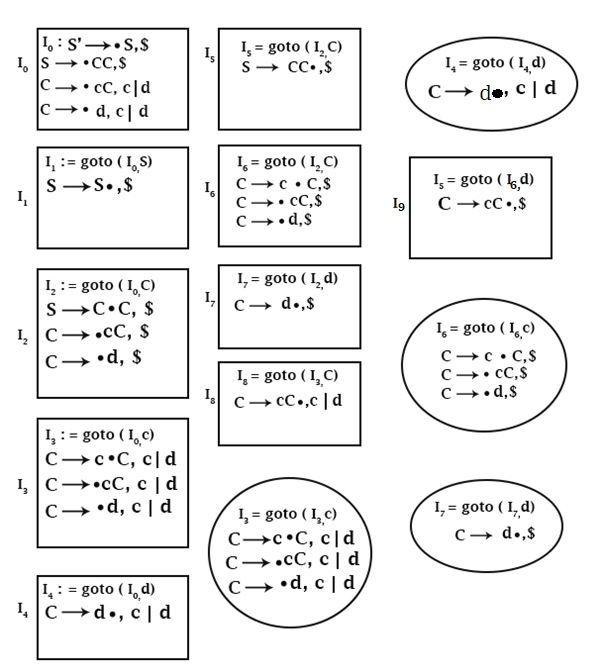
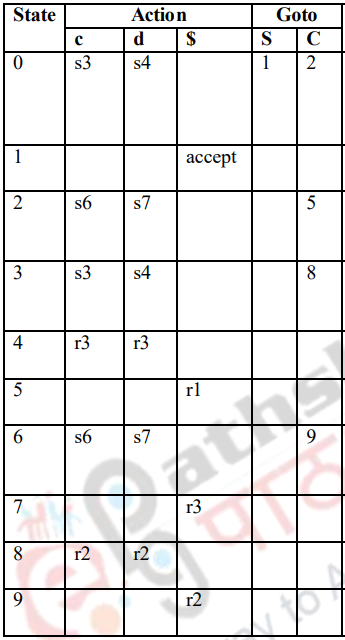
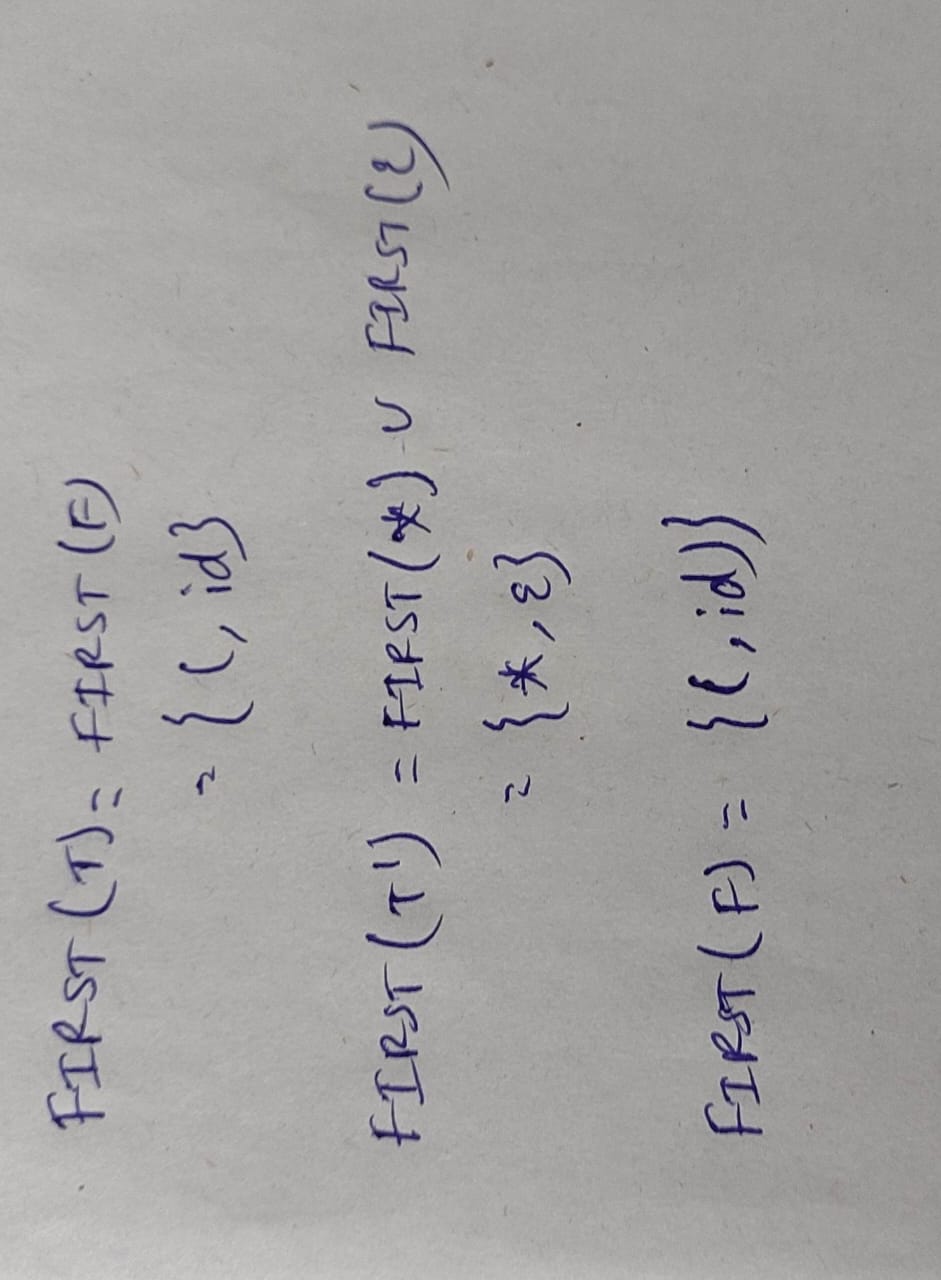
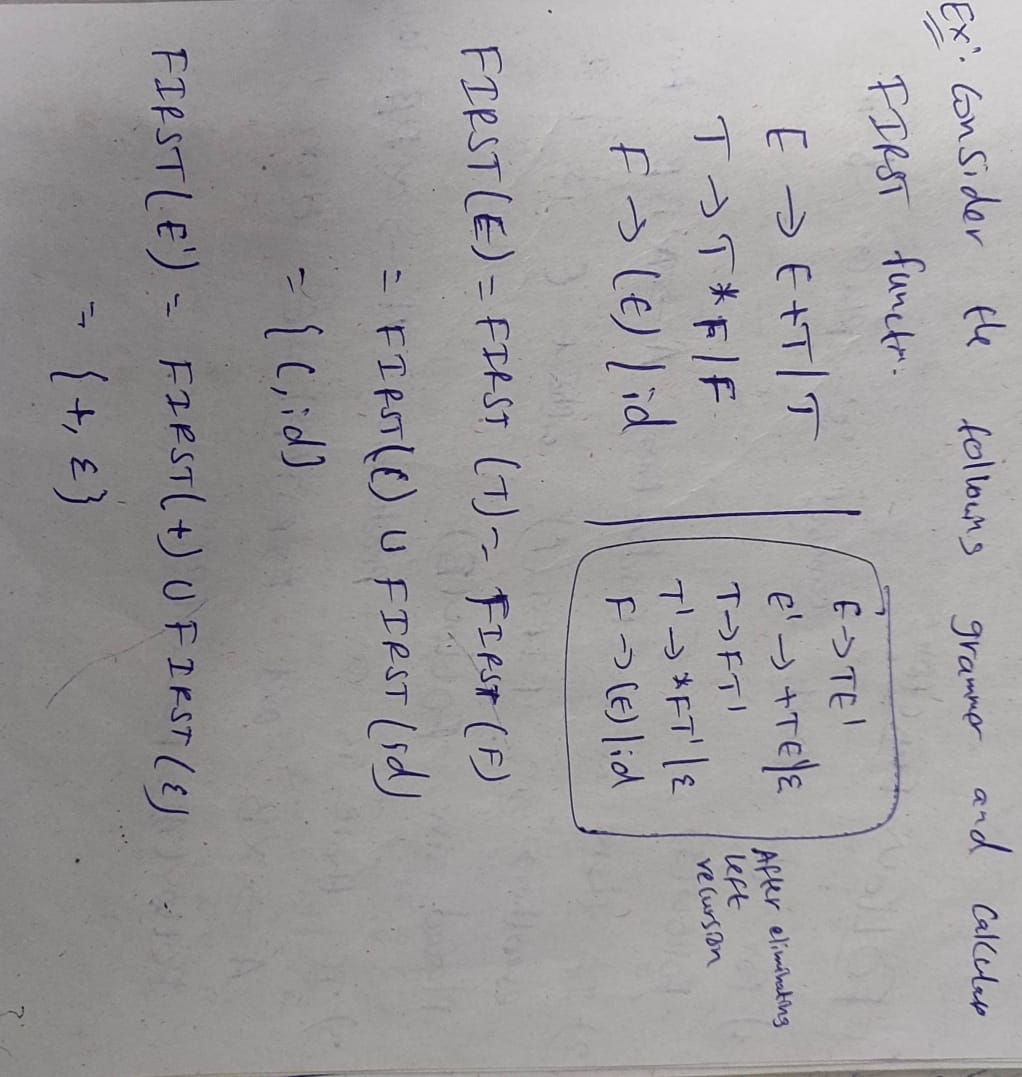
