

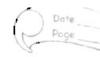
Assignment No.3

- (1) Explain the algorithm to eliminate null productions.
 - The production of the form A -> E are called as E-productions. such E-productions and makes the process of derivation complecated therefore all such E-production are unwanted. Hence eliminate all such production. However if empty string is part of language generated by grammer of language generated by grammer of i.e. NEL(G) then their must be the production significant that grammas where is is a start symbol of that grammas.

Algorithm to findout the nullable vari-

- If A→6 is a production in the grammer then A is nullable variable.
- If B→ \risk a production in the grammet & if all the symbol of \rightally found nullable than automatically \rightally also becomes nullable.

 we repeate this process until no more nullable variable found.
- Once nullable variable found successfully then transform each of the production such a that body of that production contains nullable variable once i.e. you



have to write the body of production with nullable variable once, without nullable variable once.

Repeat this process for all the production of grammer as well as for all the nullable variable of the grammer.

that empty string & is a part of language generated by that grammer,

In these case, their must be a production of these case in the resultant grammer obtained by eliminating null productions means in these case the production s -> & becomes exceptional & therefore mandatory in the grammar.

for example, O Eliminate following &-prode

i) 5 -- aSb laAb

A->E

soin. The nullable variable in thise ex is A . After that we will check the body of produin which that nullable variable appears:

After that, we write the body with nullable variable once, without nullable variable

so based on this simplified grammer upon eliminating null production can



| | be expressed as |
|-----|---|
| | i) s- asb |
| | ii) 5 → aAblab |
| | |
| (| |
| 0.2 | Explain the concept of useful and usele- |
| | as symbols in grammer. |
| | Concept of useful symbol- |
| | A symbol is called as useful, |
| | iff its satisfy following 2 criterias- |
| | iff its satisfy following 2 criterias - i) It should be reachable from statt |
| | symbol of grammer. |
| | |
| | · in It should derive certain part of Ilp |
| | - ii) It should derive certain part of I/p |
| | |
| | If a symbol is not following any of |
| | above contraint, such symbol is called |
| | as useless symbol isuch useless symbol |
| | offects on performance of parser, |
| | . i always eliminate such 'useless symbols. |
| | |
| | Examples on elimination of useless symbols |
| | S HBC BaB |
| - | A -> aA Bac aaq, ; |
| | 8 → bBbla |
| - | · · · · · · · · · · · · · · · · · · · |
| • | sol ⁰ :- |
| - | i) S→ABC |
| , | i) S -> BaB |
| - | An (iii) |
| | |

iv) A ---> BaC V) A --> aaa VI) B -> bBb viij B→a viii) $c \longrightarrow cA$ → AC Let separate all production in the grammer are above, usefull symbol first is is usefull always since, it is a start variable. All A, B, C are reachable from 5 but unfortunatly ic' obesnit derive any Ilp string , therefore first une useless symbol we found 'c' Upon finding 'c' as a useless, we can straight away delete all the produc-tion involving 'c' (either in head or in body or both) tions 15st, 4th, 9th, 9th to form the simplified CFG as i) $S \longrightarrow B \cap B \cup B$ ii) A -> aA iii) A --- aaa îu) B -- bBb Now, in the simplified grammar it is again observed that variable A is unreas chable, their itself you can announce A as an useless of we can delete all the prodn involving a ceither in head



or in body or in both)

so, accordingly and of 3rd production will be eliminated to form simplified

CFFG as

. s → B**1**B . **b** ↔ → bBb

0.3) Explain the algorithm to eliminate unit production.

The production of the form A→B is called as a unit production, where A, B

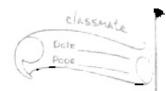
i.e. if the single variable is ederiving another single variable then such production are unit production such unit productions are absolutely unwanted.

performance of derivation such production increase the complexity of derivation.

therefore, all such unit productions must be eliminated.

Algorithm to eliminate unit production from given given algorithm -

O select the unit production $A \rightarrow B$ such a that their exist a production $B \rightarrow \infty$ where α is terminal.



| 1 For every | non-unit | produ | drammar. |
|--------------|----------------------------|-------|----------|
| repeat . the | · following | step | |
| Add | $A \longrightarrow \alpha$ | o the | drawwor. |

After completion of previous step ② eliminate the proof A → B from grammar.

For example,

5 - Aa 1B

B - Albb

A --> albClB

soin- first separate all the productions

i) s -- Aa

ii) **8**→ B

 $(ii) \Theta \longrightarrow A$

· iv B -> bb

WA - a

.vij A → bC

viij A -> B

5 ---> Aa

5 - bb

s → a

5 -- + bc

 $\beta \rightarrow a$

B- bc

B. bb

 $A \longrightarrow a$

A -> bc

 $A \longrightarrow bb$.



