CS 4384 INTRODUCTION

Theory of computation comprises:

- Automata theory
- Computability theory
- Complexity theory

Q. What are capabilities/limitations of computers?

Origin of theory of computation:

- A central notion is algorithm (e.g. Euclid's algor. for computing gcd of integers)
  → Theory of comp. started long time ago.

- A fundamental question is:
  Is there an algor. to solve a given problem?

For example, consider

Hilbert's 10th problem:

<u>Input</u>. A diophantine equation system, e.g.
$P_1(x_1, \dots, x_n), \dots, P_m(x_1, \dots, x_n)$
$\in \mathbb{Z}[x_1, \dots, x_n]$

<u>Output</u>. - yes if $\exists\, a_1, \dots, a_n \in \mathbb{Z}$
s.t. $P_1(a_1, \dots, a_n) = 0$,
.... $P_m(a_1, \dots, a_n) = 0$
- No, otherwise

<u>Q</u>. Is Hilbert's 10th problem algorithmically solvable?
(Hilbert, 1900)

1930s : Turing machines, Rec. fctns
→ Theory of computability

1970 : Matijasevic proved
Hilbert's 10th problem is
algorithmically
unsolvable.

In 1960s :

- Linguistics : N. Chomsky
intro. phrase-structured grammar
- Programming : BNF to
describe syntax of ALGOL 60
- Biology : Finite automata
to describe neural nets.

Automata theory:

How to model various types of computation?

Computability theory:

What is computability? Is a given problem computable?

Complexity theory:

If a problem is solvable, how much resources (time/memory) does it require?

Chapter 1. Math Background

Review of graphs, relations, strings, languages, ind. def., types of proof

Functions & Relations

The Cartesian product of sets A, B is $A \times B = \{ (a,b) \mid a \in A, b \in B \}$

A function / mapping takes an input and produces an output.

Same inputs produce same outputs

Set of inputs of fct. f: domain

Set of outputs contained in: range

For fct f with domain D and range R we write

$$f : D \longrightarrow R$$

For subset $A \subseteq D$ we write

$$f(A) := \{ f(a) \mid a \in A \}$$

$f: D \rightarrow R$ is said to be

- one-one if $a \neq a' \Rightarrow f(a) \neq f(a')$
  (distinct inputs produce dist. outputs)
- onto if $f(D) = R$
  $(\forall b \in R : \exists a \in D : f(a) = b)$
- a bijection if f is one-one & onto

Notation:

$\mathbb{N}$ = set of natural numbers
$(= \{1, 2, 3, \ldots\})$

$\mathbb{Z}$ = set of integers $(= \{\ldots -2, -1, 0, 1, \ldots\})$

$\mathbb{Q}$ = set of rationals

$\mathbb{R}$ = set of reals

For $k \geq 1$:

$\mathbb{Z}_k = \{0, 1, 2, \ldots, k-1\}$
= set of integers modulo k

If $f : A_1 \times \cdots \times A_k \longrightarrow R$, then inputs of f are k-tuples $(a_1, \ldots, a_k)$

$a_1, \ldots, a_k$ are arguments to f

$k = 1$: f is unary fct

$k = 2$: f is binary fct

Ex: gcd : $\mathbb{Z}^2 \rightarrow \mathbb{Z}$

sort : $\mathbb{Z}^k \rightarrow \mathbb{Z}^k$

If range of f is $\{0,1\}$ $(= \{T, F\})$ f is called predicate / property

Ex: even : $\mathbb{Z} \longrightarrow \{0,1\}$

is property of integers

even $(0) = 1$, even $(-2) = 1$, even $(4) = 1$

even $(3) = 0$, even $(5) = 0$

A property with domain $\underbrace{A \times \cdots \times A}_{k \text{ times}} = A^k$ is called a k-ary relation on A

It's a subset of A^k.

$k = 1$: unary rel. $k = 2$: bin. rel.

<u>Ex.</u> (1) $<$ is bin. rel. on $\mathbb{N}$ ($\mathbb{Z}$ ...)

$<\, : \mathbb{N}^2 \longrightarrow \{0,1\}$

$<(1,0) = 0 \qquad <(1,3) = 1$

or $1 \not< 0 \qquad 1 < 3$

(2) $=$ is bin. rel. on $\mathbb{N}$ ($\mathbb{Z}$ ...)

A k-ary rel. R on a set A can also be described as a subset of A^k.

<u>Ex.</u> (1) $< \; = \{(i,j) \mid i<j\}$

$= \{(1,2), (1,3), \ldots, (2,3), (2,4), \ldots\}$

$\subseteq \mathbb{N}^2$

(2) $= \; = \{(i,i)\} \subseteq \mathbb{N}^2$

Properties of bin. rel.

