Mathematical Preliminaries

class 1

Mathematical Preliminaries

- Sets
- Functions
- Relations
- Graphs
- Proof Techniques

SETS

A set is a collection of elements

$$A = \{1, 2, 3\}$$

$$B = \{train, bus, bicycle, airplane\}$$

We write

$$1 \in A$$

$$ship \notin B$$

Set Representations

C = { a, b, c, d, e, f, g, h, i, j, k }

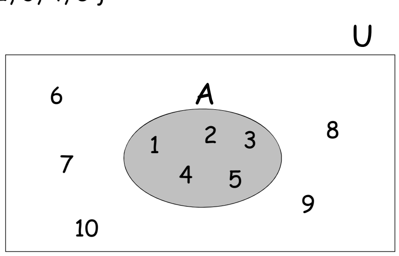
C = { a, b, …, k } ⟶ *finite set*

S = { 2, 4, 6, … } ⟶ *infinite set*

S = { j : j > 0, and j = 2k for some k>0 }

S = { j : j is nonnegative and even }

A = { 1, 2, 3, 4, 5 }



Universal Set: all possible elements

U = { 1 , … , 10 }

Set Operations

A = { 1, 2, 3 } B = { 2, 3, 4, 5}

- Union

  A U B = { 1, 2, 3, 4, 5 }

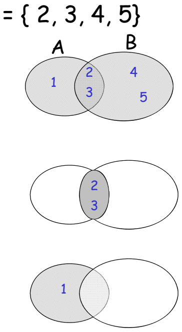
- Intersection

  A ∩ B = { 2, 3 }

- Difference

  A - B = { 1 }

  B - A = { 4, 5 }

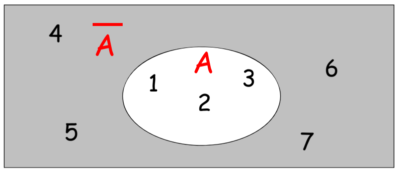


Venn diagrams

- Complement

Universal set = {1, …, 7}

A = { 1, 2, 3 } ⇒ $\overline{A}$ = { 4, 5, 6, 7}

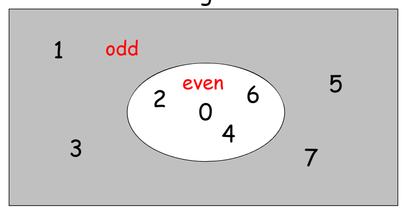


$\overline{\overline{A}} = A$

7

$\overline{\{\text{ even integers }\}} = \{\text{ odd integers }\}$

Integers



8

DeMorgan's Laws

$\overline{A \cup B} = \overline{A} \cap \overline{B}$

$\overline{A \cap B} = \overline{A} \cup \overline{B}$

9

Empty, Null Set: $\emptyset$

$\emptyset = \{ \}$

$S \cup \emptyset = S$

$S \cap \emptyset = \emptyset$

$S - \emptyset = S$

$\emptyset - S = \emptyset$

$\overline{\emptyset}$ = Universal Set

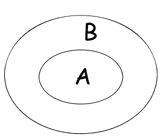
10

Subset

A = { 1, 2, 3} B = { 1, 2, 3, 4, 5 }

$A \subseteq B$

Proper Subset: $A \subset B$

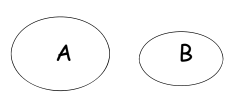


11

Disjoint Sets

A = { 1, 2, 3 } B = { 5, 6}

$A \cap B = \emptyset$



12

Set Cardinality

- For finite sets

$A = \{ 2, 5, 7 \}$

$|A| = 3$

(set size)

13

Powersets

A powerset is a set of sets

$S = \{ a, b, c \}$

Powerset of S = the set of all the subsets of S

$2^S = \{ \emptyset, \{a\}, \{b\}, \{c\}, \{a, b\}, \{a, c\}, \{b, c\}, \{a, b, c\} \}$