(14) Explain the Concept of union, concatation, kleen s closure of CFG. crus are closed wirito some of the function fundamental operation such as union. concatatation, kleen closure. However CFL's are not closed wirit intersection, complement operation, CFL's are supporting the operation like union, concatination, kleen closure as their underlying context free grammar are closed with these all operation. That means, upon performing this are operation on two or more crais the resu-Itant grammer obtained is also a CFG. Now, we will understand are supporting union, concatation and kleen closure oper ation. aoian li let G, is CFG generating CFL LI defined as G2 = (V1, 5,51, P), lly, another CFG G2 is generating CFL 12 defined as G2 = (V2, 2, 52, B). upon performing union on grammer Gri. Gz. We will get resultant CFG Gu generating CFL = LIUL2.
The grammar Gu can be defined as Gu = (Vu, Z, Pu, Su) where Vu = VI UV2 USu. I = IIP alphabet Su = start symbol of Ga.

| For ex, consider the grammer Gras |
|--|
| follows, |
| SI - AB |
| B - aAblab B - cBd/cd |
| This grammar generates CFL Li defin ed as. Li= {am bm cndn/m>1, n>1} |
| ed as le tambin cha man sing |
| Now, consider another & CFG G2 |
| $S_2 \longrightarrow \infty S_2 + 1 \propto 4$ |
| $S_2 \longrightarrow \infty S_{2y} \cdot 1 \times y$. It generates CFL L2 as |
| $L_2 = \{x^k y^k \mid k \ge 1\}$ |
| upon performing union on G1 & G2. |
| we will get resultant grammar Gu = Gil |
| G2. Expressed as |
| Gy = GrUG2: |
| Sy -> sls2 |
| $S_1 \longrightarrow AB$ |
| A aAb lab |
| B -> cBb/cd |
| S2 $\rightarrow x S_2 y xy$ with respect to $x = x + y = y = y = y$ |
| with respect to Gui, $Vu = \{51, 52, A, B, Su^2\}$, $\Sigma = \{a,b,c,d,x,y\}$ Su= Start symbol of Gui, $Pu = Set$ of production of GiV set of all production of GiV. |
| of Gu, Pu = set of production of Gu |
| set of all production of G2U su -sils2 |
| ii) concatation - |
| like union cea and |
| operation as a solici |
| ned below. |
| |

CHESCHATE



Let Gi is CFG defined as (Vi, E, Si, Pi) generating a CFL Li. Another grammer Oz defined as (Uz, I.Sz, Pz) generating another CFL as L2.

Upon performing concatation operation
i.e. joining enter grammar G2 and the of GI. we shall get resultant CFG GC expressed as Gc = (G1. G2) Gc = G1.62 GC = (Vc, E, Sc, Pc) where , VC = VIUV2 USC Z = I/p alphabet Sc = start symbol of Gc Pc = PIUP2 U {Sc-> S1.52} Now, consider one example Let context free grammar on speciafied as A -> a Ab lab . This grammar generate CFL Li defined as Li= { am bm / m>13 Another CFG G2 specified as to (cn dn/ n>13 upon performing concatation of G2. G1 the resultant grammer ac becomes Gc = G1. G2: GC: 5c - A.B A - aAblab



