}$, then

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

which is all possible strings over the alphabet Σ.

Example: If $S = \emptyset$, then $S^* = \{ \varepsilon \}$.

Example: If $S = \{ \varepsilon \}$, then $S^* = \{ \varepsilon \}$.

$S^{**} = S^*$

Remark: $S^{**} = (S^*)^*$, so S^{**} is the set of strings formed by concatenating strings from S^*.

Fact: $S^{**} = S^*$ for any set S of strings.

Proof. The way we will prove this is by showing two things:

1. $S^{**} \subseteq S^*$
2. $S^* \subseteq S^{**}$.

To show part 2,

- for any set A, we know that $A \subseteq A^*$.
- Hence, letting $A = S^*$, we see that $S^* \subseteq S^{**}$.

Proof that $S^{**} \subseteq S^*$

To show 1, need to prove that any string $w \in S^{**}$ is also in S^*.

- Since $w \in S^{**}$, can write w as a concatenation of zero or more strings from S^*.
  - $w = w_1 w_2 \cdots w_k$ for some $k \geq 0$, where each $w_i \in S^*$.
- Each string $w_i \in S^*$ can be written as a concatenation of zero or more strings from S.
- Thus, the original string w can be written as a concatenation of zero or more strings from S.
- Since S^* is the collection of all strings that are concatenation of zero or more strings from S, this implies that the original string $w \in S^*$.
- Therefore, $w \in S^{**}$ implies $w \in S^*$, so $S^{**} \subseteq S^*$.

∎

Positive Closure S^+

Definition: If S is a set of strings, then the **positive closure** of S is

$$\begin{aligned} S^+ &= S^{(1)} \cup S^{(2)} \cup S^{(3)} \cup \cdots \\ &= \{ w_1 w_2 \cdots w_k \mid k \geq 1 \text{ and each } w_i \in S \}. \end{aligned}$$

- S^+ is the set of all strings formed by concatenating *one or more* strings from S.

Example: If $\Sigma = \{ a \}$, then

$$\Sigma^+ = \{ a,\ aa,\ aaa,\ \ldots \} \neq \Sigma^*.$$

Example: If $S = \{ a,\ ba \}$, then

$$S^+ = \{ a,\ aa,\ ba,\ aaa,\ aba,\ baa,\ aaaa,\ aaba,\ \ldots \} \neq S^*.$$

Example: If $S = \{ \varepsilon,\ a,\ ba \}$, then

$$S^+ = \{ \varepsilon,\ a,\ aa,\ ba,\ aaa,\ aba,\ baa,\ aaaa,\ aaba,\ \ldots \} = S^*.$$

Functions and Operations

Definition: A **function** (or **operator**, **operation**, or **mapping**) f maps each element in a **domain** D to a *single* element in a **range** R.

- We denote this by $f : D \to R$.

Remarks:

- If f is a function that outputs $b \in R$ when the input is $a \in D$, we write
$$f(a) = b.$$
- We say that the mapping f
  - defined on the domain D
  - R-valued mapping.
- A **real-valued function** has range $R \subseteq \mathcal{R}$, where $\mathcal{R}$ denotes the set of real numbers.

Examples of Functions

Examples:

- We can define a function $f : \mathcal{Z} \to \mathcal{Z}$ as
$$f(x) = x^2 - 5.$$
Note that $f(3) = f(-3) = 4$.
- Integer **addition** has function $g : \mathcal{Z} \times \mathcal{Z} \to \mathcal{Z}$ with
$$g(x, y) = x + y.$$
- If Σ is an alphabet, then we can define $f : \Sigma^* \to \mathcal{Z}_+$ such that for any string $w \in \Sigma^*$,
$$f(w) = |w|,$$
which is the length of w.
- Let Σ be an alphabet. Then we can define **concatenation** as the function $f : \Sigma^* \times \Sigma^* \to \Sigma^*$ with
$$f(x, y) = xy$$

Closed Under an Operation

Let A be some collection of objects.

Definition: We say that A is **closed under operation** f if applying f to members of A always returns a member of A.

Examples:

- $\mathcal{N} = \{1, 2, 3, \ldots\}$ is closed under addition.
- $\mathcal{N}$ is not closed under subtraction since $4, 7 \in \mathcal{N}$, but $4 - 7 = -3 \notin \mathcal{N}$.
- $L_1 = \{\, a^n \mid n = 1, 2, 3, \ldots \}$ is closed under concatenation.
- Is $L_2 = \{\, a^{2n+1} \mid n = 0, 1, 2, \ldots \}$ closed under concatenation?

String Reversal

Definition: For any string w, the **reverse** of w, written as reverse(w) or $w^{\mathcal{R}}$, is the same string of symbols written in reverse order.

- If $w = w_1 w_2 \cdots w_n$, where each w_i is a symbol, then $w^{\mathcal{R}} = w_n w_{n-1} \cdots w_1$.