Observation: $| 2^S | = 2^{|S|}$ $(8 = 2^3)$

14

Cartesian Product

$A = \{ 2, 4 \}$ $B = \{ 2, 3, 5 \}$

$A \times B = \{ (2, 2), (2, 3), (2, 5), (4, 2), (4, 3), (4, 5) \}$

$|A \times B| = |A|\ |B|$

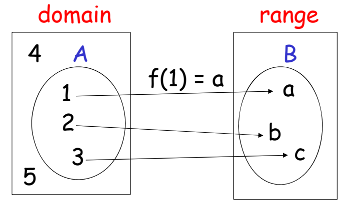
Generalizes to more than two sets

$A \times B \times \ldots \times Z$

15

FUNCTIONS

domain range



$f : A \to B$

If A = domain

then f is a total function

otherwise f is a partial function

16

RELATIONS

$R = \{(x_1, y_1), (x_2, y_2), (x_3, y_3), \ldots\}$

$x_i \, R \, y_i$

e. g. if R = '>': $2 > 1$, $3 > 2$, $3 > 1$

17

Equivalence Relations

- Reflexive: $x R x$
- Symmetric: $x R y \Rightarrow y R x$
- Transitive: $x R y$ and $y R z \Rightarrow x R z$

Example: R = '='

- $x = x$
- $x = y \Rightarrow y = x$
- $x = y$ and $y = z \Rightarrow x = z$

18

Equivalence Classes

For equivalence relation R

equivalence class of x = {y : x R y}

Example:

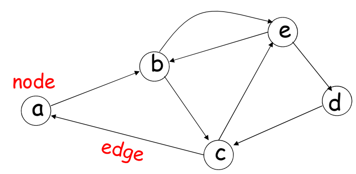
R = { (1, 1), (2, 2), (1, 2), (2, 1),
(3, 3), (4, 4), (3, 4), (4, 3) }

Equivalence class of 1 = {1, 2}

Equivalence class of 3 = {3, 4}

GRAPHS

A directed graph



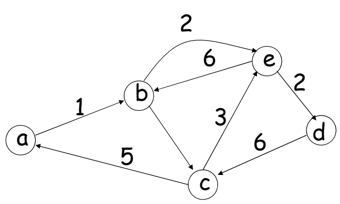
- Nodes (Vertices)

V = { a, b, c, d, e }

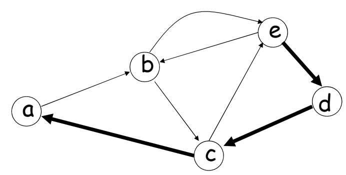
- Edges

E = { (a,b), (b,c), (b,e),(c,a), (c,e), (d,c), (e,b), (e,d) }

Labeled Graph



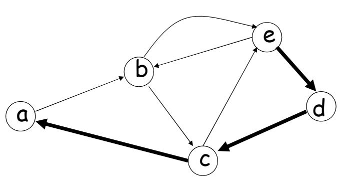
Walk



Walk is a sequence of adjacent edges

(e, d), (d, c), (c, a)

