Lecture 2

Math Review cont.

Remark: If the range of a function f is $\{0,1\}$ (or, in general, any binary set), then f is called a predicate / property.

e.g. $\text{even} : \underbrace{\mathbb{Z}}_{\text{domain}} \longrightarrow \underbrace{\{T, F\}}_{\text{range}}$

$\text{even}(1) = F$ ... not even

$\text{even}(4) = T$

$$\text{equals} : \mathbb{Z} \times \mathbb{Z} \longrightarrow \{T, F\}$$

$$\text{equals}(1, 2) = F$$
$$\text{equals}(1, 1) = T$$

Exercise: Define the signature of equality function between three integers.

e.g. $eq(1,1,1) = T$

$eq(1,2,2) = F$

$$<= : \mathbb{Z} \times \mathbb{Z} \longrightarrow \{T, F\}$$

property ... predicate ... relation

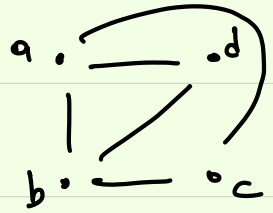
Review: reflexive, symmetric, transitive, antisymmetric

Graphs

A graph G is a pair (V, E) where

- V is the set of nodes
- $E \subseteq V \times V$ is the set of edges.

e.g.

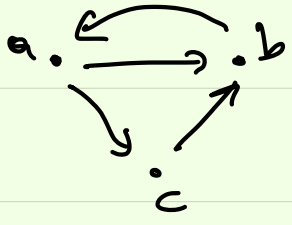


undirected graph

- adjacency matrices
- adjacency lists
- list of edges

- Check: degree of a node, path, circuit/cycle

Directed graphs:



- indegree of a node.
- outdegree of a node
- directed path
- connectedness

Strings (Words) and Formal Languages

Definition: An **alphabet** Σ is a **finite set of symbols**

e.g. $\Sigma = \{\diamondsuit, \square\}$

$\Sigma = \{0, 1\}$

$\Sigma = \{a, b, c, \ldots, x, y, z\}$

$\Sigma = ASCII$

$\Sigma = \{0, 1, 2, \ldots, 9\}$

$\Sigma = \{C, D, E, F, G, A, B\}$

$\Sigma = \{1\}$

Definition: A string (word) over an alphabet Σ is a finite sequence of symbols of Σ.

e.g. $\Sigma = \{0, 1\}$ $\quad w = 010$

$w = 1$

empty string $\quad w = \lambda \ldots (\epsilon)$

* $|w|$ ... length of w

... # of symbols in w

e.g. $w = 010 \Rightarrow |w| = 3$

$w = 1 \Rightarrow |w| = 1$

$w = \lambda \Rightarrow |w| = 0$

$|_| : \text{Word} \longrightarrow \mathbb{N}$

* Given a string $w = w_1 w_2 \cdots w_n$ (i.e $|w| = n$)

the reverse of w is denoted by w^R

and $w^R = w_n w_{n-1} \cdots w_3 w_2 w_1$

e.g. $w = 011 \Rightarrow w^R = 110$

$w = \lambda \Rightarrow w^R = \lambda$

$w = 1 \Rightarrow w^R = 1$

$w = 010 \Rightarrow w^R = 010$

Remark: If $w = w^R$, then w is called a "palindrome".

Exercise: Let Σ be an alphabet and P is the set of all palindromes over Σ. ($\Sigma = \{a, b\}$). Then $|P| = ?$

* Concatenation of strings.

$x = x_1 \cdots x_m$ and $y = y_1 \cdots y_n$

$\Rightarrow$ $x \cdot y = x_1 \cdots x_m y_1 \cdots y_n$

$_\cdot_ : Word \times Word \to Word$

$|x \cdot y| = |x| + |y|$

Remark:

- $w = x \cdot y \cdot z$ "y is a substring of w"
- $w = x \cdot y$ "x is a prefix of w"

e.g. $w = 0011$ $w = 0 \cdot 011$

$w = 00 \cdot 11$

- $w = x \cdot y$ "y is a suffix of w"

* Lexicographic Order

- Shorter strings come before

$\Sigma = \{a, b\}$ and $a < b$

e.g. $x = ab$ $y = a$ $\Rightarrow y <_{lex} x$

- If same length, then use dictionary order

e.g. $x = bab$ $y = bba$ $\Rightarrow x <_{lex} y$

Definition: A language L over an alphabet Σ is a set of strings over Σ.

e.g. $\Sigma = \{0, 1\}$

$L_1 = \{\}$ ... the empty language over Σ.

$L_2 = \{0\}$

$L_3 = \{\lambda\}$

Note: $L_1 \neq L_3$ because $|L_1| = 0$ but $|L_3| = 1$.

Σ^* = the set of all strings over Σ.

$= \{\lambda, 0, 1, 00, 01, 10, 11, 000, \ldots\}$

$L_4 = \{w \in \Sigma^* \mid |w| = 2k, k \in \mathbb{N}\}$

$\Rightarrow L_4 \subseteq \Sigma^*$

Remark: For every language L over Σ, we have $L \subseteq \Sigma^*$

Recursive Definitions

Example: Set of palindroms P over $\Sigma = \{0, 1\}$

Basis: $\lambda, 0, 1 \in P$

Rec. step: If $w \in P$, so are $0w0 \in P$ and $1w1 \in P$

No other rule is needed.

Example: Dyck language D ... set of well-formed parentheses. $\Sigma = \{(,)\}$

e.g. $() \in D$ $((\notin D$

Basis: $\lambda \in D$

Rec. step: If $x, y \in D$, then so are (x) (or (y)) and $x \cdot y$ (or $y \cdot x$) $\in D$.

No other rule is needed.

e.g. $x = ()()$ $y = (())$

$(x) = (()()) \in D$

$x \cdot y = ()() \cdot (()) \in D$.