Examples:

- $(cat)^{\mathcal{R}} = tac$ and $\varepsilon^{\mathcal{R}} = \varepsilon$.
- The set $A = \{\, 0,\ 11,\ 01,\ 10 \,\}$ is closed under reversal since if $w \in A$, then $w^{\mathcal{R}} \in A$.
- Let B be the set of strings over $\Sigma = \{0, 1, 2, \ldots, 9\}$ such that the first symbol is not 0.
  - Note that $10 \in B$, but $(10)^{\mathcal{R}} = 01 \notin B$.
  - Thus, B is not closed under reversal.

Palindrome

Definition: Over the alphabet $\Sigma = \{a, b\}$, the language **PALINDROME** is defined as

$$\begin{aligned}\textbf{PALINDROME} &= \{\, w \in \Sigma^* \mid w = w^{\mathcal{R}} \,\} \\ &= \{\, \varepsilon,\ a,\ b,\ aa,\ bb,\ aaa,\ aba, \ldots \}\end{aligned}$$

Remark:

- Strings $abba, a \in$ **PALINDROME**,
  - but their concatenation $abbaa$ is not in **PALINDROME**.
- Thus, **PALINDROME** is not closed under concatenation.

Defining Functions

Remark: Sometimes we define a function using a table.

Example: Consider function $f : \{0, 1, 2, 3, 4\} \to \{0, 1, 2, 3, 4\}$ as

n	$f(n)$
0	1
1	2
2	3
3	4
4	0

Note that $f(n) = (n + 1)$ mod 5.

- a mod b returns the remainder after dividing a by b.
- **Example:** 5 mod 7 $=$ 5, and 15 mod 7 $=$ 1.

Example of Function

Example: Let $A = \{$ ROCK, PAPER, SCISSORS $\}$ and $B = \{$ TRUE, FALSE $\}$. Consider the function

beats $: A \times A \to B$

defined by the table

beats	ROCK	PAPER	SCISSORS
ROCK	FALSE	FALSE	TRUE
PAPER	TRUE	FALSE	FALSE
SCISSOR	FALSE	TRUE	FALSE

- Then *beats* defines the game Rock-Paper-Scissors.
- For example,

beats(ROCK, SCISSORS) = TRUE,
beats(ROCK, PAPER) = FALSE.

k-ary Functions

Definition: When the domain of a function f is $A_1 \times A_2 \times \cdots \times A_k$ for some sets $A_1, A_2, \ldots, A_k$,

- input to function f is k-tuple $(a_1, a_2, \ldots, a_k) \in A_1 \times A_2 \times \cdots \times A_k$,
- we call each a_i an **argument** to f.

Definition: A function f with k arguments is a k-**ary function**.

- k is called the **arity** of f.

Definition: A **unary** function has arity $k = 1$.

- e.g., $f(x) = 3x + 4$ or $f(w) = |w|$.

Definition: A **binary** function has arity $k = 2$

- e.g., *beats* is a binary function.

Predicates and Relations

Definition: A **predicate** or **property** is a function whose range is $\{\text{TRUE}, \text{FALSE}\}$,

- e.g., *beats* is a property.

Definition: A property whose domain is a set $A \times \cdots \times A$ of k-tuples is called a **relation**, a k-**ary relation**, or a k-**ary relation on** A.

Definition: A 2-ary relation is a **binary relation**,

- e.g., *beats* is a binary relation.

Remark: If R is a binary relation, aRb means $aRb = \text{TRUE}$.

Example: For the binary relation "$<$", we have $2 < 5 \ = \ \text{TRUE}$.

Predicates

Remark:

- Sometimes more convenient to describe predicates with sets instead of functions.
- Sometimes write predicate $P : D \to \{\,\text{TRUE, FALSE}\,\}$ as
  - (D, S), where $S = \{\, a \in D \mid P(a) = \text{TRUE}\,\}$,
  - or just S when domain D is obvious.
- For example, *beats* can be written as

  $$\{\,(\text{ROCK}, \text{SCISSORS}),\ (\text{PAPER}, \text{ROCK}),\ (\text{SCISSORS}, \text{PAPER})\,\}$$

  which is the set $\{\,(x, y) \mid (x, y) \in D$ and xRy (i.e., x beats y) $\}$.

Reflexive, Symmetric and Transitive Relations

Definition: A binary relation R is

- **reflexive** if for every x, xRx;
- **symmetric** if for every x and y, xRy if and only if yRx;
- **transitive** if for every x, y, and z, xRy and yRz implies xRz.

Definition: A binary relation is an **equivalence relation** if it is reflexive, symmetric, and transitive.

Example:

- Let $\mathcal{N} = \{0, 1, 2, \ldots\}$.
- For fixed positive integer k, define relation $\equiv_k$ on $\mathcal{N} \times \mathcal{N}$ as follows:
  - for $a, b \in \mathcal{N}$, $a \equiv_k b$ iff $a - b$ is a multiple of k.
  - i.e., $a \equiv_k b$ iff $(a - b) = rk$, for some $r \in \mathcal{Z}$.
- $\equiv_k$ defines the standard "modulo k" relation.
- Prove that this is an equivalence relation.

