Theory of Computation: CS-202

Suggested readings

1. An introduction to FORMAL LANGUAGES and AUTOMATA by PETER LINZ.
2. Introduction to Automata Theory, Languages, And Computation by JOHN E. HOPCROFT, RAJEEV MOTWANI, JEFFREY D. ULLMAN
3. Theory of computer science: automata, languages and computation **by** K.L.P MISHRA

Content

- Review of set theory
- Formal Language
- String Operations
- Grammar

Review of set theory

A Set is a well defined collection of distinct objects.

- list of elements: $A = \{6, 12, 28\}$
- characteristic property: $B = \{x \mid x \text{ is a positive, even integer}\}$

Set membership: $12 \in A$, $9 \notin A$

Barber Paradox

Barber: One who shaves all those and those only, who do not shave themselves.

Question: Does Barber shave himself?

1. If he shaves himself he ceases to be Barber.
   (As Barber cant shave himself.) Contradiction
2. If the Barber does not shave himself then he must shave himself.
   (Barber can shave himself) Contradiction

- Subset- If A and B are two set, such that every element of A is also an element of B then $A \subseteq B$.

- Proper Subset- If A is subset of B, but B contains an element not in A, then $A \subset B$.
- Disjoin set- If A and B has no common element then $A \cap B = \varphi$, then the sets are said to be disjoint.
- Null Set-A set with no element is called null set.

Eg. {} or φ

Set Operations

Union ($\cup$) : $S_1 \cup S_2 = \{x : x \in S_1 \text{ or } x \in S_2\}$

Intersection ($\cap$) : $S_1 \cap S_2 = \{x : x \in S_1 \text{ and } x \in S_2\}$

Difference (-) : $S_1 - S_2 = \{x : x \in S_1 \text{ and } x \notin S_2\}$

Complement : $\overline{S} = \{x : x \in U \text{ and } x \notin S\}$

Example: Consider the set A and B

$A = \{6, 12, 28\}$

$B = \{x \mid x \text{ is a positive, even integer}\}$

then

$A \cup \varphi = A$

$A \cap \varphi = \varphi$

$\Phi' = U$

union: $A \cup \{9, 12\} = \{6, 9, 12, 28\}$

intersection: $A \cap \{9, 12\} = \{12\}$

difference: $A - \{9, 12\} = \{6, 28\}$

Set theory (continued)

Another set operation, called "taking the complement of a set", assumes a **universal set** .

Let $U = \{0, 1, 2, 3, 4, 5, 6, 7, 8, 9\}$ be the universal set.
Let $A = \{2, 4, 6, 8\}$
Then $\overline{A} = U - A = \{0, 1, 3, 5, 7, 9\}$

The **empty set** : $\varnothing = \{\}$

Set theory (continued)

The **cardinality** of a set is the number of elements in a set.

Let S = {2, 4, 6}

Then |S| = 3

The **powerset** of S, represented by 2^S, is the set of all subsets of S.

$2^S = \{\{\}, \{2\}, \{4\}, \{6\}, \{2,4\}, \{2,6\}, \{4,6\}, \{2,4,6\}\}$

The number of elements in a powerset is $|2^S| = 2^{|S|}$

Theory of Computation
Basic Concepts

- *Automaton*: a formal construct that accepts input, produces output, may have some temporary storage, and can make decisions
- *Formal Language*: a set of sentences formed from a set of symbols according to formal rules
- *Grammar*: a set of rules for generating the sentences in a formal language

In addition, the theory of computation is concerned with questions of computability (the types of problems computers can solve in principle) and complexity (the types of problems that can solved in practice).

Formal language

Alphabet = finite set of symbols or characters
examples: $\Sigma = \{a,b\}$, binary, ASCII

String = finite sequence of symbols from an alphabet
examples: aab, bbaba, also computer programs

A **formal language** is a set of strings over an alphabet
Examples of formal languages over alphabet $\Sigma = \{a, b\}$:

$L_1 = \{aa, aba, aababa, aa\}$
$L_2 = \{$all strings containing just two a's and any number of b's$\}$

A formal language can be finite or infinite.

String Operations

String Concatenation: The concatenation of two strings w and v is the string obtained by appending the symbols of v to the right of w.

Eg. $w=a_1a_2\dots.a_n$

$v=b_1b_2\dots.b_n$

$wv=a_1a_2\dots.a_n\ b_1b_2\dots.b_n$

String Reversal: It is obtained by writing the symbols in reverse order.

Eg. $w=a_1a_2\dots.a_n$

$w^R=a_na_{n-1}\dots a_1$

String Length: Let *w* be a string then |w| is the number of symbols in the string.

$w=a_1a_2$

$|w|=2$

Sub-String: Any string of consecutive character in some w is called Sub-String.
Example,

$w=a_1a_2a_3a_4a_5$

Substring of w= a_1a_2 ,a_2a_3 ….. etc

Prefix and sufix: Let w=vu be a string then the substring v and u are said to be a prefix and sufix of w.
Example,

w=abbab

Then, prefix={λ, a, ab,abb,abba,abbab}

Sufix={λ ,b,ab,bab,bbab}

The **empty string** , denoted λ , has some special properties:

$$|\lambda| = 0$$

$$\lambda w = w \lambda = w$$

Kleen or Star Closure Σ^*

If w is a string, then w^n stands for the string obtained by repeating w n times.

$w^0 = \lambda$

Σ^*=set of string obtained by concatening zero or more symbols from Σ.

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

$\Sigma^0=\{\lambda\}$

$\Sigma^1=\{a,b\}$

$\Sigma^2=\{aa,ab,ba,bb\}$.

……

$\Sigma^*= \{ \Sigma^0 \cup \Sigma^1 \cup \Sigma^2 \cup \ldots\ldots \}$

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

Note: Σ is finite by assumption, Σ^* and Σ^+ will always be infinite.

Suggested readings

1. An introduction to FORMAL LANGUAGES and AUTOMATA by PETER LINZ.
2. Introduction to Automata Theory, Languages, And Computation by JOHN E. HOPCROFT, RAJEEV MOTWANI, JEFFREY D. ULLMAN
3. Theory of computer science: automata, languages and computation **by** K.L.P MISHRA