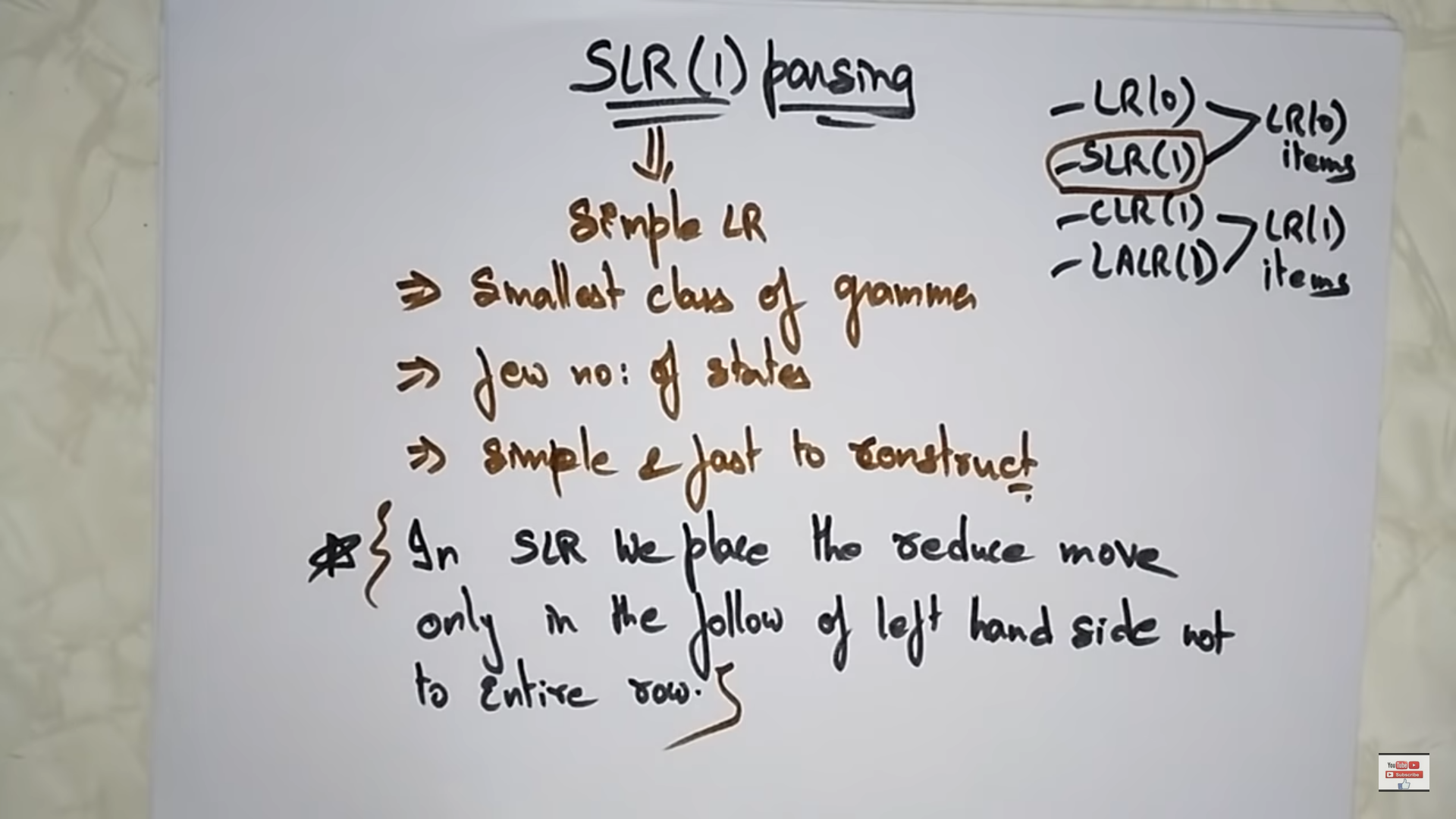
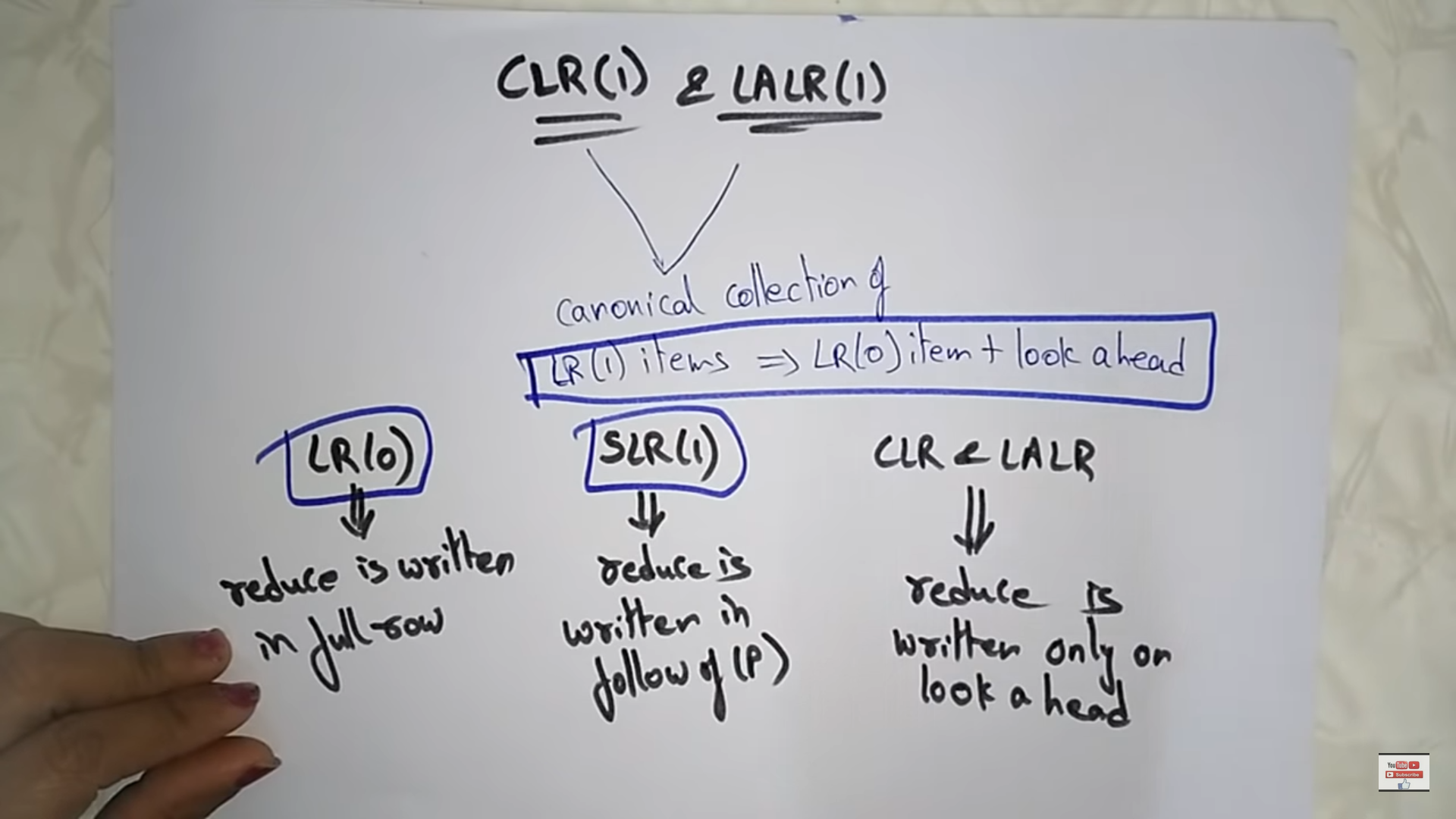
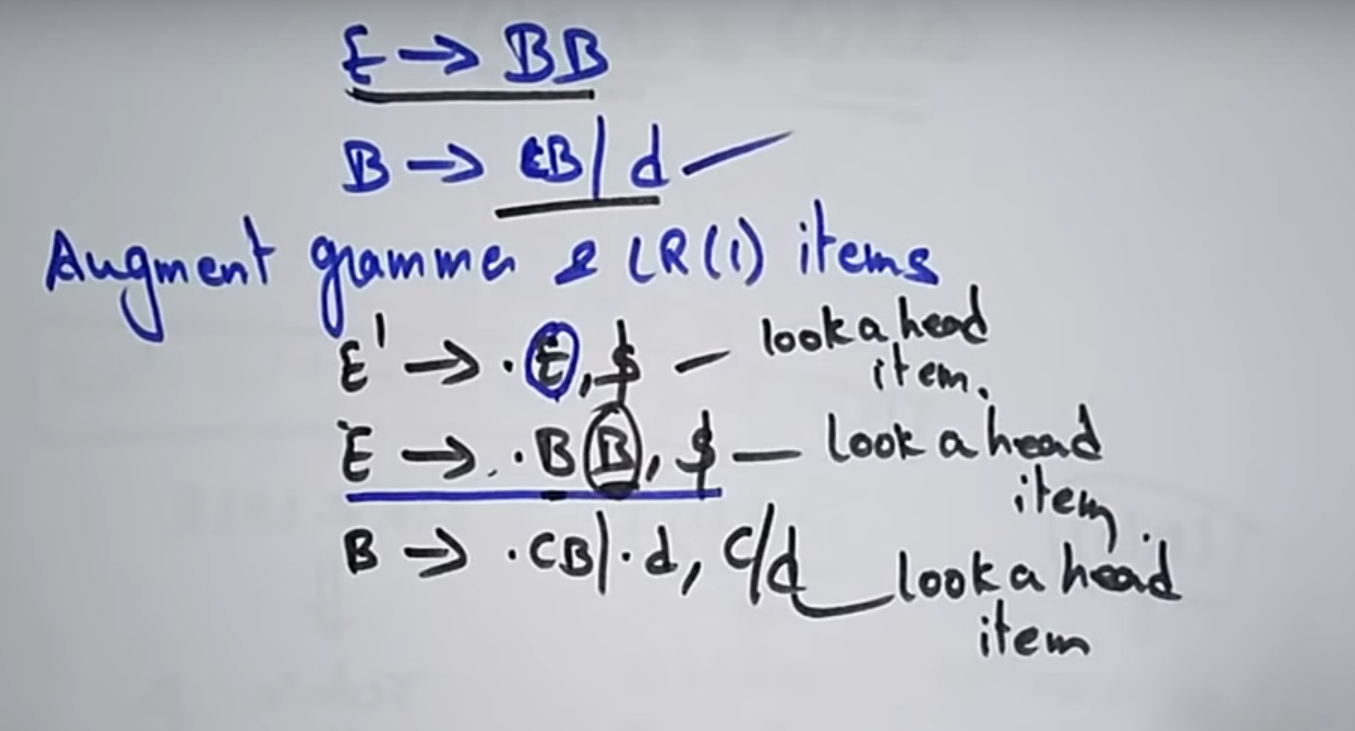
# LR(0) Parsing

# SLR(1) Parsing



# CLR(1) Parsing

