TOC

Assignment - III

Dhananjay Porwal
EN19CS301110

1 Explain the closure properties of regular languages?

Ans Closure properties on regular language are defined as certain operations on regular language which are to produce regular language. Closure refers to some operation on a language, resulting in a new language that is of same "type" as originally operated on i.e. regular.

a) Kleen Closure

RS is a regular expression whose languages is L, M, R^* is a regular expression whose language is L^*.

b) Positive closure

RS is a regular expression whose language is L, M, R^+ is a regular expression whose language is L^+.

c) Complement

The complement of a language L (w.r.t. an alphabet E such that E^* contains L) is $E^* - L$. Since E^* is surely regular, the complement of a regular language is always regular.

d) Reverse Operator

Given language L, L^R is the set of strings whose reversal is in L

Ex - $L^R = \{0, 10, 001\}$

e) Union

Let L & M be the language of regular expression R & S respectively. Then R+S is a regular expression whose language is (L∪M)

f) Intersection

Let L & M be the languages of regular expression R & S, respectively then it is a regular expression whose language is L intersection M.

g) Set Difference operator

If L & M are regular expression, then so is L-M = strings in L but not M.

2 Clasrify the chomosky hierarchy?

Ans This are divided into four parts -

a) Type 0 : Unrestricted Grammer

Grammer production in the form of $(V+T)^* V (V+T)^*$
where, V = variable, T = Terminals

In type 0 there must be at least one variable on left side of production

for ex -

$$Sab \longrightarrow ba$$

$$A \longrightarrow S$$

Here, variables are S, A & Terminals are a, b.

b) Type 1 : Context Sensitive Grammer

& first of all, Type 1 Grammer should be Type 0.

& Grammer production in the form of $| | <= | |$
i.e. count of symbol in is less than or equal to

for Ex - $S \rightarrow AB$

$$AB \longrightarrow abc$$
$$B \longrightarrow b.$$

c) Type 2: Context free Grammar

} first of all, it should be Type 1.

} Left hand-side of production can have only one variable.

$$1 \quad 1 = 1$$

There is no restriction

for ex - $S \longrightarrow AB$

$A \rightarrow a$

$B \rightarrow b$

d) Type 3: Regular Grammar

This is most restricted form of grammer.
Type 3 should be in the given form only.
$N \longrightarrow VT / T$ (left-regular grammar)
$V \longrightarrow TN / T$ (right-regular grammar)

for ex - $S \longrightarrow a$

The above form is called as strictly regular grammar there is another form of regular grammar called extended regular grammar.

$V \longrightarrow VT^* / T^*$ (extended left - reg. grammar)

3 Explain Cock- Younger - Kasami (CYK) algo. in detail with example?

Ans The standard version of CYK operates only on context free grammars given in Chomsky normal form (CNF). Using Big O notations, the worst case running time of CYK is $O(n^3 \cdot |G|)$, where n is the length of the parsed string & $|G|$ is the size of the CNF grammar G.

The dynamic programming alg. requires the context free grammar to be rendered into Chomsky norm form. because it tests for possibilities to split the current sequence into two smaller sequences.

7	S						
6		VP					
5							
4	S						
3		VP			PP		
2	S					NP	
1	NP	V VP	det	N	P	det	N

<u>CYK table</u>

where,

$S \rightarrow NP\,VP$, $VP \rightarrow VP\ PP$, $NP \rightarrow V\ NP$,

$VP \rightarrow$ eats , $PP \rightarrow P\ NP$, $NP \rightarrow Det\ N$,

$NP \rightarrow$ She , $V \rightarrow$ eats , $P \rightarrow$ with ,

$N \rightarrow$ fish , $N \rightarrow$ fork , det $\rightarrow$ a

Note the ~~setence~~ sentence she eats a fish with a fork is analysed using CYK algo. In the following table P[i, j, k], is the no. of the row, j is the no. of column.

for readibility, the CYK table for P is represented here as a 2-D matrix m containing a set of non-terminal symbols, such that

R_k is in $M[i,j]$ if, and only if, $P[i,j,k]$.
In the above example, since a start symbol S is in $M[1,1]$, the sentence can be generated by the grammer.