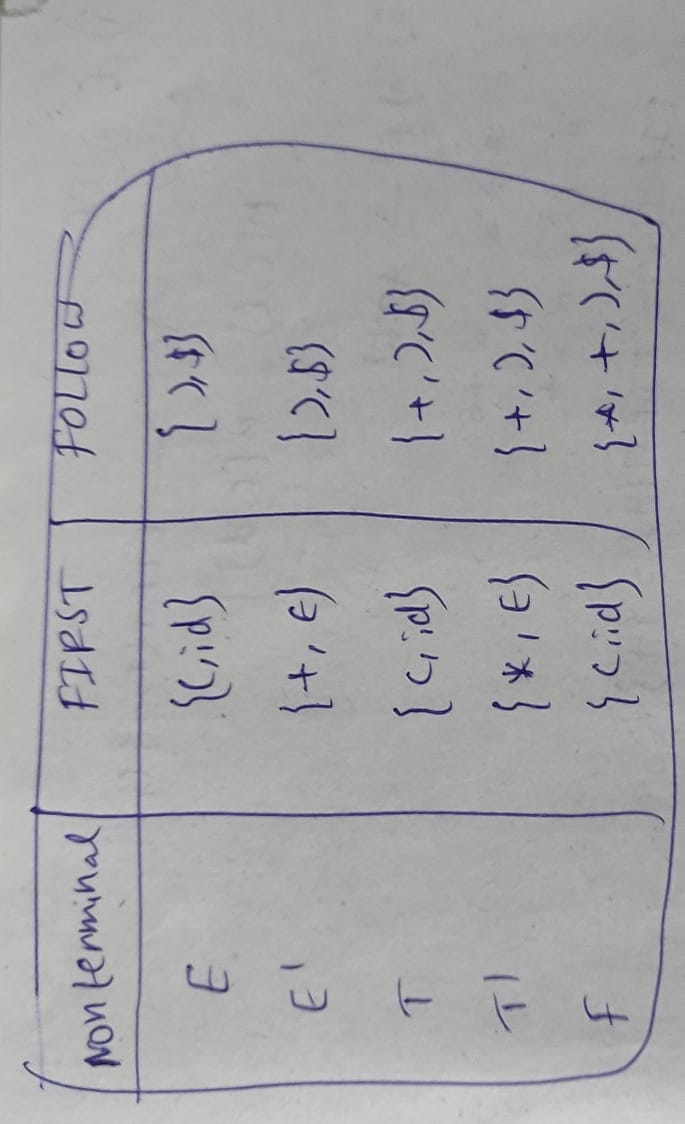
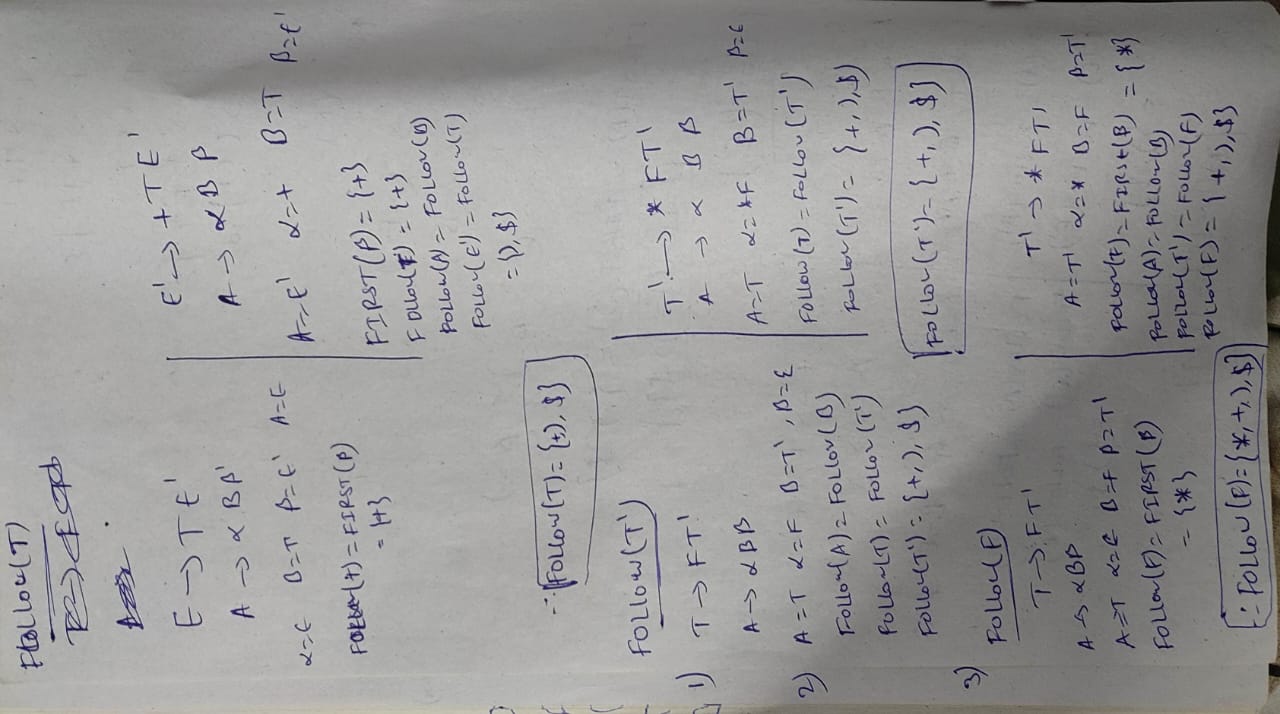
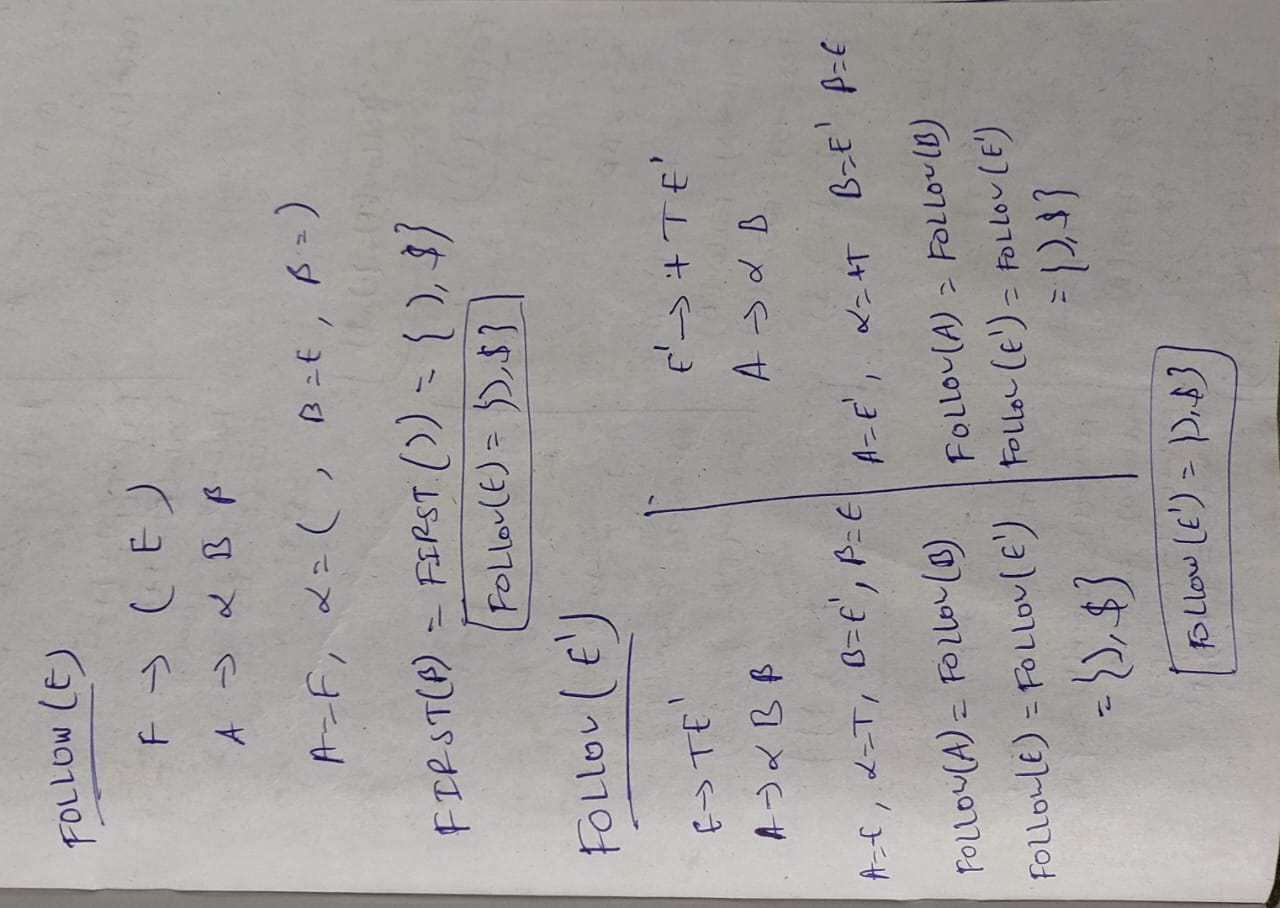
**1)Construct CLR parsing table for the grammar S->CC C->cC | d.**