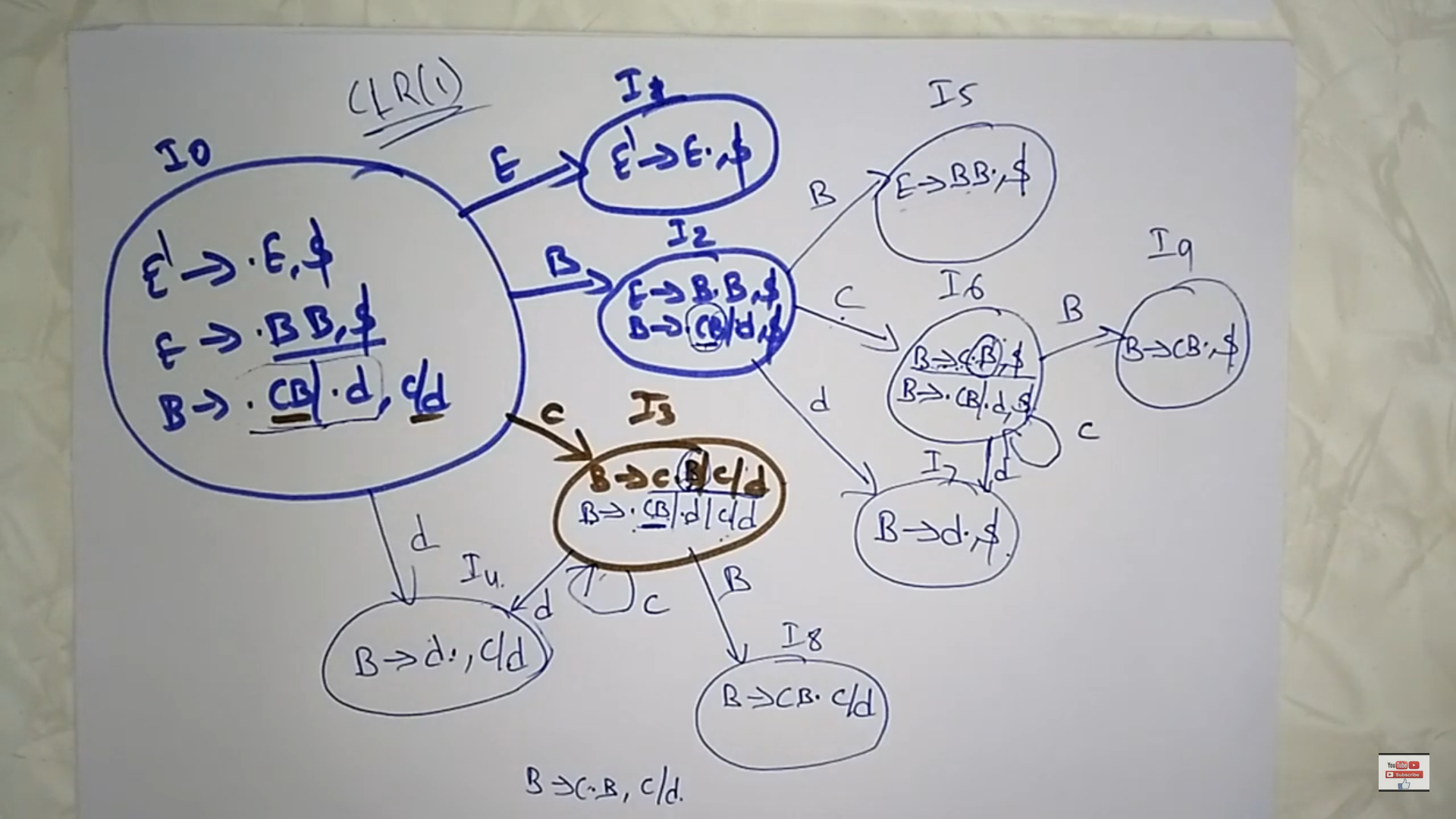


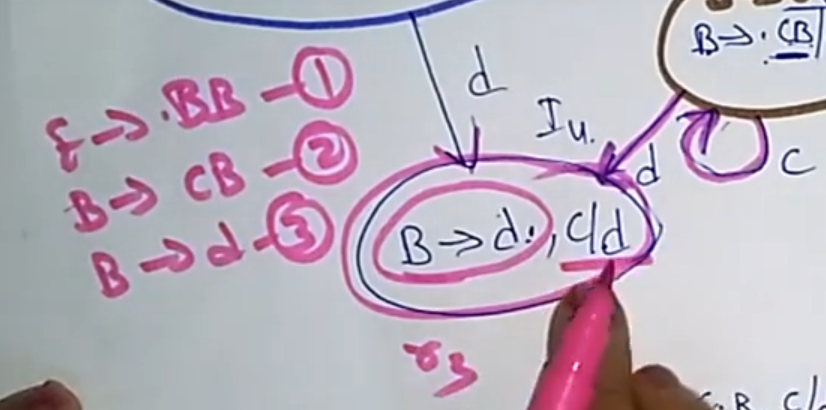


E’ 🡪.E ( augment grammar for LR(0) item)  
E’ 🡪.E,$ (After E there is no more Non-terminal