Introduction

Why do we study Theory of Computation

Outline

- Importance of Theory of Computation
- Languages
- Languages and Problems

What is Computation ?

- Sequence of mathematical operations ?
  - What are, and are not, mathematical operations?
- Sequence of well-defined operations
  - How many operations ?
    - The fewer, the better.
  - Which operations ?
    - The simpler, the better.

What do we study in Theory of Computation ?

- What is computable, and what is not ?
- Basis of
  - Algorithm analysis
  - Complexity theory
- What a computer can and cannot do
- Are you trying to write a non-existing program?
  - Can you make your program more efficient?

What do we study in Complexity Theory ?

- What is easy, and what is difficult, to compute ?

- What is easy, and what is hard for computers to do?
- Is your cryptographic scheme safe?

Applications in Computer Science

- Analysis of algorithms
- Complexity Theory
- Cryptography
- Compilers
- Circuit design

History of Theory of Computation

- 1936 Alan Turing invented the *Turing machine*, and proved that there exists an *unsolvable problem*.
- 1940's Stored-program computers were built.
- 1943 McCulloch and Pitts invented *finite automata*.
- 1956 Kleene invented *regular expressions* and proved the equivalence of regular expression and finite automata.

History of Theory of Computation

- 1956 Chomsky defined *Chomsky hierarchy*, which organized languages recognized by different automata into hierarchical classes.
- 1959 Rabin and Scott introduced *nondeterministic finite automata* and proved its equivalence to (deterministic) finite automata.
- 1950's-1960's More works on languages, grammars, and compilers

History of Theory of Computation

- 1965 Hartmantis and Stearns defined *time complexity*, and Lewis, Hartmantis and Stearns defined *space complexity*.
- 1971 Cook showed the first *NP-complete problem*, the *satisfiability* problem.
- 1972 Karp Showed many other NP-complete problems.

History of Theory of Computation

- 1976 Diffie and Helllman defined *Modern Cryptography* based on NP-complete problems.
- 1978 Rivest, Shamir and Adelman proposed a public-key encryption scheme, *RSA*.

Alphabet and Strings

- An *alphabet* is a finite, non-empty set of symbols.
  - $\{0,1\}$ is a binary alphabet.
  - $\{A, B, \ldots, Z, a, b, \ldots, z\}$ is an English alphabet.
- A *string* over an alphabet Σ is a sequence of any number of symbols from Σ.
  - 0, 1, 11, 00, and 01101 are strings over $\{0, 1\}$.
  - *Cat*, *CAT*, and *compute* are strings over the English alphabet.

Empty String

- An *empty string*, denoted by ε, is a string containing no symbol.
  - ε is a string over any alphabet.

Length

- The length of a string x, denoted by $length(x)$, is the number of positions of symbols in the string.

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

$length(automata) = 8$

$length(computation) = 11$

$length(\varepsilon) = 0$

- $x(i)$, denotes the symbol in the i^{th} position of a string x, for $1 \leq i \leq length(x)$.

String Operations

- Concatenation
- Substring
- Reversal

Concatenation

- The concatenation of strings x and y, denoted by $x \cdot y$ or $x\,y$, is a string z such that:
  - $z(i) = x(i)$ for $1 \le i \le length(x)$
  - $z(i) = y(i)$ for $length(x) < i \le length(x) + length(y)$
- Example
  - $automata \cdot computation = automatacomputation$

Concatenation

The concatenation of string x for n times, where $n \geq 0$, is denoted by x^n

- $x^0 = \varepsilon$
- $x^1 = x$
- $x^2 = x\,x$
- $x^3 = x\,x\,x$
- …

Substring

Let x and y be strings over an alphabet Σ

The string x is a substring of y if there exist strings w and z over Σ such that $y = w\,x\,z$.

- ε is a substring of every string.
- For every string x, x is a substring of x itself.

Example

- ε, *comput* and *computation* are substrings of *computation*.