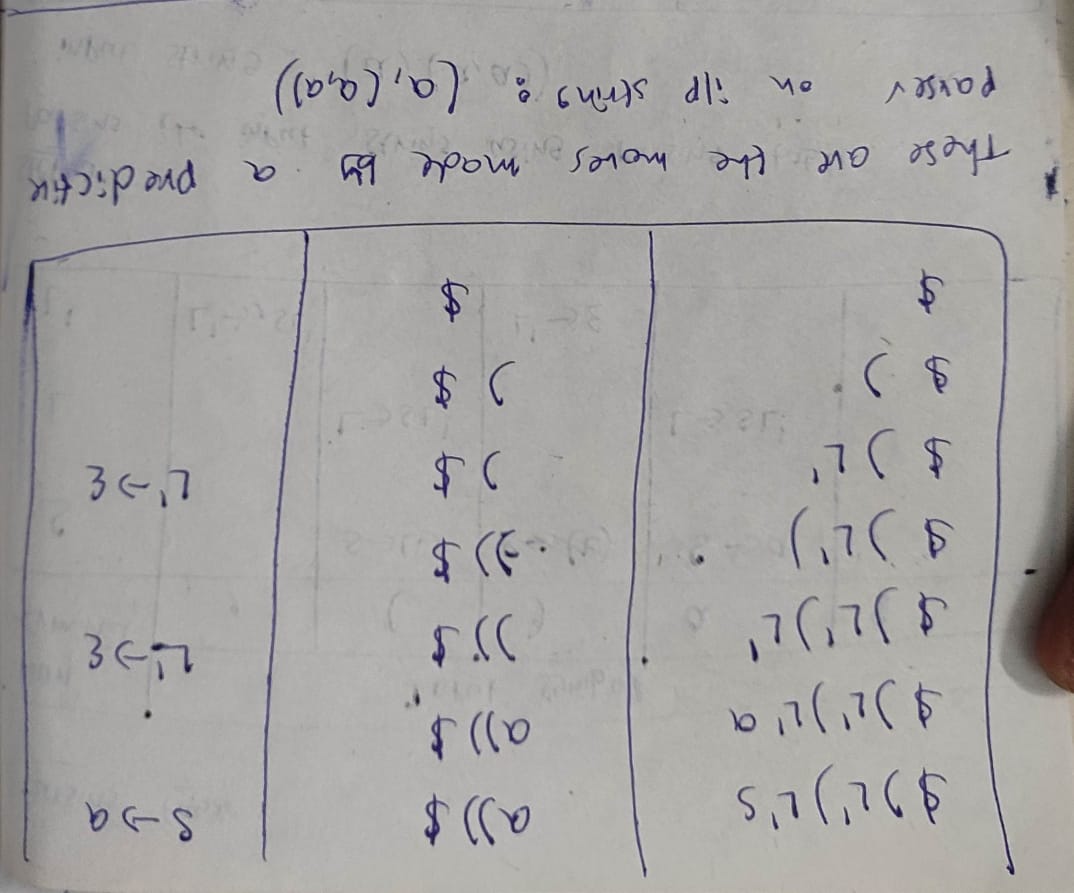
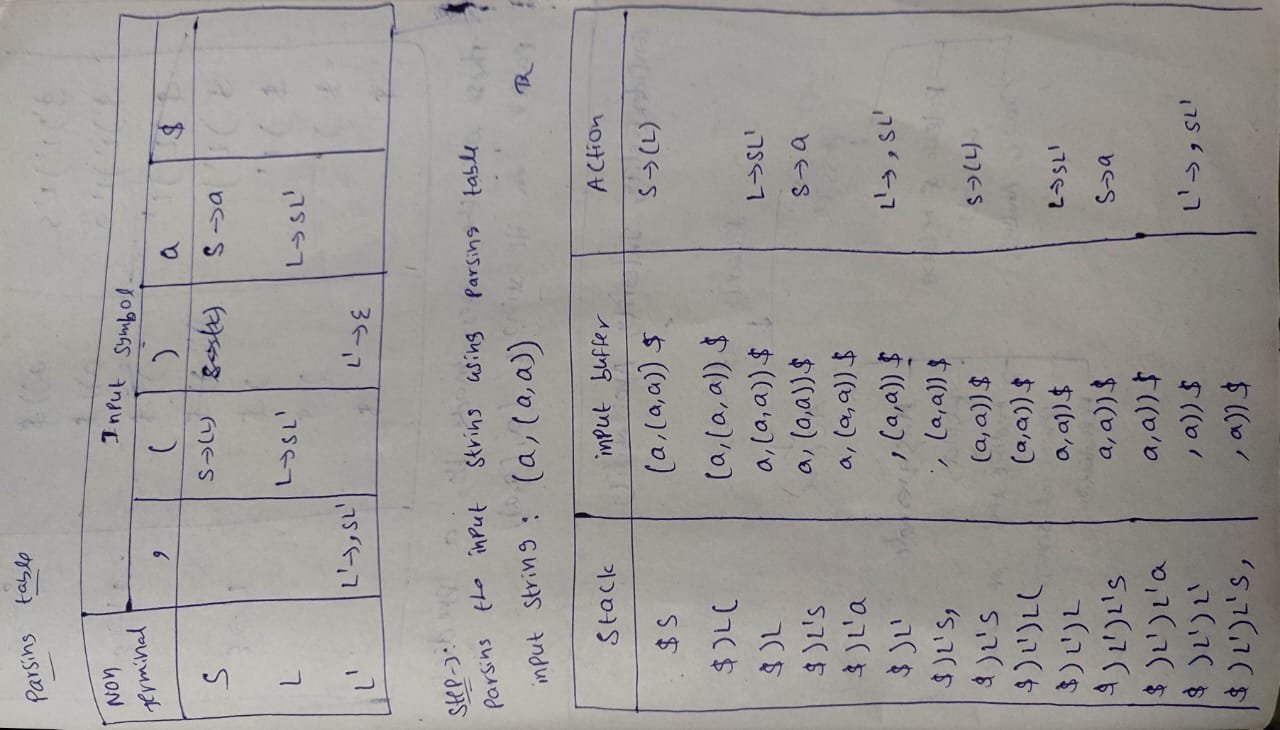
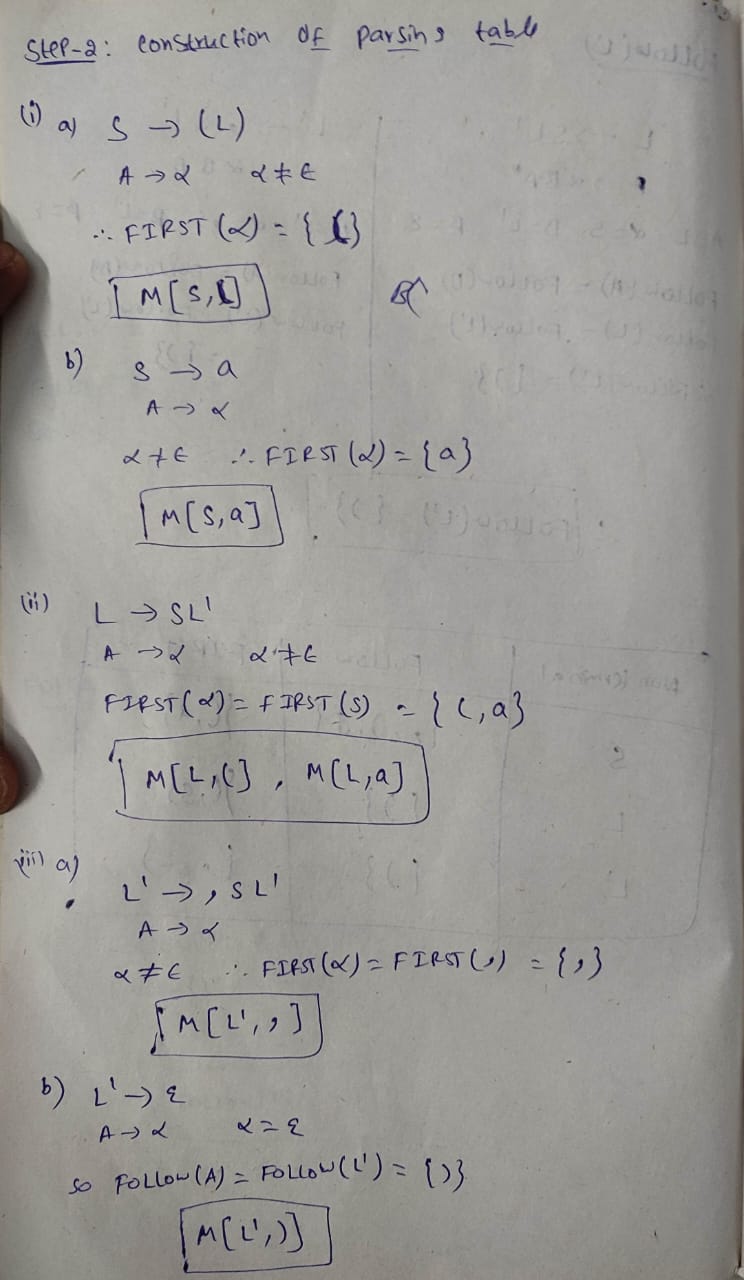