Let R be a bin. rel. on A. If $(a,b) \in R$ we write $a R b$; otherwise $a \not R b$

R is said to be

reflexive if $a R a$ $\forall a \in A$

irreflexive if $a \not R a$ $\forall a \in A$

symmetric if $a R b \Rightarrow b R a$ $\forall a, b \in A$

antisymmetric if $a R b \wedge b R a \Rightarrow a = b$ $\forall a, b \in A$

transitive if $a R b \wedge b R c \Rightarrow a R c$ $\forall a, b, c \in A$

Ex. (1) $\leq$ on $\mathbb{N}$ is refl., antisym, trans.

(2) $<$ on $\mathbb{N}$ is irrefl, antisym., trans.

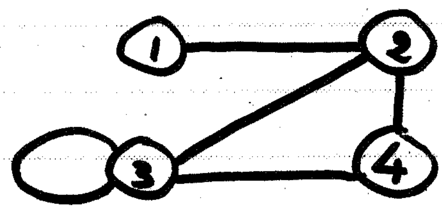
R is called an equiv. rel. if R is refl., sym. & trans.

Ex. For $k \geq 1$, def. $\equiv_k$ by $x \equiv_k y$ if $x - y = \lambda k$ for some $\lambda \in \mathbb{Z}$.
$\equiv_k$ is an equiv. rel.

Graphs & Digraphs

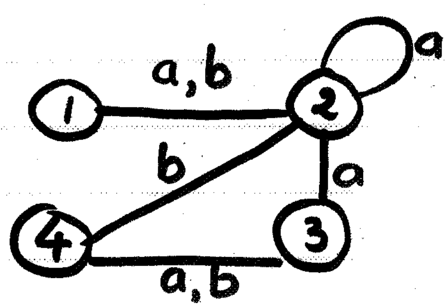
A graph is a pair $G = (V, E)$ where
- V is set of vertices
- $E \subseteq V \times V$ is set of edges

Ex.



A labeled graph is a graph whose edges are assigned labels from a finite sets

Ex.



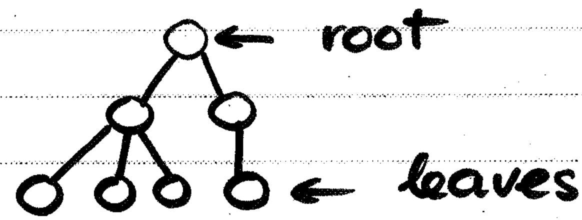
$\Sigma = \{a, b\}$

Other notions concerning graphs:

degree of a node
path, simple path, cycle
connected graph

A graph $T = (V, E)$ is a tree if T is connected and cycle-free.
Usually T has a distinguished node called root.
Node with degree 1 which is not a root is called a leaf; otherwise internal node.
(A single node tree has 1 leaf and 0 internal node.)

Ex.



If edges of $G = (V, E)$ are oriented, G is a directed graph (digraph)

Other notions concerning digraphs:

indegree, outdegree of a node
directed path, simple path, cycle
strongly connected digraph

Strings & Languages

An alphabet Σ (or T) is a finite set of symbols.

Ex: $\Sigma = \{0,1\}$ (or $\{a,b\}$): bin. alph.

$\Sigma = \{0,1,\dots,9, A,B,\dots,Z\}$

$\Sigma = \{0\}$ (or $\{1\}$ or $\{a\}$): unary alph.

A string over Σ is a finite sequence of symbols from Σ.

The length of a string w over Σ is the number of occurrences of symbols

Notation: $|w|$

For $a \in \Sigma$, $|w|_a$ denotes the number of occurrences of a in w. ($\#_a(w)$)

Other notation for $|w|_a$ is $\#_a(w)$

Ex: $\Sigma = \{0,1\}$ $w = 011001$

$|w| = 6$, $|w|_0 = 3 = \#_1(w)$

The empty string is of length 0 and denoted by ε or λ

The reverse of a string $w = w_1 w_2 \dots w_k$, $w_1, \dots, w_k \in \Sigma$, is $w^R = w_k w_{k-1} \dots w_1$

The concatenation of two strings

$x = x_1 \dots x_m$ and $y = y_1 \dots y_n$ is

$$xy = x \cdot y = x_1 \dots x_m \, y_1 \dots y_n$$

Thus: $|xy| = |x| + |y|$

$$\#_a(xy) = \#_a(x) + \#_a(y)$$

If $\left\{ \begin{array}{l} w = xyz \\ w = yz \\ w = xy \end{array} \right\}$, then y is $\left\{ \begin{array}{l} \text{substring} \\ \text{prefix} \\ \text{suffix} \end{array} \right\}$ of w

The lexicographic ordering of strings:

- shorter strings precede longer ones
- strings of same length are ordered according to dictionary ordering (assuming a total order of Σ)

A language over Σ is simply a set of strings over Σ.