is there, so go to starting of E’ and the starting of E’ is nothing so add $)

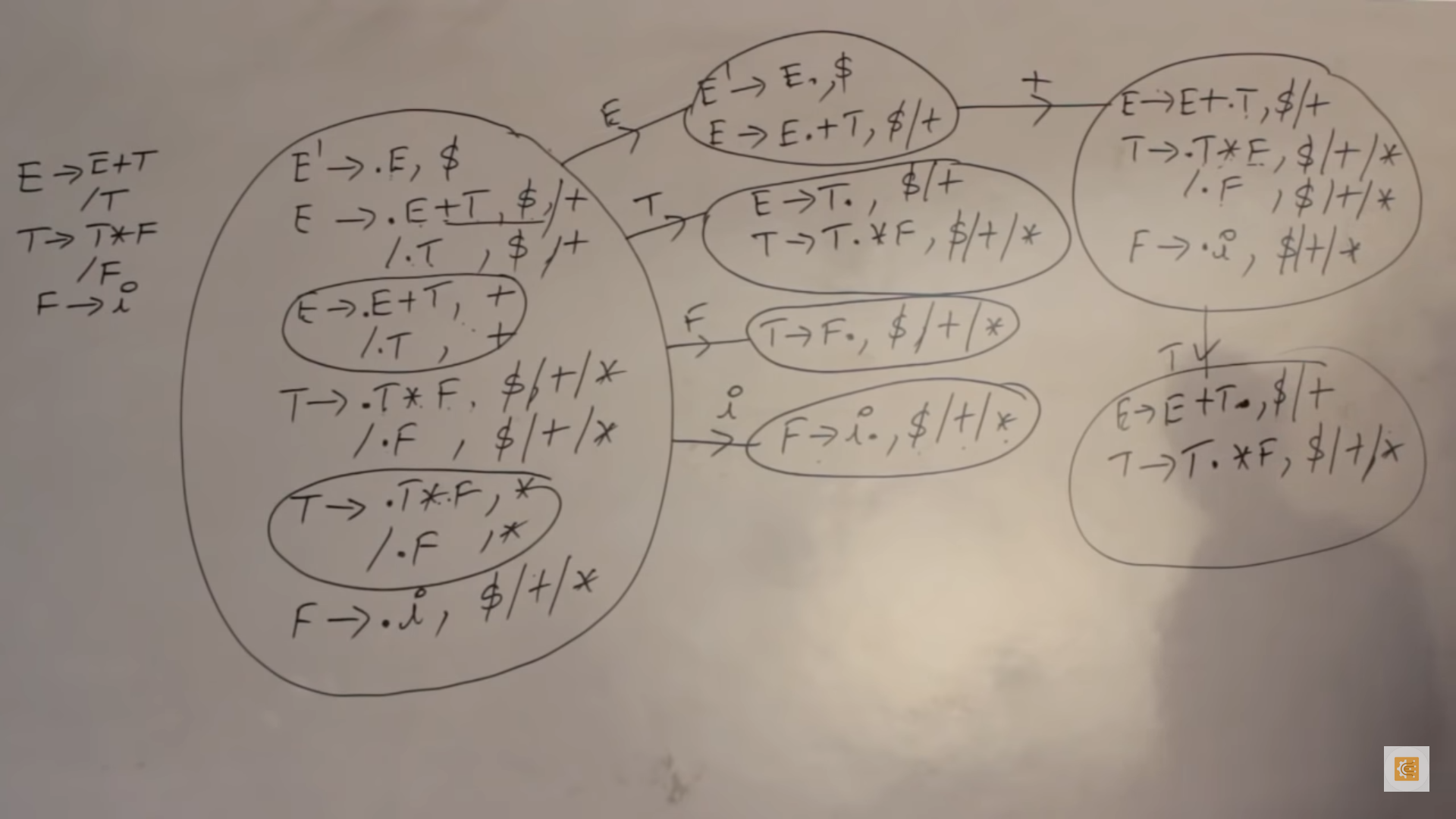
After E, if any non-terminal write all the productions of the Non-terminal  