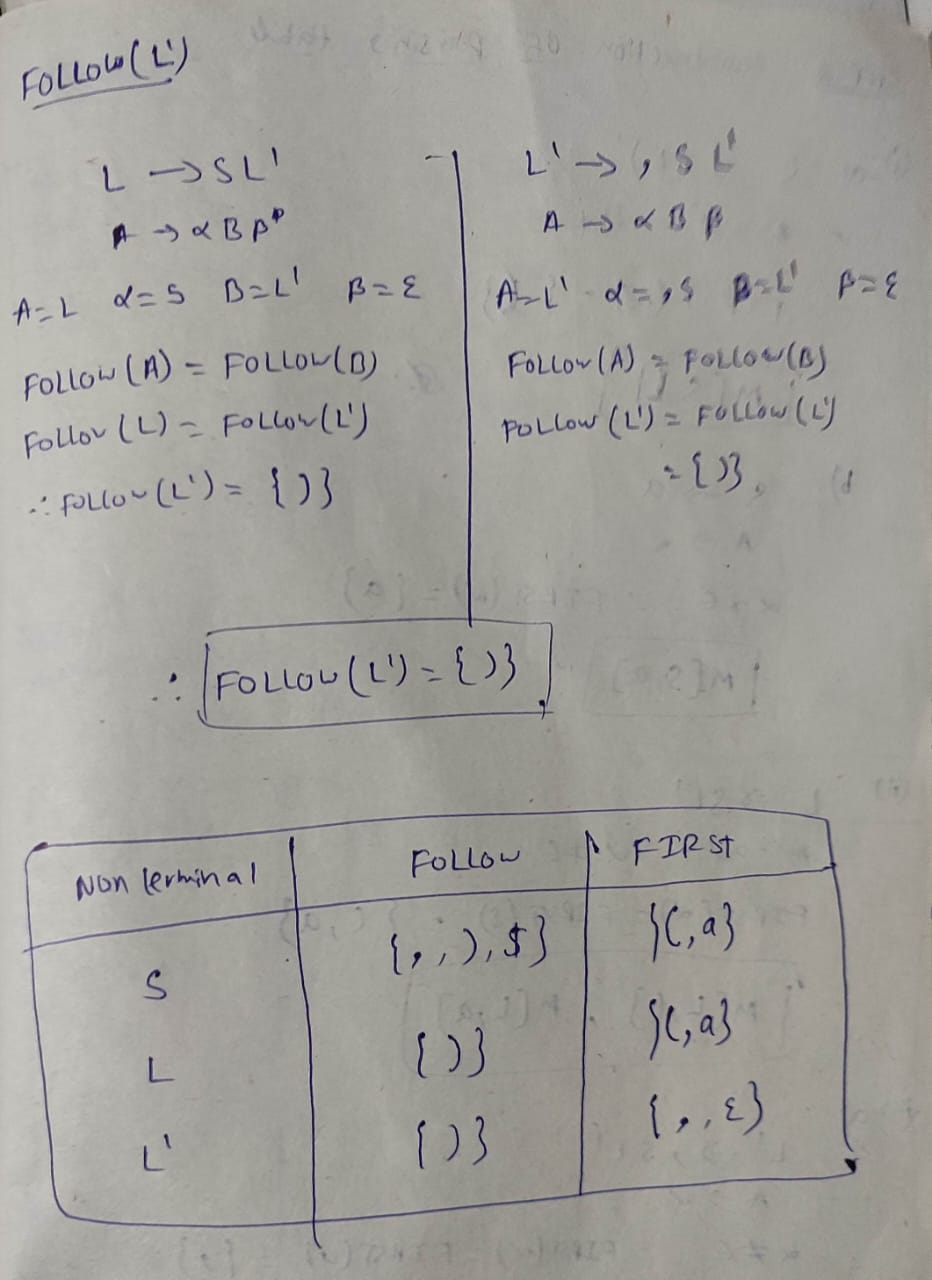
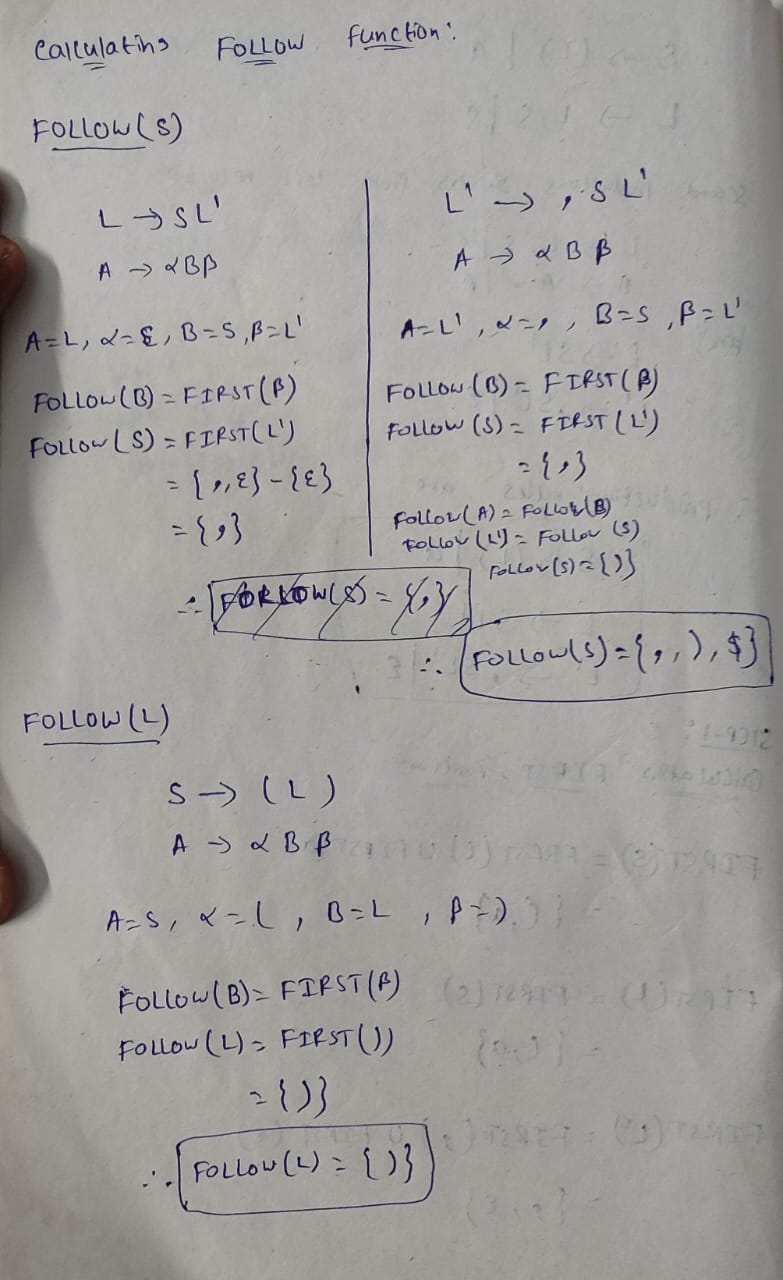
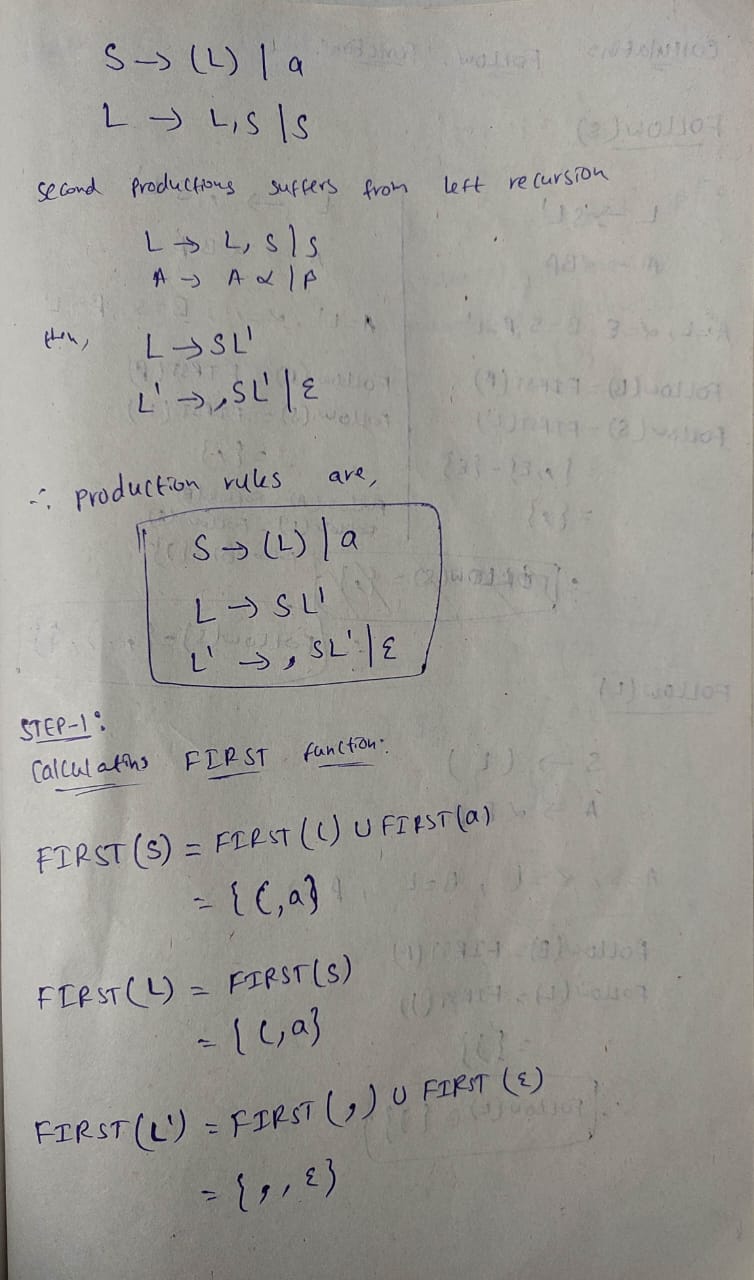