If $w = w^R$, w is a palindrome.

Recursive Definitions

Ex: (set of well-formed parentheses strings D over $\Sigma = \{(,)\}$)

(1) Basis. ε is in D

(2) Recursive step. If x, y are strings in D, then so are (x) and xy.

(3) Nothing else is in D unless it is obtained from (1) by finitely many applications of (2).

Ex: (set of palindromes over Σ)

(1) Basis. ε and a are palindromes $\forall a \in \Sigma$

(2) Recursive step. If w is a palindrome, then so are awa, $\forall a \in \Sigma$

(3) As before ...

Remark. If a, $a \in \Sigma$, is not included in the basis, then odd-length palindromes are not included in the definition.

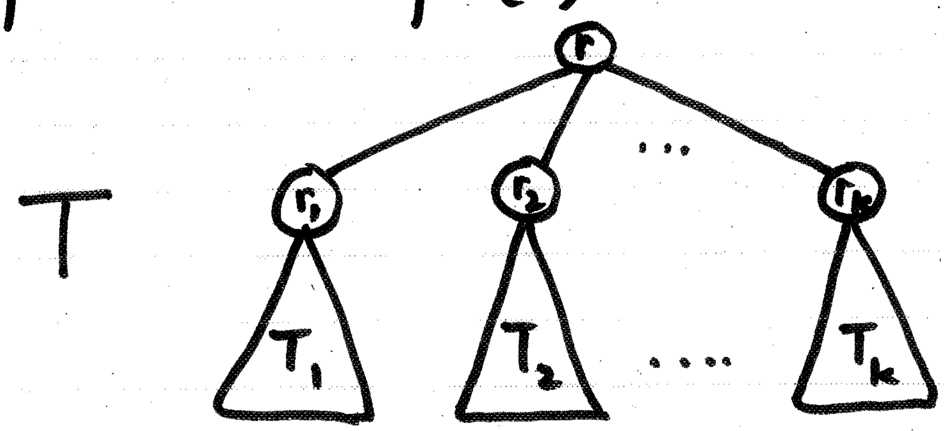
Ex. (Ordered trees)

(1) Basis. Every single node n is an ordered tree with n being root

(2) Recursive step. Let $T_1, \ldots, T_k$ be ordered tree with roots $r_1, \ldots, r_k$, and r be a new node. Then a new ordered tree T is obtained by

- adding edges $(r, r_1), \ldots, (r, r_k)$
- making r root of T

(3) Nothing else is a tree unless it's obtained from (1) by finitely many applications of (2).



Inductive Proofs

Ex: Claim. $\forall w \in D: \#_((w) = \#_)(w)$

Proof. (1) Basis. $\#_((\varepsilon) = 0 = \#_)(\varepsilon)$

(2) Ind. step.

Ind. Hypothesis: Assume that $x, y \in D$ satisfy

$\#_((x) = \#_)(x)$ and $\#_((y) = \#_)(y)$.

Consider $w = (x)$ or $w = xy$.

Case 1. $w = (x)$. Then

$$\#_((w) = \#_((\underbrace{(x)}_{w}) = \#_((x) + 1$$
$$= \#_)(x) + 1 \quad \text{(ind. hyp.)}$$
$$= \#_)((x)) = \#_)(w)$$

Case 2. $w = xy$

$$\#_((w) = \#_((xy) = \#_((x) + \#_((y)$$
$$= \#_)(x) + \#_)(y) \quad \text{(ind. hyp)}$$
$$= \#_)(xy)$$
$$= \#_)(w) \quad \square$$

(accord. to rec. def)

Ex: **Claim.** For any palindrome w : $w = w^R$

Proof. (1) **Basis.** $w = \varepsilon$: $\varepsilon = \varepsilon^R$

$w = a \in \Sigma$: $a = a^R$

(2) **Ind. Step**

Ind. hyp: Assume that a given palindrome x sat. $x = x^R$

Consider $w = axa$, $a \in \Sigma$

Clearly,
$$\begin{aligned} w^R &= (axa)^R \\ &= a x^R a \\ &= axa \quad \text{(ind. hyp)} \\ &= w \qquad \square \end{aligned}$$

Ex: A bin. tree is strictly binary if every internal node has exactly two children

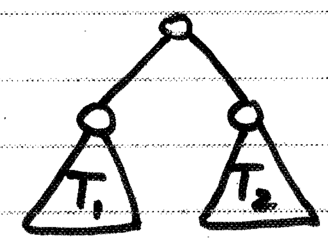
Claim. For all $n \geq 1$, every strictly bin. tree with n leaves has $n-1$ internal nodes.