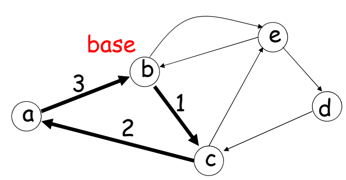
Path



Path is a walk where no edge is repeated

Simple path: no node is repeated

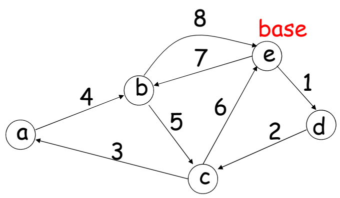
Cycle



Cycle: a walk from a node (base) to itself

Simple cycle: only the base node is repeated

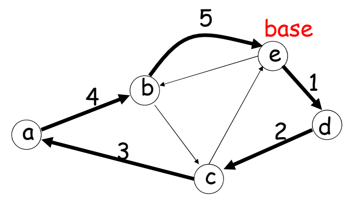
Euler Tour



A cycle that contains each edge once

25

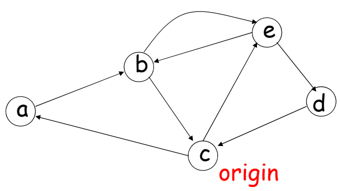
Hamiltonian Cycle



A simple cycle that contains all nodes

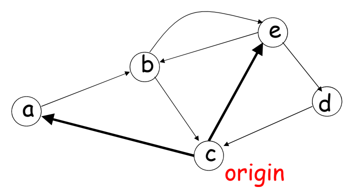
26

Finding All Simple Paths



27

Step 1

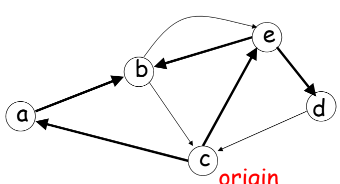


(c, a)

(c, e)

28

Step 2



(c, a)

(c, a), (a, b)

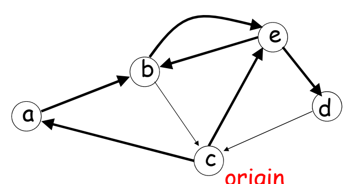
(c, e)

(c, e), (e, b)

(c, e), (e, d)

29

Step 3



(c, a)

(c, a), (a, b)

(c, a), (a, b), (b, e)

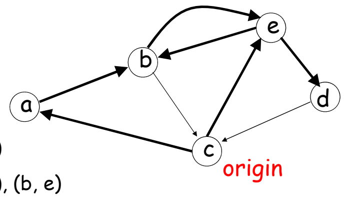
(c, e)

(c, e), (e, b)

(c, e), (e, d)

30

Step 4



(c, a)

(c, a), (a, b)

(c, a), (a, b), (b, e)

(c, a), (a, b), (b, e), (e,d)

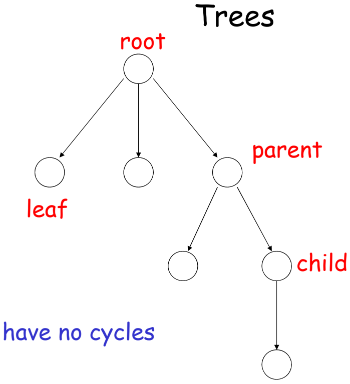
(c, e)

(c, e), (e, b)

(c, e), (e, d)

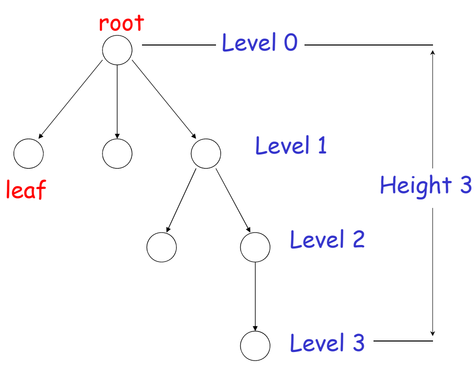
31

Trees



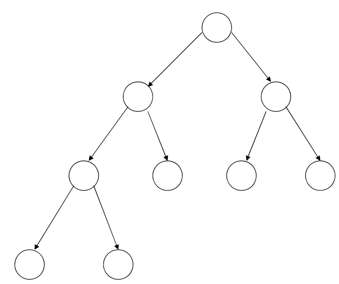
Trees have no cycles

32



33

Binary Trees



34

PROOF TECHNIQUES

- Proof by induction

- Proof by contradiction

35

Induction

We have statements $P_1, P_2, P_3, \ldots$

If we know

- for some b that $P_1, P_2, \ldots, P_b$ are true
- for any k >= b that

  $P_1, P_2, \ldots, P_k$ imply P_{k+1}

Then

Every P_i is true

36

Proof by Induction

- Inductive basis

  Find $P_1, P_2, \ldots, P_b$ which are true

- Inductive hypothesis

  Let's assume $P_1, P_2, \ldots, P_k$ are true,

  for any $k \geq b$

- Inductive step

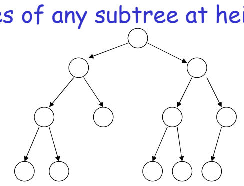
  Show that P_{k+1} is true

Example

Theorem: *A binary tree of height n has at most 2^n leaves.*

Proof by induction:

let $L(i)$ be the maximum number of *leaves of any subtree at height i*



We want to show: $L(i) <= 2^i$

- Inductive basis

  $L(0) = 1$ (the root node) 