Reversal

Let x be a string over an alphabet Σ

The reversal of the string x, denoted by x^r, is a string such that

- if x is ε, then x^r is ε.
- If a is in Σ, y is in Σ^* and $x = a\,y$, then $x^r = y^r\,a$.

Example of Reversal

$$
\begin{aligned}
&(automata)^r \\
&= (utomata)^r\, a \\
&= (tomata)^r\, ua \\
&= (omata)^r\, tua \\
&= (mata)^r\, otua \\
&= (ata)^r\, motua \\
&= (ta)^r\, amotua \\
&= (a)^r\, tamotua \\
&= (\varepsilon)^r\, atamotua \\
&= \ atamotua
\end{aligned}
$$

Σ^*

- The set of strings created from any number (0 or 1 or ...) of symbols in an alphabet Σ is denoted by Σ^*.
- That is, $\Sigma^* = \cup_{i=0}^{\infty} \Sigma^i$
  - Let $\Sigma = \{\mathbf{\mathit{0}}, \mathbf{\mathit{1}}\}$.
  - $\Sigma^* = \{\varepsilon, \mathbf{\mathit{0}}, \mathbf{\mathit{1}}, \mathbf{\mathit{00}}, \mathbf{\mathit{01}}, \mathbf{\mathit{10}}, \mathbf{\mathit{11}}, \mathbf{\mathit{000}}, \mathbf{\mathit{001}}, \mathbf{\mathit{010}}, \mathbf{\mathit{011}}, \ldots\}$.

Σ^+

- The set of strings created from at least one symbol (1 or 2 or ...) in an alphabet Σ is denoted by Σ^+.
- That is, $\Sigma^+ = \cup_{i=1}^{\infty} \Sigma^i$

  $= \cup_{i=0..\infty} \Sigma^i - \Sigma^0$

  $= \cup_{i=0..\infty} \Sigma^i - \{\varepsilon\}$
- Let $\Sigma = \{\mathbf{0}, \mathbf{1}\}$. $\Sigma^+ = \{\mathbf{0}, \mathbf{1}, \mathbf{00}, \mathbf{01}, \mathbf{10}, \mathbf{11}, \mathbf{000}, \mathbf{001}, \mathbf{010}, \mathbf{011}, \ldots\}$.

  Σ^* and Σ^+ are infinite sets.

Languages

- A language over an alphabet Σ is a set of strings over Σ.
  - Let $\Sigma = \{0, 1\}$ be the alphabet.
  - $L_e = \{\omega \in \Sigma^* \mid$ the number of 1's in ω is even$\}$.
  - ε, $0, 00, 11, 000, 110, 101, 011, 0000, 1100, 1010, 1001, 0110, 0101, 0011, \ldots$ are in L_e

Operations on Languages

- Complementation
- Union
- Intersection
- Concatenation
- Reversal
- Closure

Complementation

Let L be a language over an alphabet Σ.
The complementation of L, denoted by $\overline{L}$, is $\Sigma^* - L$.

Example:

Let $\Sigma = \{0, 1\}$ be the alphabet.

$L_e = \{\omega \in \Sigma^* \mid$ the number of 1's in ω is even$\}$.

$\overline{L_e} = \{\omega \in \Sigma^* \mid$ the number of 1's in ω is not even$\}$.

$\overline{L_e} = \{\omega \in \Sigma^* \mid$ the number of 1's in ω is odd$\}$.

Union

Let L_1 and L_2 be languages over an alphabet Σ.

The union of L_1 and L_2, denoted by $L_1 \cup L_2$, is $\{x \mid x$ is in L_1 or $L_2\}$.

Example:

$\{x \in \{0,1\}^* \mid x$ begins with $0\} \cup \{x \in \{0,1\}^* \mid x$ ends with $0\}$

$= \{x \in \{0,1\}^* \mid x$ begins or ends with $0\}$

Intersection

Let L_1 and L_2 be languages over an alphabet Σ.

The intersection of L_1 and L_2, denoted by $L_1 \cap L_2$, is $\{ x \mid x$ is in L_1 and $L_2\}$.