**E 🡪 .BB , $** ( Just copy the $)

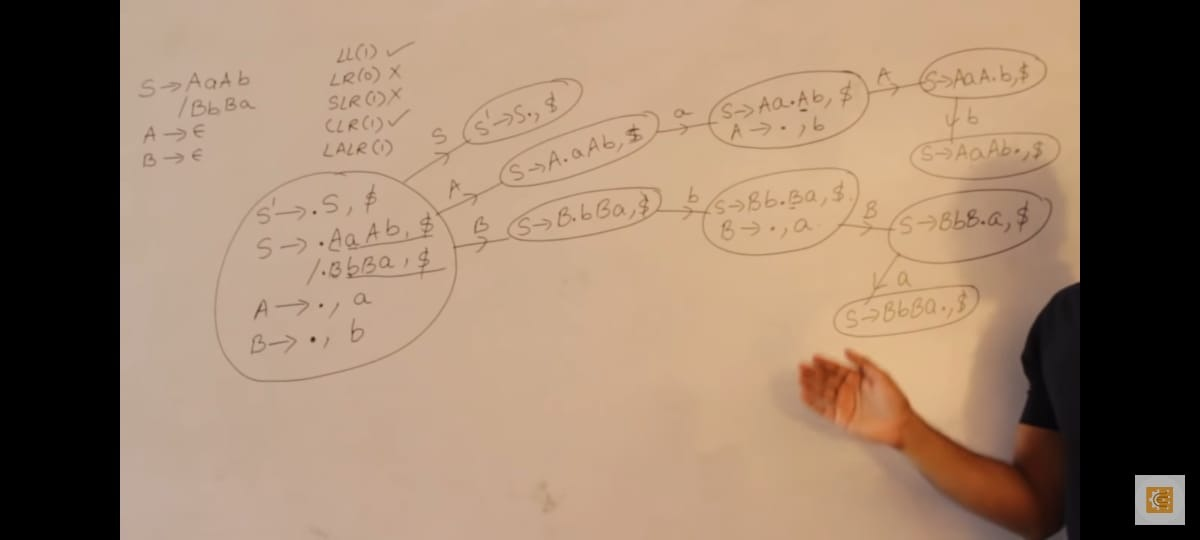
After moving the dot **E 🡪 B.B , $** another Non-terminal is there.   
So write First(B), First(B) = c,d.  
So finally **B🡪.cB|.d , c|d**





In I4 state, Under the look ahead (i.e c and d) write r3.  
In I5 state, Under the look ahead($) write r1.  
In I7 state, Under the look ahead($) write r3.  
In I8 state, Under the look ahead(i.e c and d) write r2.  
In I9 state, Under the look ahead($) write r2.





# LALR(1) Parsing