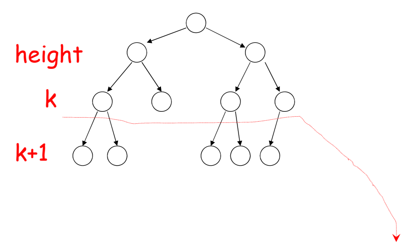
- Inductive hypothesis

  Let's assume $L(i) <= 2^i$ for all $i = 0, 1, \ldots, k$

- Induction step

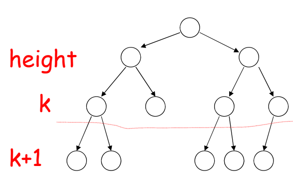
  we need to show that $L(k + 1) <= 2^{k+1}$

Induction Step



From Inductive hypothesis: $L(k) <= 2^k$

Induction Step



$L(k) <= 2^k$

$L(k+1) <= 2 * L(k) <= 2 * 2^k = 2^{k+1}$

(we add at most two nodes for every leaf of level k)

Remark

Recursion is another thing

Example of recursive function:

$f(n) = f(n-1) + f(n-2)$

$f(0) = 1, \quad f(1) = 1$

Proof by Contradiction

We want to prove that a statement P is true

- we assume that P is false
- then we arrive at an incorrect conclusion
- therefore, statement P must be true

43

Example

Theorem: $\sqrt{2}$ is not rational

Proof:

Assume by contradiction that it is rational

$\sqrt{2} = n/m$

n and m have no common factors

We will show that this is impossible

44

$\sqrt{2} = n/m \Longrightarrow 2m^2 = n^2$

Therefore, n^2 is even $\Longrightarrow$ n is even
n = 2 k

$2m^2 = 4k^2 \Longrightarrow m^2 = 2k^2 \Longrightarrow$ m is even
m = 2 p

Thus, m and n have common factor 2

Contradiction!

45

Languages

46

A language is a set of strings

String: A sequence of letters

Examples: "cat", "dog", "house", …

Defined over an alphabet:

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

47

Alphabets and Strings

We will use small alphabets: $\Sigma = \{a, b\}$

Strings

a

ab $\qquad u = ab$

$abba$ $\qquad v = bbbaaa$

$baba$ $\qquad w = abba$

$aaabbbaabab$

48

String Operations

$w = a_1 a_2 \cdots a_n$ $\qquad$ $abba$

$v = b_1 b_2 \cdots b_m$ $\qquad$ $bbbaaa$

Concatenation

$wv = a_1 a_2 \cdots a_n b_1 b_2 \cdots b_m$ $\qquad$ $abbabbbaaa$

49

$w = a_1 a_2 \cdots a_n$ $\qquad$ $ababaaabbb$

Reverse

$w^R = a_n \cdots a_2 a_1$ $\qquad$ $bbbaaababa$

50

String Length

$w = a_1 a_2 \cdots a_n$

Length: $|w| = n$

Examples: $|abba| = 4$

$|aa| = 2$

$|a| = 1$

51

Length of Concatenation

$|uv| = |u| + |v|$

Example: $u = aab, \quad |u| = 3$

$v = abaab, \quad |v| = 5$

$|uv| = |aababaab| = 8$

$|uv| = |u| + |v| = 3 + 5 = 8$

52

Empty String

A string with no letters: λ

Observations: $|\lambda| = 0$

$\lambda w = w\lambda = w$

$\lambda abba = abba\lambda = abba$

53

Substring

Substring of string:

a subsequence of consecutive characters

String	Substring
$\underline{ab}bab$	ab
$\underline{abba}b$	$abba$
$ab\underline{b}ab$	b
$a\underline{bbab}$	$bbab$