Example:

$\{ x \in \{0,1\}^* \mid x$ begins with $0\} \cap \{ x \in \{0,1\}^* \mid x$ ends with $0\}$

$= \{ x \in \{0,1\}^* \mid x$ begins and ends with $0\}$

Concatenation

Let L_1 and L_2 be languages over an alphabet Σ. The concatenation of L_1 and L_2, denoted by $L_1 \cdot L_2$, is $\{w_1 \cdot w_2 \mid w_1$ is in L_1 and w_2 is in $L_2\}$.

Example

$\{ x \in \{0,1\}^* \mid x$ begins with $0\} \cdot \{x \in \{0,1\}^* \mid x$ ends with $0\}$

$= \{ x \in \{0,1\}^* \mid x$ begins and ends with 0 and $length(x) \geq 2\}$

$\{ x \in \{0,1\}^* \mid x$ ends with $0\} \cdot \{x \in \{0,1\}^* \mid x$ begins with $0\}$

$= \{ x \in \{0,1\}^* \mid x$ has 00 as a substring$\}$

Reversal

Let L be a language over an alphabet Σ.

The reversal of L, denoted by L^r, is $\{w^r \mid w$ is in $L\}$.

Example

$\{x \in \{0,1\}^* \mid x$ begins with $0\}^r$

$= \{x \in \{0,1\}^* \mid x$ ends with $0\}$

$\{x \in \{0,1\}^* \mid x$ has 00 as a substring$\}^r$

$= \{x \in \{0,1\}^* \mid x$ has 00 as a substring$\}$

Kleene's closure

Let L be a language over an alphabet Σ.

The Kleene's closure of L, denoted by L^*, is $\{x \mid$ for an integer $n \geq 0$ $x = x_1 x_2 \ldots x_n$ and $x_1, x_2, \ldots, x_n$ are in $L\}$.

That is, $L^* = \cup_{i=0}^{\infty} L^i$

Example: Let $\Sigma = \{0,1\}$ and

$L_e = \{\omega \in \Sigma^* \mid$ the number of 1's in ω is even$\}$

$L_e^* = \{\omega \in \Sigma^* \mid$ the number of 1's in ω is even$\}$

$(\overline{L}_e)^* = \{\omega \in \Sigma^* \mid$ the number of 1's in ω is odd$\}^*$

$= \{\omega \in \Sigma^* \mid$ the number of 1's in $\omega > 0\}$

Closure

Let L be a language over an alphabet Σ.

The closure of L, denoted by L^+, is $\{ x \mid$ for an integer $n \geq 1$, $x = x_1 x_2 \ldots x_n$ and $x_1, x_2, \ldots, x_n$ are in $L\}$

That is, $L^+ = \cup_{i=1}^{\infty} L^i$

Example:

Let $\Sigma = \{0, 1\}$ be the alphabet.

$L_e = \{\omega \in \Sigma^* \mid$ the number of 1's in ω is even$\}$

$L_e^+ = \{\omega \in \Sigma^* \mid$ the number of 1's in ω is even$\}$

$= L_e^*$

Observation about Closure

$L^+ = L^* - \{\varepsilon\}$?

Example:

L = $\{\omega \in \Sigma^*$ | the number of 1's in ω is even$\}$

L^+ = $\{\omega \in \Sigma^*$ | the number of 1's in ω is even$\}$ = L_e^*

Why?

$L^* = L^+ \cup \{\varepsilon\}$?