**2) b) Compute First and Follow for the grammar E->E+T/T, T->T\*F/F, F-> (E)/id.** 



**3) a) Illustrate recursive descent parser for the grammar. S->cAd A->ab|a with a string cad**

**3) b) Construct the LL (1) for the following grammar S–>(L) | a L–>L, S | S**



**5)a) Construct predictive LR parsing table for the grammar S->iEtSeS | iEtS |a E->b.**

**5)b) Explain in brief about the shift reduce parser with example.**

EXAMPLE:

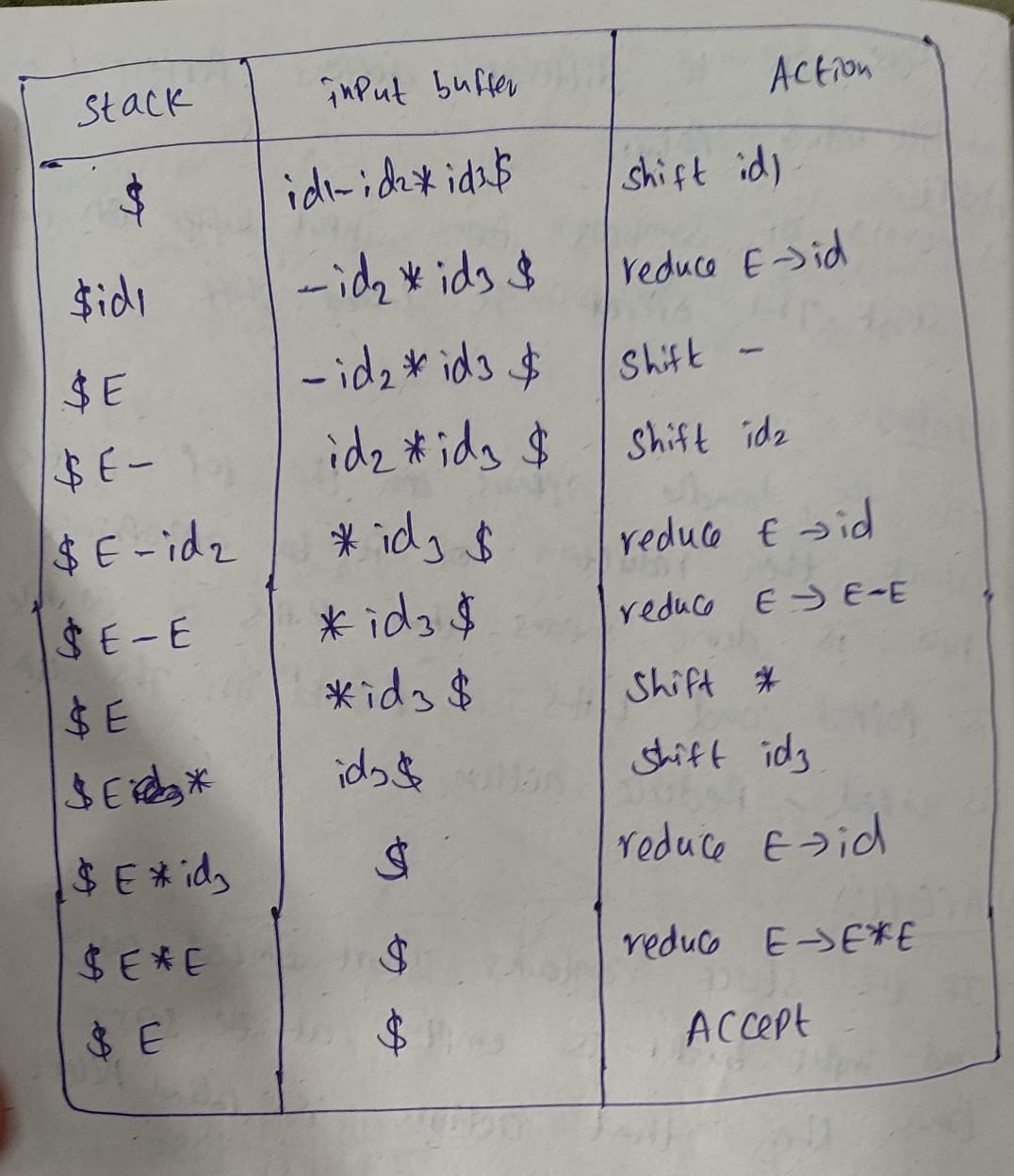
Consider,

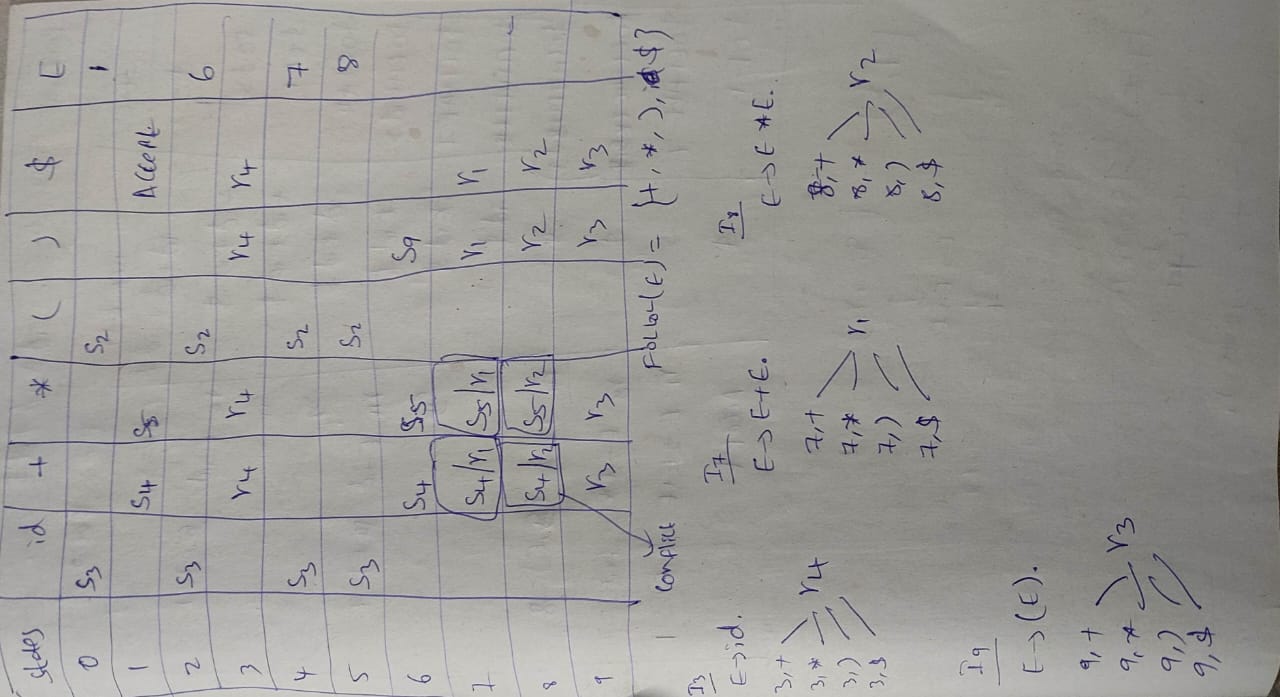
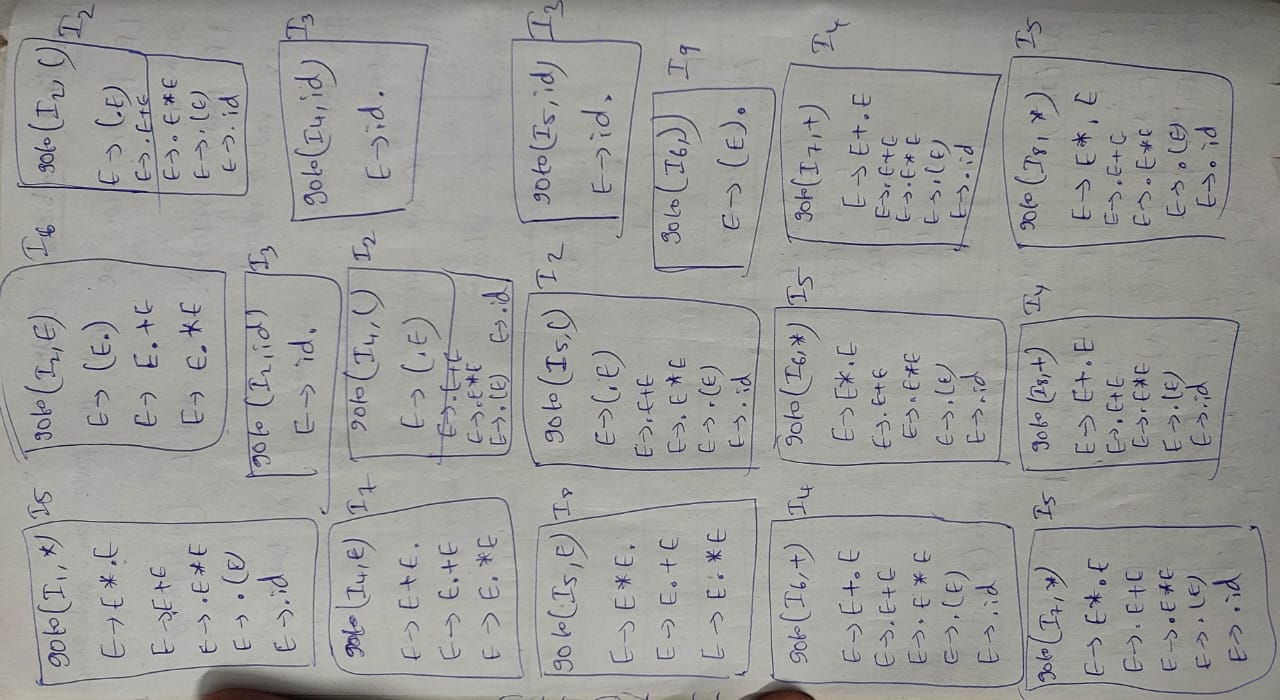
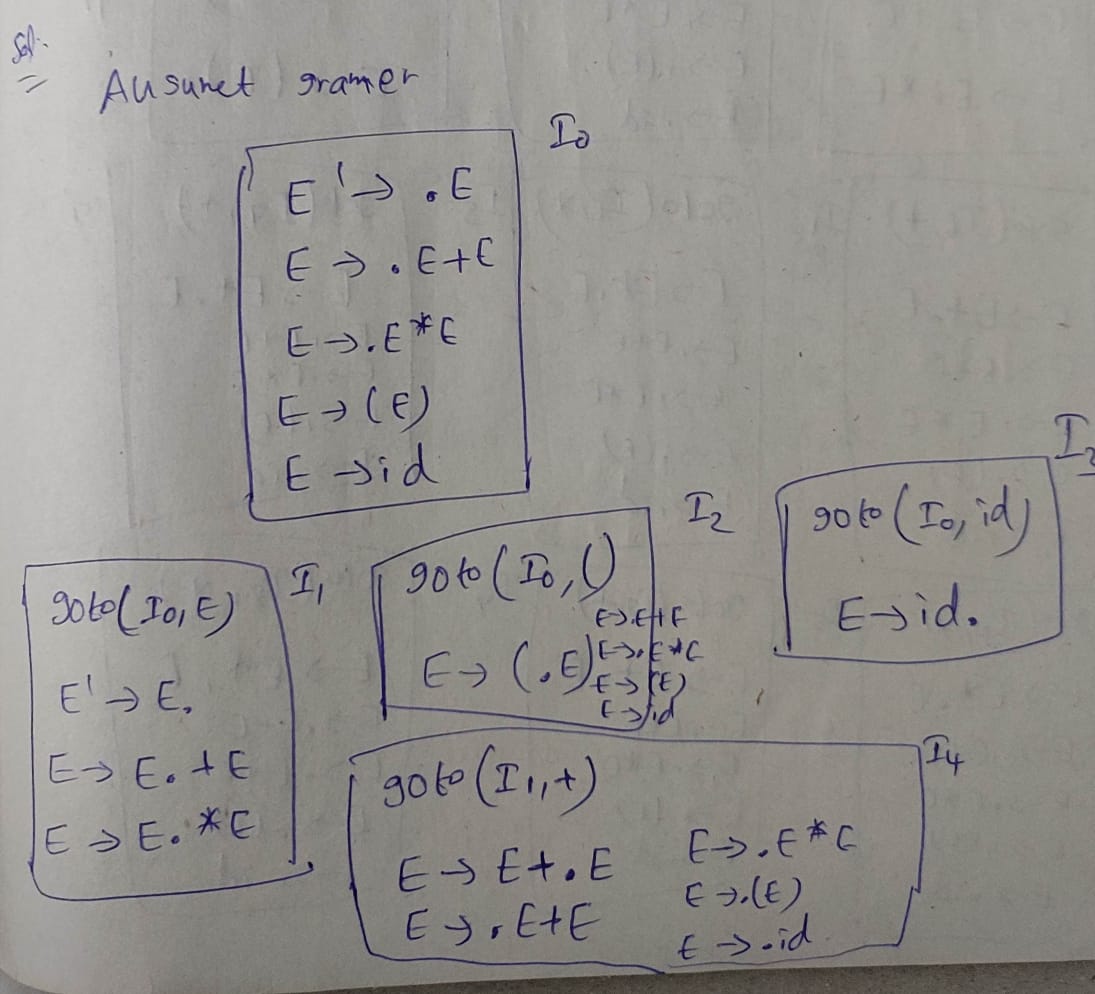
E -> E-E