54

Prefix and Suffix

$abbab$

Prefixes	Suffixes
λ	$abbab$
a	$bbab$
ab	bab
abb	ab
$abba$	b
$abbab$	λ

$w = uv$

prefix

suffix

55

Another Operation

$$w^n = \underbrace{ww\cdots w}_{n}$$

Example: $(abba)^2 = abbaabba$

Definition: $w^0 = \lambda$

$$(abba)^0 = \lambda$$

56

The * Operation

Σ^* : the set of all possible strings from alphabet Σ

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

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

57

The + Operation

Σ^+ : the set of all possible strings from alphabet Σ except λ

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

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

$\Sigma^+ = \Sigma^* - \lambda$

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

58

Languages

A language is any subset of Σ^*

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

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

Languages: $\{\lambda\}$

$\{a, aa, aab\}$

$\{\lambda, abba, baba, aa, ab, aaaaaa\}$

59

Note that:

Sets $\quad \emptyset = \{\ \} \neq \{\lambda\}$

Set size $\quad |\{\ \}| = |\emptyset| = 0$

Set size $\quad |\{\lambda\}| = 1$

String length $\quad |\lambda| = 0$

60

Another Example

An infinite language $L = \{a^n b^n : n \geq 0\}$

$$\left.\begin{array}{l} \lambda \\ ab \\ aabb \\ aaaaabbbbb \end{array}\right\} \in L \qquad abb \notin L$$

61

Operations on Languages

The usual set operations

$$\{a, ab, aaaa\} \cup \{bb, ab\} = \{a, ab, bb, aaaa\}$$
$$\{a, ab, aaaa\} \cap \{bb, ab\} = \{ab\}$$
$$\{a, ab, aaaa\} - \{bb, ab\} = \{a, aaaa\}$$

Complement: $\overline{L} = \Sigma^* - L$

$$\overline{\{a, ba\}} = \{\lambda, b, aa, ab, bb, aaa, \ldots\}$$

62

Reverse

Definition: $L^R = \{w^R : w \in L\}$

Examples: $\{ab, aab, baba\}^R = \{ba, baa, abab\}$

$$L = \{a^n b^n : n \geq 0\}$$

$$L^R = \{b^n a^n : n \geq 0\}$$

63

Concatenation

Definition: $L_1 L_2 = \{xy : x \in L_1, y \in L_2\}$

Example: $\{a, ab, ba\}\{b, aa\}$

$$= \{ab, aaa, abb, abaa, bab, baaa\}$$

64

Another Operation

Definition: $L^n = \underbrace{LL \cdots L}_{n}$

$\{a,b\}^3 = \{a,b\}\{a,b\}\{a,b\} =$
$\{aaa, aab, aba, abb, baa, bab, bba, bbb\}$

Special case: $L^0 = \{\lambda\}$

$$\{a, bba, aaa\}^0 = \{\lambda\}$$

65

More Examples

$$L = \{a^n b^n : n \geq 0\}$$

$$L^2 = \{a^n b^n a^m b^m : n, m \geq 0\}$$

$$aabbaaabbb \in L^2$$

66

Star-Closure (Kleene *)

Definition: $L^* = L^0 \cup L^1 \cup L^2 \cdots$

Example:

$$\{a,bb\}^* = \left\{ \begin{array}{l} \lambda, \\ a, bb, \\ aa, abb, bba, bbbb, \\ aaa, aabb, abba, abbbb, \ldots \end{array} \right\}$$

67

Positive Closure

Definition: $L^+ = L^1 \cup L^2 \cup \cdots$

$$= L^* - \{\lambda\}$$

$$\{a,bb\}^+ = \left\{ \begin{array}{l} a, bb, \\ aa, abb, bba, bbbb, \\ aaa, aabb, abba, abbbb, \ldots \end{array} \right\}$$

68