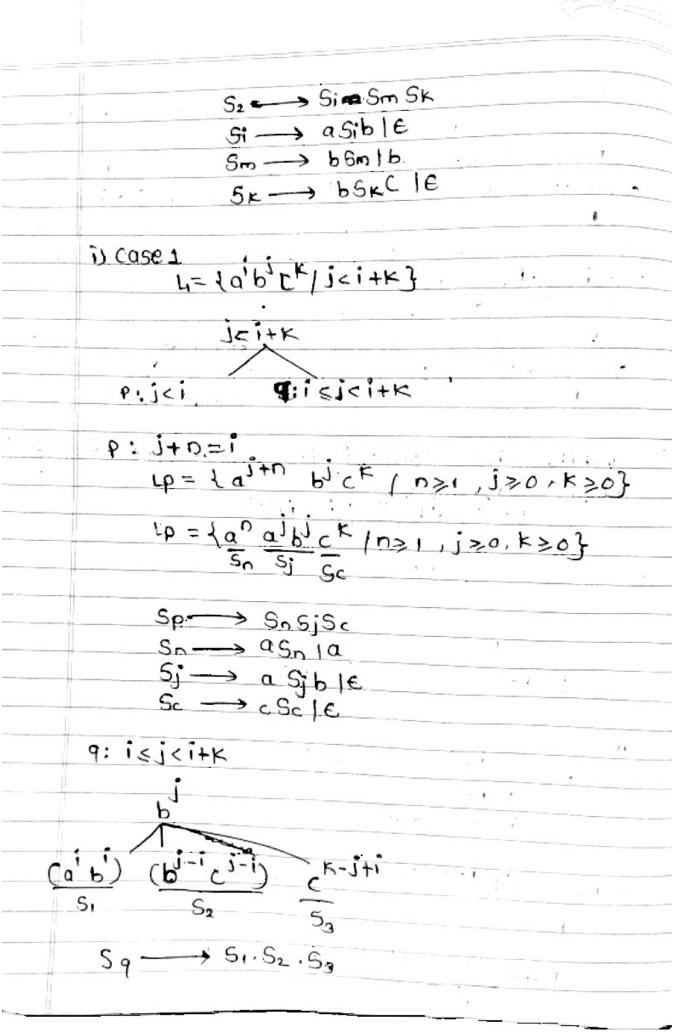
B-> cBd/cd where, vc= {Sc, A, B} Z = { a, b, c,d} Sc = { Sc} Pc = PIUP2U (S2-3) kleen closure operation .like union, concatation it is quite possible to perform kleen closure operation of context free grammer. let G is CFG defined as $G = (V, \Sigma, S, P)$ This CFG 'G' generates o CFL'L'.

0, it is quite possible to perform klee closure operation on given grammer G. upon performing the kleen obsure on G pivour purely context in nature.

Let's all is grammar as Gr GK can be defined by VK = VUSK
where . VK = set of all variable of GK obtained by VK = VUSK. ZK= Z Sk = Start symbol of proposed grammar GK. Pk = set of all the produ of grammer Gr Objatined as PK = PU (SK -> SSKIEZ



| ===== | |
|-------|---|
| | for example, consider following grammar |
| | $A \longrightarrow ab$ |
| | This grammer is generating a language |
| | containing only two string fails |
| | containing only two string (a,b). If we perform kleen closure on above |
| | grammar, the proposed graman or can be |
| | rewrittens as |
| | rewritten's -> ASIE |
| | Now, this grammar generate |
| | a universal language over laibs |
| | Now, this grammax generate a universal language over (a,b) This language denoted as L* |
| 0.51 | Write the context free Grammare for Follo- |
| ענייט | Write the context field diditione |
| | wing language, L= {aibick / j = i+k} |
| | r= fasc. (1+1+15) |
| _ | j = i + F |
| | |
| | |
| | casel case 2 |
| | jci+k j>i+k |
| | |
| | ii) case 2 |
| | L2 = {a'b'c' j>1+K} |
| | J>i+K |
| | j=1+K+m |
| - | (i, (i+k+m) b |
| | : L2 = { aib(i+k+m) ck / ixo, m>1, K>03 |
| - | Rewrite |
| | L= {a'b'bmbkck i>0, m>1, k>0} |
| - | Si Sm Sk |
| \ | OI ON) UK |





| 5, a 5, b lab |
|---|
| $62 \longrightarrow bS_2 C \in$ |
| $S_3 \longrightarrow CS_3 \mid C$ |
| , |
| The grammer for language li can be |
| designed performing union on respective grammers for the language Lp or Lq. i.e. proposed grammer should have capability to unite the grammar for Lp as well as Lq. |
| grammers for the language up or iq. |
| i.e. proposed grammar should have capa |
| bility to unite the grammar for up as |
| well as 19. |
| ist production should be |
| 1st production should be |
| $S_1 \longrightarrow Sp(Sq)$ |
| $5p \longrightarrow Sn SjSc$ |
| $s_n \longrightarrow a s_n a$ |
| sj> asjbje |
| $Sc \longrightarrow cSc 16$ |
| $Sq \longrightarrow S_1 \cdot S_2 \cdot S_3$ $S_1 \longrightarrow aS_1b \mid ab$ |
| 52 → b52c € |
| 53 → cS3 C |
| S S.152 |
| $51 \longrightarrow Sp \setminus Sq$ |
| $sp \longrightarrow sn sisc$ |
| $S_n \longrightarrow aS_n 1a$ |
| Sj -> aSjble |
| Sc → cSc € |
| 5q S1.52.53 |
| Si - a Siblab |
| S2 -> bSc 16 |
| $53 \longrightarrow cS_3 C$ |
| |