E-> E\*E

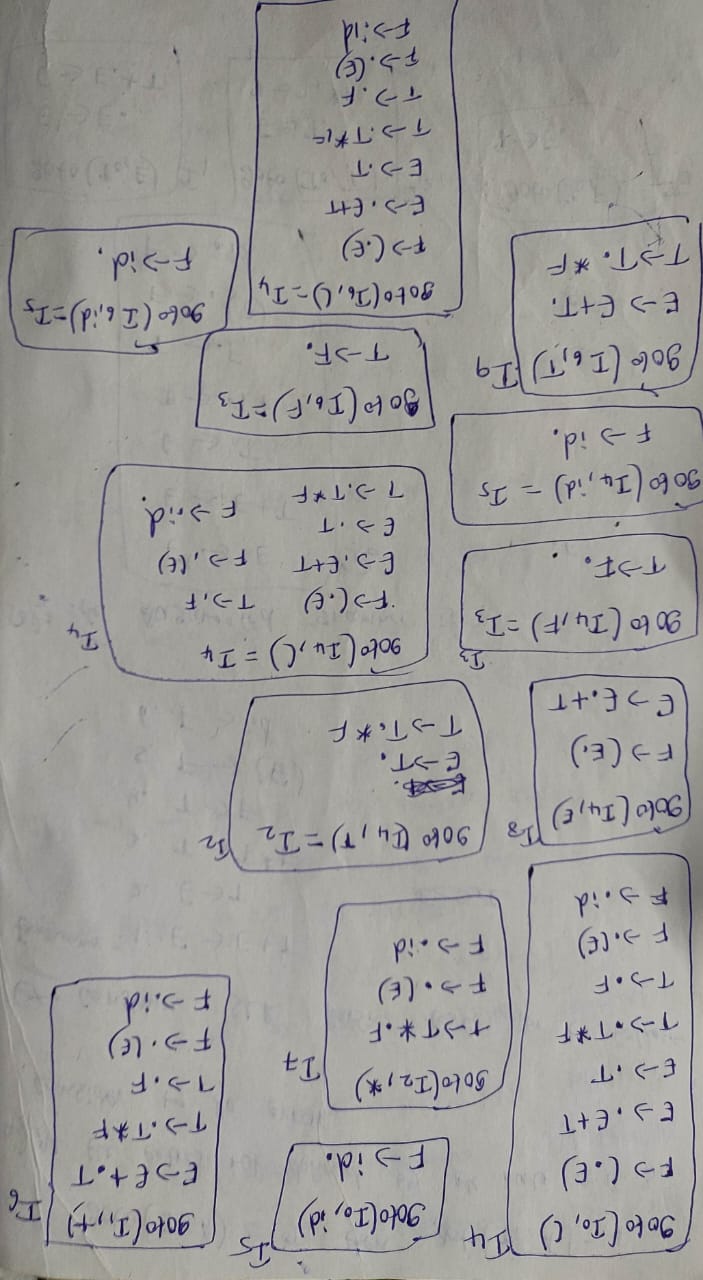
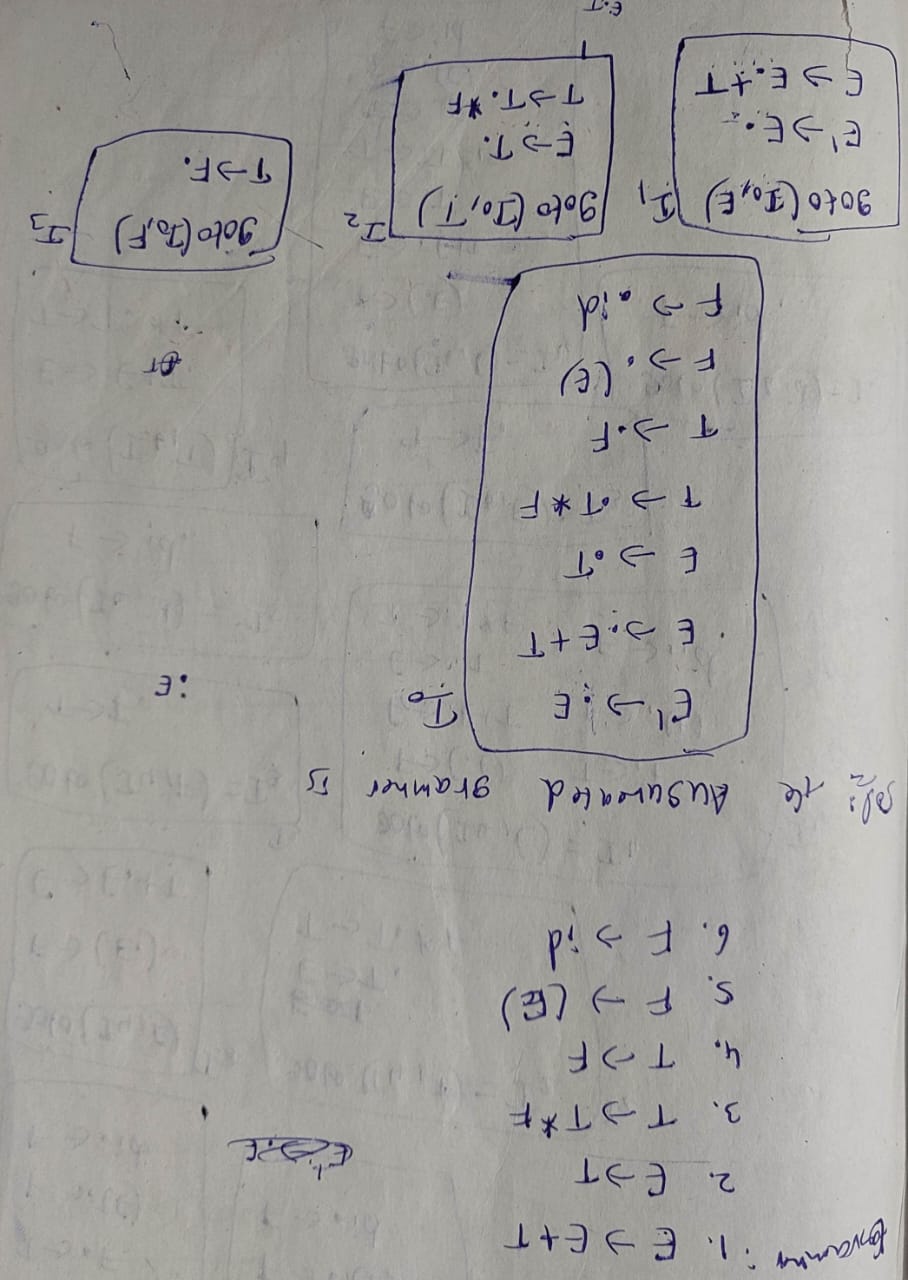
E->id