$\equiv_k$ is an Equivalence Relation

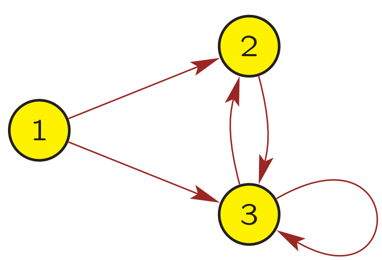
- **Recall**: $a \equiv_k b$ iff $(a - b) = rk$, for some $r \in \mathcal{Z}$.
- **Reflexive**: Show that $x \equiv_k x$.
  - $\forall\, x \in \mathcal{N}$, $x - x = 0 = 0k$.
  - Since $0 \in \mathcal{Z}$, this shows that $x \equiv_k x$.
  - Therefore, $\equiv_k$ is reflexive.
- **Symmetric**: Show that $x \equiv_k y \quad \Rightarrow \quad y \equiv_k x$.
  - Consider $x, y \in \mathcal{N}$ such that $x \equiv_k y$.
  - Therefore $(x - y) = zk$ for some $z \in \mathcal{Z}$ by definition.
  - But this means $(y - x) = -zk$.
  - Since $-z \in \mathcal{Z}$ as well, this shows that $y \equiv_k x$.
  - Therefore, $\equiv_k$ is symmetric.

$\equiv_k$ is an Equivalence Relation

- **Transitive**: Show that $x \equiv_k y$ and $y \equiv_k z$ imply $x \equiv_k z$.
  - Suppose $x \equiv_k y$ and $y \equiv_k z$.
  - Then $(x - y) = ik$ for some $i \in \mathcal{Z}$.
  - Also, $(y - z) = jk$ for some $j \in \mathcal{Z}$.
  - Thus, $(x - y) + (y - z) = ik + jk$.
  - But $(x - y) + (y - z) = (x - z)$ and $ik + jk = (i + j)k$.
  - $(i + j) \in \mathcal{Z}$ since $i \in \mathcal{Z}$ and $j \in \mathcal{Z}$.
  - So $(x - z) = (i + j)k$.
  - Hence, $x \equiv_k z$.
  - Therefore, $\equiv_k$ is transitive.
- Since $\equiv_k$ is reflexive, symmetric, and transitive, it is an equivalence relation.

Graphs

Definition: A **directed graph** is a set of **nodes** (or **vertices**) and directed **edges** (or **arcs**).

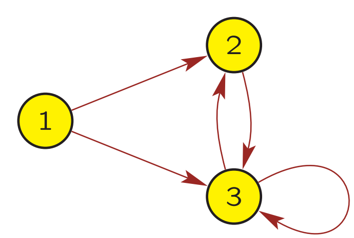


- In a graph G that contains nodes i and j, the pair (i, j) represents a directed edge from node i to node j.
- An **undirected** graph has undirected edges.

Example of Directed Graph

- Graph $G = (V, E)$, where
  - V is the set of nodes of G
  - $E \subseteq V \times V$ is the set of edges.
- For the graph below,

$$\begin{aligned} V &= \{1, 2, 3\}, \\ E &= \{\, (1,2),\ (1,3),\ (2,3),\ (3,2),\ (3,3)\,\}, \\ G &= (\,\{1,2,3\},\ \{\,(1,2),\ (1,3),\ (2,3),\ (3,2),\ (3,3)\,\}\,). \end{aligned}$$



Boolean Logic

- Boolean logic is a mathematical system built around two values: TRUE and FALSE.
- Sometimes TRUE and FALSE are written as $1, 0$.
- You should be familiar with
  - conjunction (AND), denoted by $\wedge$
  - disjunction (OR), denoted by $\vee$
  - negation, denoted by $\neg$ or bar, e.g., $\neg 0$ and $\bar{0}$ are 1
  - exclusive or (XOR), denoted by $\oplus$
  - equality operator ($\leftrightarrow$)
  - implication operator ($\rightarrow$)
  - distributive laws

Some Properties of Boolean Logic

- The implication operator has the following **truth table**:

x	y	$x \rightarrow y$	$(\neg x) \vee y$
0	0	1	1
0	1	1	1
1	0	0	0
1	1	1	1

- This means that an implication $x \rightarrow y$ is always true if x is false.
- The implication operator can be rewritten as "(not x) or y".

Summary of Chapter 0

- A language is a set of strings.
- Kleene-star operation:
  $S^* = \{ w_1 w_2 \cdots w_k \mid k \geq 0 \text{ and each } w_i \in S \}$.
- Set operations and relations: subsets, union, equality, intersection, subtraction, complement, concatenation, cardinality, Cartesian product, power set
- Functions, k-ary functions, predicates, relations
- Set S is closed under a function f if applying f to elements in S always results in something in S.
- Graphs
- Boolean logic