Proof. (1) **Basis.** $n = 1$. T is a single-node tree with 1 leaf and 0 internal node

(2) Ind. step.

Ind. hyp: Assume for some $n \geq 1$ that claim holds true for all strictly bin. trees with k leaves where $1 \leq k \leq n$.

Consider a strictly bin. tree T with $n+1$ leaves. Then T has the form

$T =$ 

That is T has 2 strictly bin. subtrees T_1, T_2 with n_1 and n_2 leaves, where $n_1 + n_2 = n+1$

By ind. hyp:

$\#$ of intern. nodes in $T_1 = n_1 - 1$

$\#$ of intern. nodes in $T_2 = n_2 - 1$

Hence, $\#$ of intern. nodes in T = $\#$ of intern. nodes in T_1 + $\#$ of intern. nodes in T_2 + 1 ← root

$= (n_1 - 1) + (n_2 - 1) + 1$

$= n_1 + n_2 - 1 = (n+1) - 1 = n$ $\blacksquare$

Ex: (Correctness of a formula to calculate the amount of monthly payments of mortgages)

P := principal = loan amount

I := yearly interest rate

Y := mothly payment

Q. $Y = ?$ s.t. mortgage is paid off in 30 years given P and I ?

For convenience, def.

monthly multiplier $M = 1 + \frac{I}{12}$

Let P_t := loan amount after t months

Then:

$$
\begin{aligned}
P_0 &= P \\
P_1 &= P_0\left(1+\tfrac{I}{12}\right) - Y = P_0 M - Y \\
P_2 &= P_1 M - Y = (P_0 M - Y)M - Y \\
&= P_0 M^2 - Y(1+M) \\
P_3 &= P_0 M^3 - Y(1+M+M^2) \\
&\vdots \\
P_t &= P_0 M^t - Y(1+M+M^2+\cdots+M^{t-1})
\end{aligned}
$$

Claim. $P_t = PM^t - Y\dfrac{M^t - 1}{M-1}$

Claim can also be proved by ind.

<u>Pf.</u> (Claim) (1) <u>Basis</u>. $t=0$: $P_0 = P$

(2) <u>Ind. step</u>.

<u>Ind. Hyp</u>. Suppose that claim holds true for $t=k$

$k \to k+1$:

$$
\begin{aligned}
P_{k+1} &= P_k M - Y \\
&= \left[PM^k - Y \frac{M^k - 1}{M-1} \right] M - Y \quad \text{(ind hyp)} \\
&= PM^{k+1} - Y \frac{M^k - 1}{M-1} M - Y \\
&= PM^{k+1} - Y \left(\frac{M^k - 1}{M-1} M + 1 \right) \\
&= PM^{k+1} - Y \left(\frac{M^{k+1} - M}{M-1} + \frac{M-1}{M-1} \right) \\
&= PM^{k+1} - Y \frac{M^{k+1} - 1}{M-1} \quad \square
\end{aligned}
$$

If $t = 360$ ($= 30$ years $\times 12$), then $P_{360} = 0$ implies

$$P.M^{360} - Y \left(\frac{M^{360} - 1}{M-1} \right) = 0$$

or

$$
\begin{aligned}
Y &= PM^{360} \cdot \frac{M-1}{M^{360} - 1} \\
&= \frac{I}{12} \cdot P \cdot \frac{M^{360}}{M^{360} - 1} \quad \square
\end{aligned}
$$

Other types of proofs

Proof by Contradiction

Ex: Claim: $\sqrt{2}$ is irrational

Pf. We'll make use of the following

Fact. k is even $\Longleftrightarrow$ k^2 is even

Now suppose by way of contradiction that $\sqrt{2}$ were rational. Then

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

where $\gcd(m,n) = 1$, i.e., m, n are relatively prime.

Squarring both sides gives:

$$2n^2 = m^2$$

Clearly, m^2 is even. From above fact, it follows that m is even.

Hence, $m = 2\ell$ for some ℓ.

Therefore, $m^2 = 4\ell^2 = 2n^2$

Thus, $n^2 = 2\ell^2$.