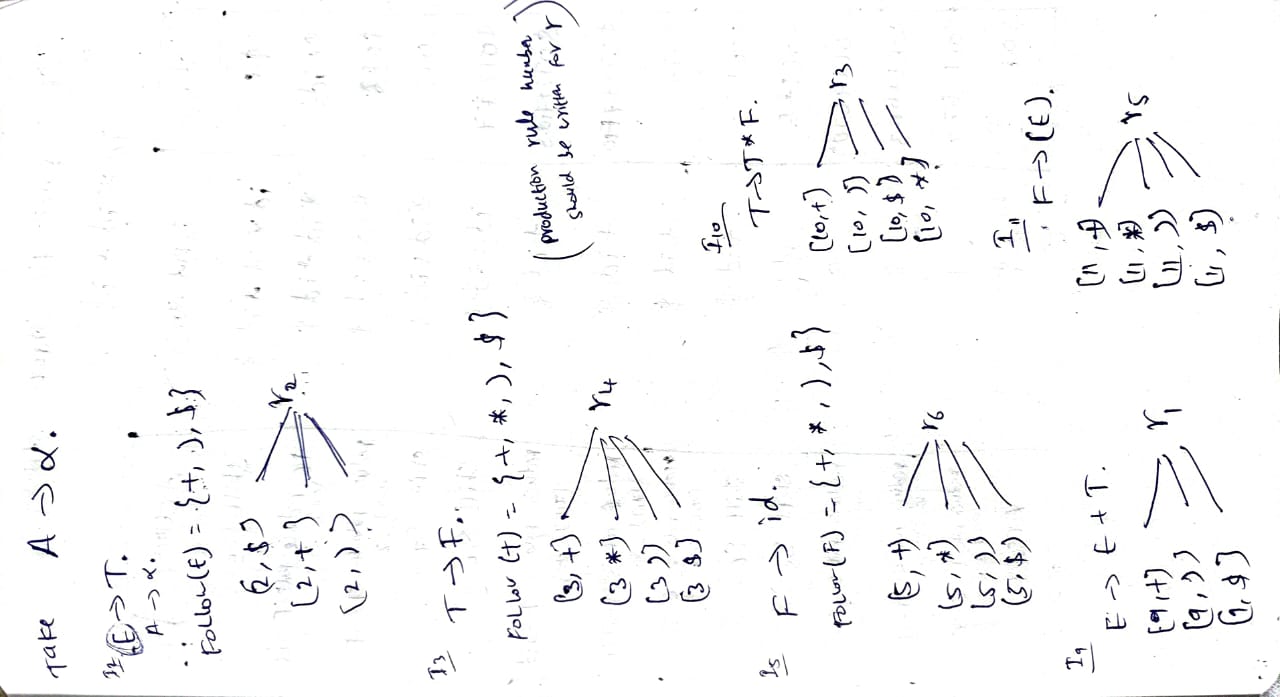
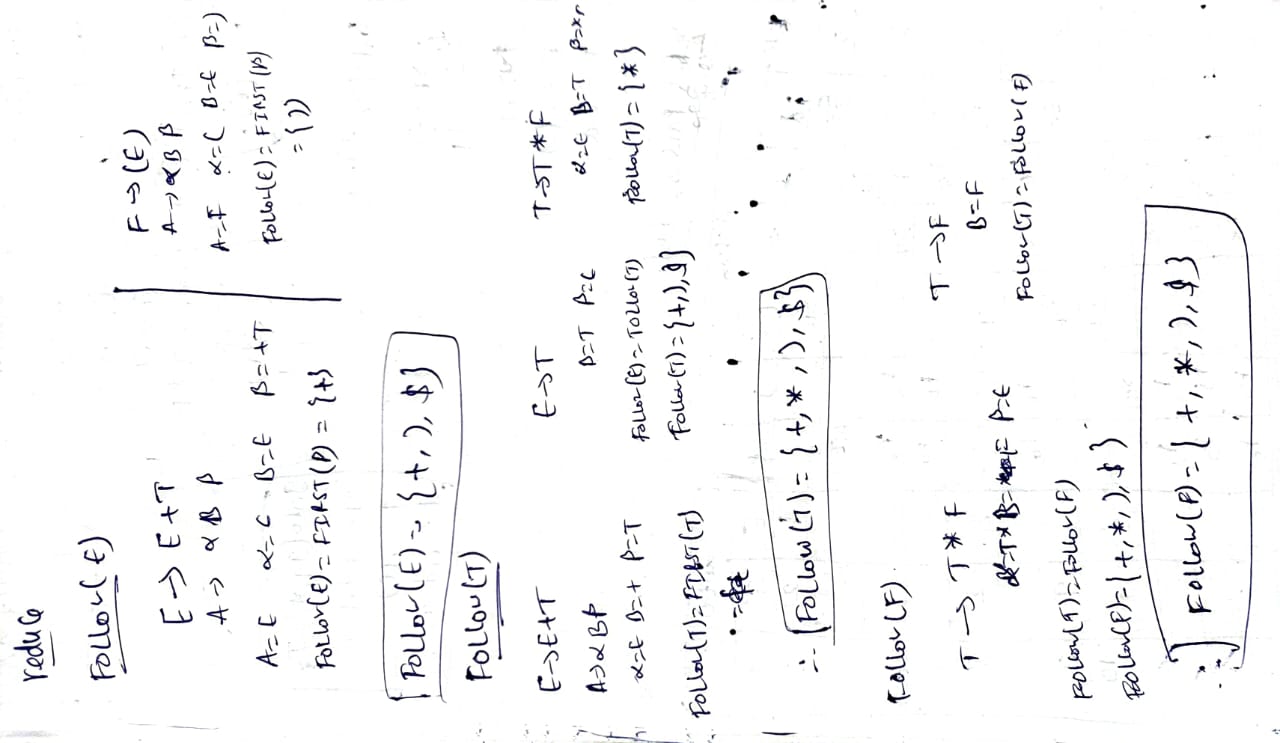
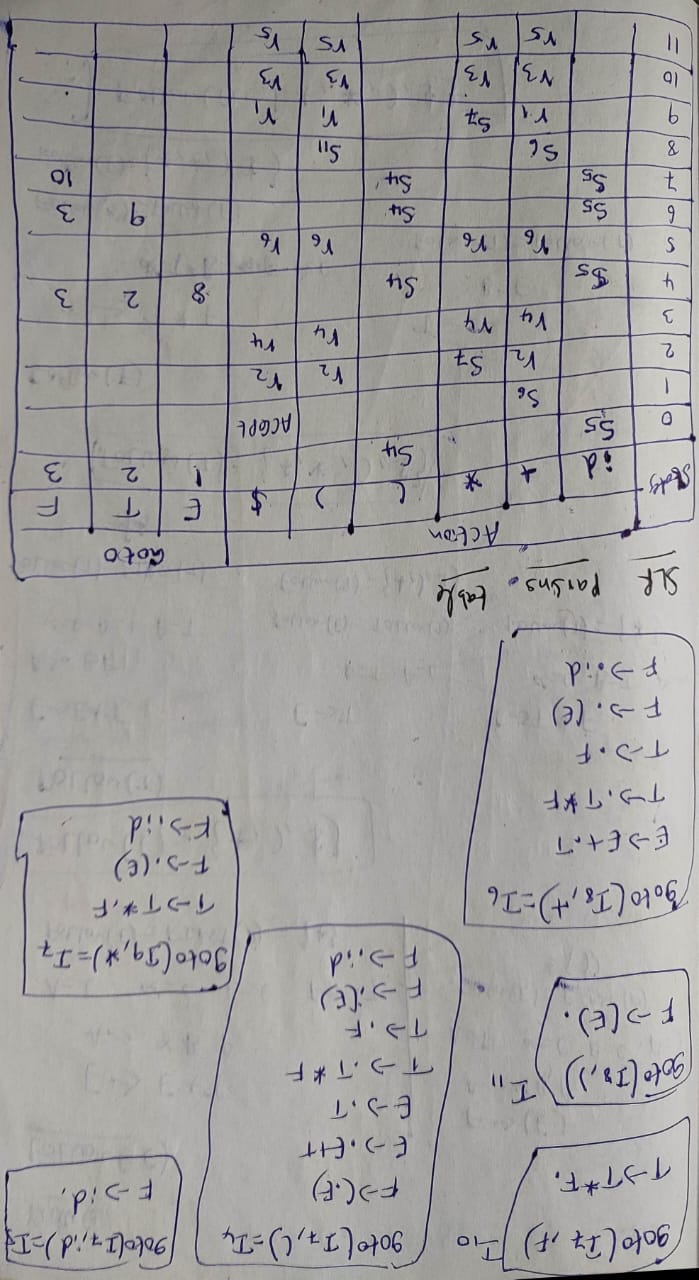
Input string: id – id \* id



