Theory of Computation
Paper Code: CS-202

Content

- Formal Language
- Grammar
- Equivalence of Grammar

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.

Language

Language is defined as a subset of Σ^* .

Any set of strings on the alphabet Σ can be considered as a language

Example 1: $\Sigma = \{a,b\}$

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

Then the set $S=\{a, aa, aab\}$ is a language on Σ .

Example 2: $L=\{a^n b^n : n \geq 0\}$ is also a language on Σ.

The strings ab, aabb, aaabbb $\in L$

While abb $\notin L$.

Language (Continue…)

Note: Since languages are sets, so the union, intersection and difference of two languages are immediately defined.

Operations on languages

Set operations:

$L_1 \cup L_2 = \{x \mid x \in L_1 \text{ or } x \in L_2\}$ is union

$L_1 \cap L_2 = \{x \mid x \in L_1 \text{ and } x \in L_2\}$ is intersection

$L_1 - L_2 = \{x \mid x \in L_1 \text{ and } x \notin L_2\}$ is difference

$\overline{L} = \Sigma^* - L$ is complement

String operations:

$L^R = \{w^R \mid w \in L\}$ is "reverse of language"

$L_1L_2 = \{xy \mid x \in L_1, y \in L_2\}$ is "concatenation of languages"

$L^* = L^0 \cup L^1 \cup L^2 \ldots$ is "Kleene star" or "star closure"

$L^+ = L^1 \cup L^2 \ldots$ is positive closure

Grammars

A grammar G is defined as a quadruple:

$G = (V, T, S, P)$

Where V is a finite set of objects called variables

T is a finite set of objects called terminal symbols

$S \in V$ is a special symbol called the Start symbol

P is a finite set of productions or "production rules"

Sets V and T are nonempty and disjoint

All production rules are of the form

$X \to Y$

Where, $\mathbf{X \in (V \cup T)^{+}}$

$\mathbf{Y \in (V \cup T)^{*}}$

& the productions are applied as follows:

Suppose

$w = uXv$

and $X \to y$

$\Rightarrow z = uyv$

$w \Rightarrow z$

We say that w derives z or **z is derived from w**.

A production can be used wherever it is applicable, if

$\mathbf{w_1 \Rightarrow w_2 \Rightarrow w_3 \Rightarrow w_4 \Rightarrow \ldots\ldots\ldots \Rightarrow w_n}$

we say that $\Rightarrow$ $w_1 \overset{*}{\Rightarrow} w_n$

Here, * indicates that an unspecified number of steps (including zero) can be taken to derive **w_n from w_1**.

Note:

1. By applying the production rules in a different order, a given grammar can normally generate many strings.

2. The set of all such strings is the language defined or generated by the grammar.

Grammars

What is the relationship between a language and a grammar?

Let G = (V, T, S, P)

The set

$$L(G) = \{w \in T^* : S \stackrel{*}{\Rightarrow} w\}$$

is the language generated by G.

If $w \in L(G)$ then the sequence

$$S \Rightarrow w_1 \Rightarrow w_2 \Rightarrow w_3 \Rightarrow w_4 \Rightarrow \ldots\ldots\ldots \Rightarrow w_n \Rightarrow w$$

is a derivation of the sentence w.

The strings S, w_1, w_2 , …. w_n which contain variable as well as terminals are called "sentential form" of the derivation and '*w*' which contain all terminal is called sentence.

Example: 1

Consider the grammar

$$G_1 = (\{S\},\{a,b\},S, P)$$

With P is given by

1. $S \rightarrow aSb$
2. $S \rightarrow \lambda$

Then $S \xRightarrow{1} aSb \xRightarrow{1} aaSbb \xRightarrow{2} aabb$

So we can write

$S \Rightarrow aabb$

String aabb is a sentence and aaSbb is a sentential form.

So, $L(G_1)=\{a^n b^n : n \geq 0\}$

Example: 2

Consider the grammar

$G_2 = (\{S,A\},\{a,b\},S, P_1)$

With P_1 given by

1. $S \rightarrow aAb$
2. $S \rightarrow \lambda$
3. $A \rightarrow aAb$
4. $A \rightarrow \lambda$

Then $S \overset{1}{\Rightarrow} aAb \overset{3}{\Rightarrow} aaAbb \overset{4}{\Rightarrow} aabb$

So we can write

$S \Rightarrow aabb$

String aabb is a sentence and aaAbb is a sentential form.

So, $L(G_2)=\{a^n b^n : n \geq 0\}$

Equivalence of grammar

- Two grammars are equivalent if they generate the same language.

- In the above example G_1 and G_2 are equivalent

Example: 3

Find the grammar that generates language

$L = \{a^n b^{n+1} : n \geq 0\}$

Let $G=(\{S,A\},\{a,b\}, S,P)$

With P:

1. $S \rightarrow Ab$
2. $A \rightarrow aAb$
3. $A \rightarrow \lambda$

Example: 4

Take $\Sigma=\{a,b\}$ and let $n_a(w)$ and $n_b(w)$ denotes the number of a's and b's in the string w

The grammar G with production

1. $S \to SS$
2. $S \to aSb$
3. $S \to bSa$
4. $S \to \lambda$

Generate L(G).

$L=\{w : n_a(w) = n_b(w)\}$

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