Languages and Problems

- Problem
  - Example: What are prime numbers > 20?
- Decision problem
  - Problem with a YES/NO answer
  - Example: Given a positive integer n, is n a prime number > 20?
- Language
  - Example: $\{n \mid n$ is a prime number $> 20\}$

    $= \{23, 29, 31, 37, \ldots\}$

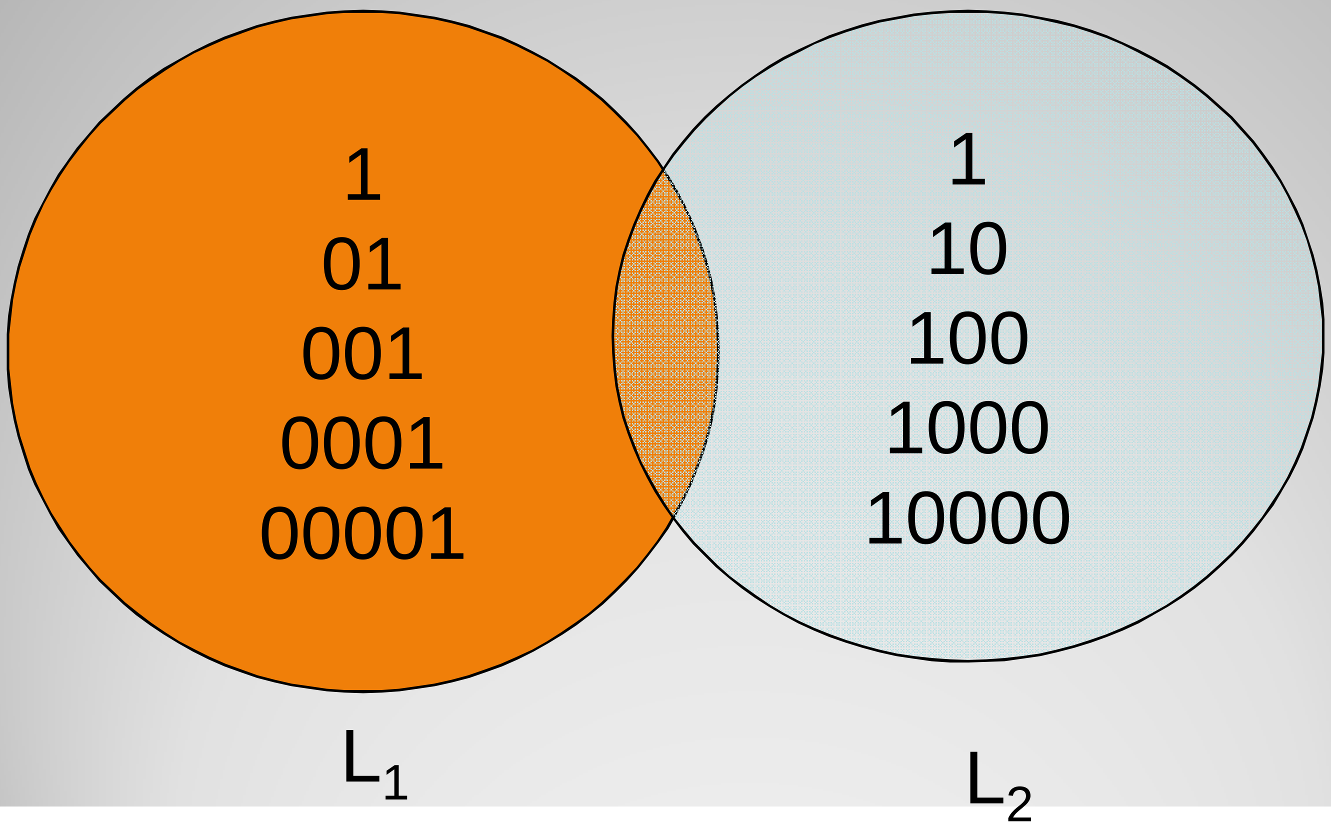
Language Recognition and Problem

- A problem is represented by a set of strings of the input whose answer for the corresponding problem is “YES”.
- a string is in a language = the answer of the corresponding problem for the string is “YES”
  - Let “Given a positive integer n, is n a prime number > 20?” be the problem P.
  - If a string represents an integer i in $\{m \mid m$ is a prime number > 20$\}$, then the answer for the problem P for $n = i$ is true.

Common misconception

Beware

A language is a set.



And, there is also a set of languages.