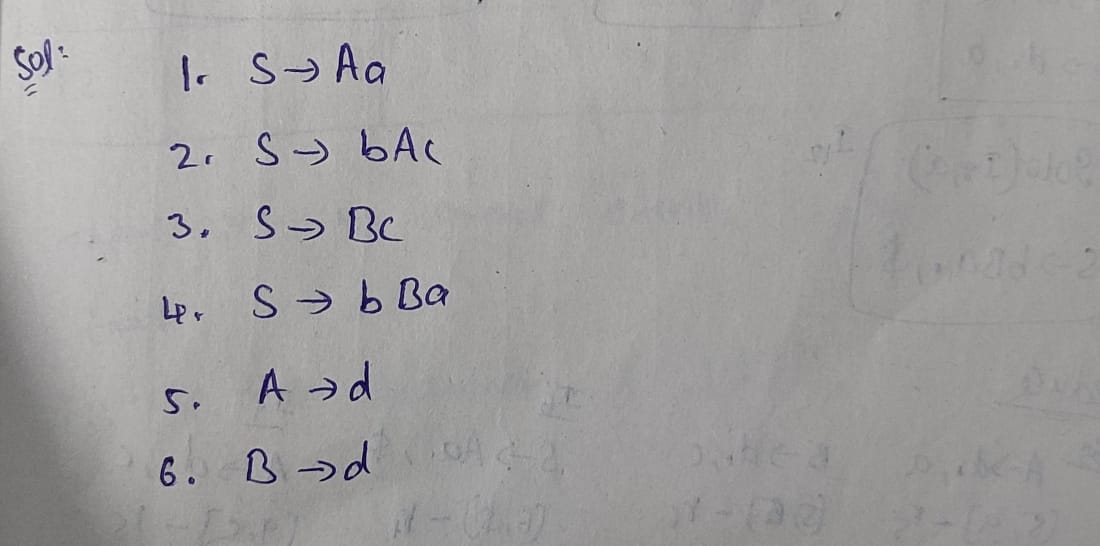
**6) b) Construct the LR parse table for the given grammar. E->E+E E->E\*E E->E-E E-> (E) | id.** 

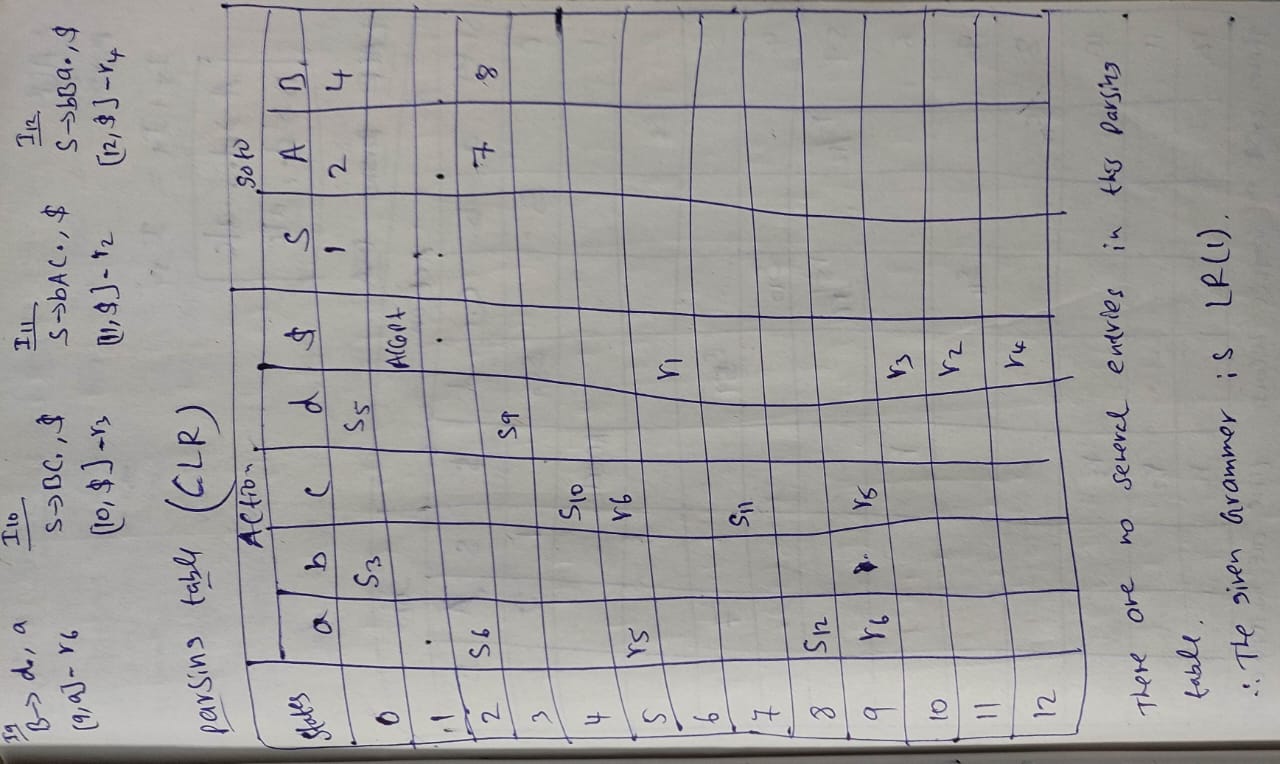
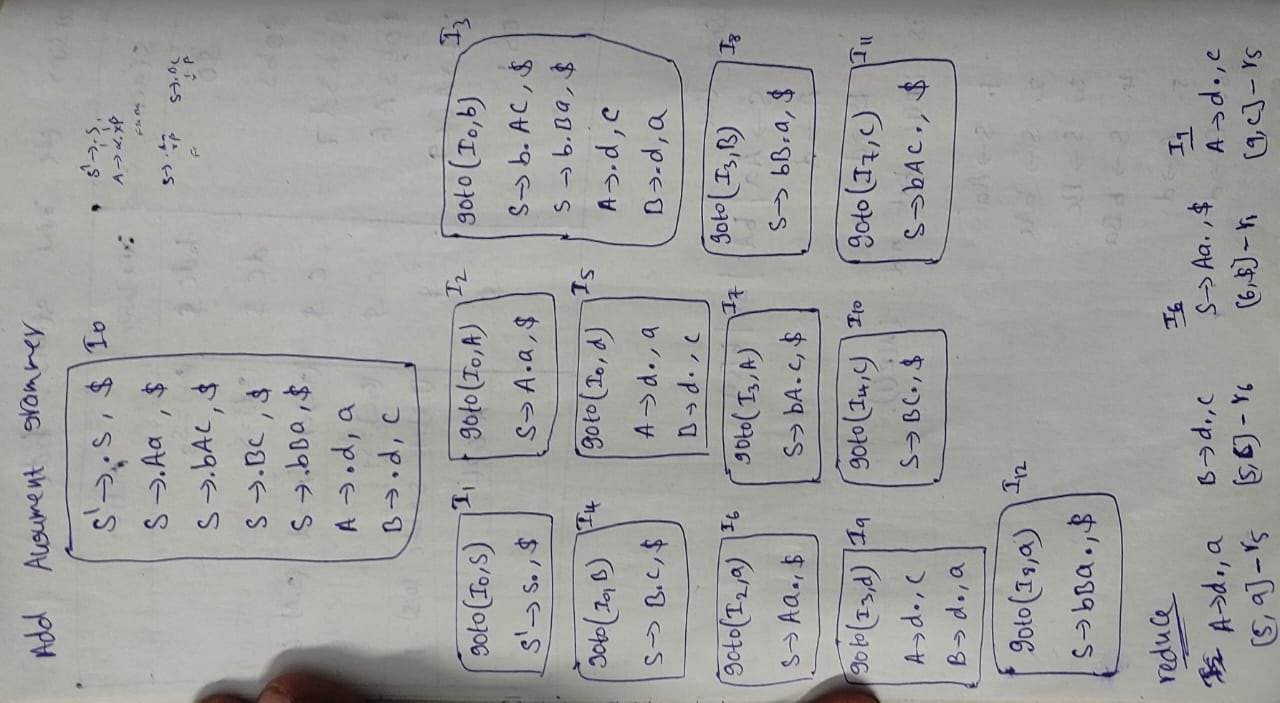
**7)Find the validity of the input string “id+id\*id“by the SLR parser for the given grammar E->E+T | T T->T\*F | F F-> (E) | id.**





**8) Construct LR (1) parsing table. S->Aa , S->bAc ,S->Bc ,S->bBa ,A->d,B->d is LR(1)**





**9) Construct the CLR parsing table for the following grammar: S->L=R | R L-> \*R | id R->L**