So, n^2 is even. Again by above fact, n is even. Since m, n are both even, they cannot be rel. prime $\rightsquigarrow$ Contrad. $\square$

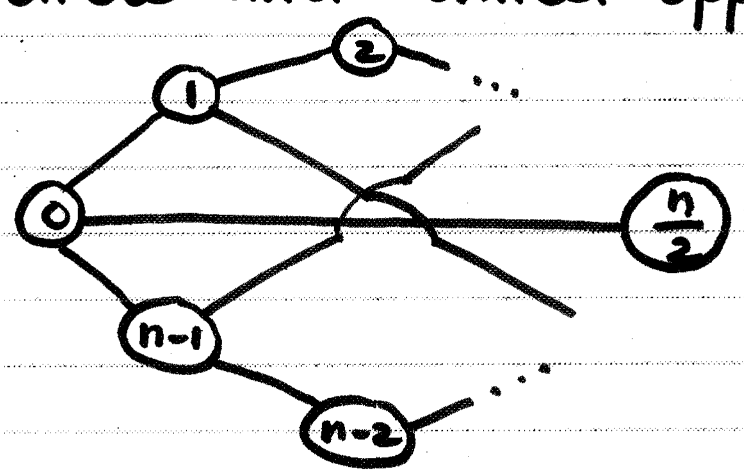
Proof by construction

Most proofs in CS are constructive. We devise algorithms to construct objects we're looking for. (Mathematicians are interested in proving the existence of such objects.)

Ex: A k-regular graph is a graph in which every node has degree k.

Claim. $\forall$ even $n \geq 4$: $\exists$ 3-regular graphs with n nodes.

Proof. Place the n nodes $0, 1, \ldots, n-1$ on a circle and connect opposite nodes:



$$V = \{0, 1, 2, \ldots, n-1\}$$

$$E = \{(0,1), (1,2), \ldots, (n-1, 0)\} \cup \{(0, \tfrac{n}{2}), (1, \tfrac{n}{2}+1), \ldots, (\tfrac{n}{2}-1, n-1)\}$$

$\square$

Some Definitions

Let Σ be an alphabet. Then

$\Sigma^* =$ set of all strings over Σ

e.g. $\{a,b\}^* = \{\varepsilon, a, b, aa, ab, ba, bb, \ldots\}$

For $L_i \subseteq \Sigma^*$, $i=1,2$, the concatenation of L_1 with L_2 is

$$L_1L_2 = L_1 \cdot L_2 = \{uv \mid u \in L_1 \wedge v \in L_2\}$$

For $x \in \Sigma^*$ and $n \geq 0$:

$$x^n := \begin{cases} \varepsilon & \text{if } n=0 \\ x^{n-1}x & \text{otherwise} \end{cases}$$

For $L \subseteq \Sigma^*$ and $n \geq 0$:

$$L^n := \begin{cases} \{\varepsilon\} & \text{if } n=0 \\ L^{n-1}L & \text{otherwise} \end{cases}$$

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

(1) $L_1 = \{a, ab\}$ $\quad L_2 = \{\varepsilon, b\}$

$$L_1L_2 = \{a, ab, \cancel{ab}, abb\}$$

So, $\text{Card}(L_1L_2) = 3 \neq \text{Card}(L_1) \times \text{Card}(L_2)$

(2) $L_1 = \{a^n \mid n \geq 0\}$, $L_2 = \{b^n \mid n \geq 0\}$

$L_1 L_2 = \{a^m b^n \mid m, n \geq 0\}$

$L_1 L_1 = \{a^n \mid n \geq 0\} = L_1$

(3) $L_1 = \{a^n \mid n \geq 1\}$

$L_1 L_1 = \{a^n \mid n \geq 2\} \neq L_1$

<u>Def.</u> For $L \subseteq \Sigma^*$:

The Kleene closure of L is

$$L^* = L^0 \cup L^1 \cup L^2 \cup \dots = \bigcup_{n=0}^{\infty} L^n$$

The positive closure of L is

$$L^+ = L^1 \cup L^2 \cup L^3 \cup \dots = \bigcup_{n=1}^{\infty} L^n$$

<u>Ex.</u> (1) $L_1 = \{a^n \mid n \geq 0\}$

$L_1^2 = L_1$. By ind. $L_1^n = L_1 \quad \forall n \geq 1$

Hence, $L_1^* = L_1 = L_1^+$

(2) $L_2 = \{a^n \mid n \geq 1\}$

$L_2^* = \{\varepsilon\} \cup L_2 \cup L_2^2 \cup \dots = L_1$

$L_2^+ = L_2 \cup L_2^2 \cup \dots = L_2$

$\neq$