

Compiler Design

Assignment-2

Note: Submit it on whatsapp group created for compiler design latest by 07.04.2020

1. Calculate and write all existing lexems and different token in the following statement

Printf(“total=%d\n”, score);

2. Consider the following grammar. Compute FIRST & FOLLOW sets for each non-terminal. Check whether the grammar is LL(1).

S-> ACB/CbB/Ba

A->da/BC; B->g/ε; C->h/ε

3. Compute FIRST and FOLLOW for each non-terminal in the grammar. Check the grammar is LL(1) or not.

S->aABb, A->ε, B->d/ε

4. Construct the following grammar to an unambiguous grammar. Check whether the resulting grammar is LL(1) or not.

R->R+R/ RR/ R */ (R) /a/b

5. Check whether the following grammar is LR(0), SLR(1)

E1 ->E #,

E-> T+E / T

T-> x

6. Check whether the following grammar is LALR(1) & LR(1)

S->AA, A->aA/b

7. Design SLR parser

$E \rightarrow E + E$

$/ E * E$

$/ id$

8. Design LR (0) parsing table:

$E1 \rightarrow E \#$

$E \rightarrow E + T / T$

$T \rightarrow i$