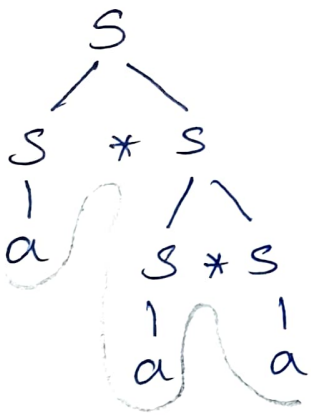
Ambiguity LL(k) grammers:

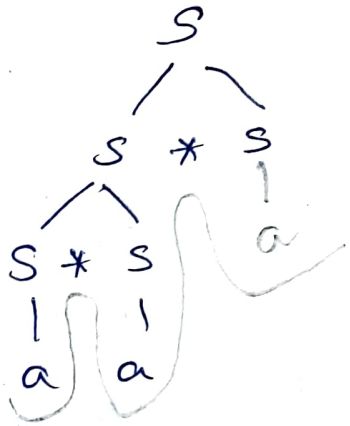
→ If two or more left derivation trees are possible is called ambiguity.

→ One or more than one tree can be generated for the same string.

Ex1: If G1 is the Grammer, S→S+S/S*S/a show that G1 is ambiguous with the string

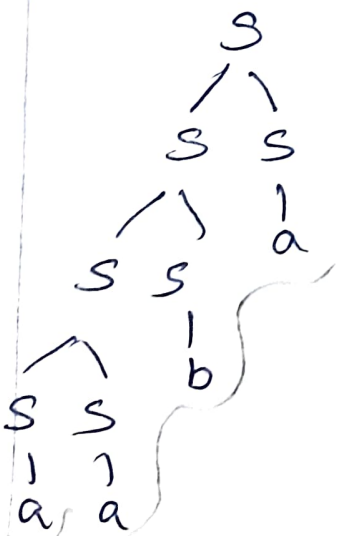
w = a * a * a.

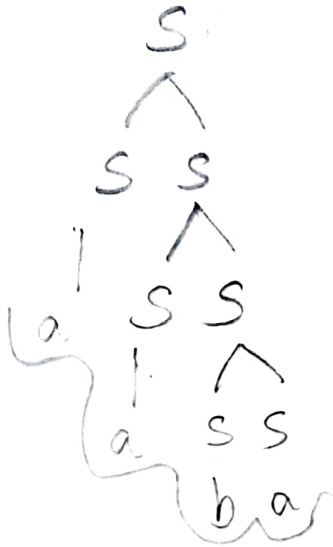




Ex2: P: S→SS/a/b.

w = aaba.





aaba.

LL parser:

→ It accepts (LL) grammer. Its denoted as LL(K).

→ The first L → represents the I/p from left to right.

→ The second L → represents the left most derivation.

→ K represents the no. of look a heads (How many time you trag the list. generally k=1

∴ LL(K) = LL(1).

→ A grammar G is LL(1), If there are two distinct production.

A → (α/β)

condition:

(1) For no teeminal α & β derive string beginning with a.

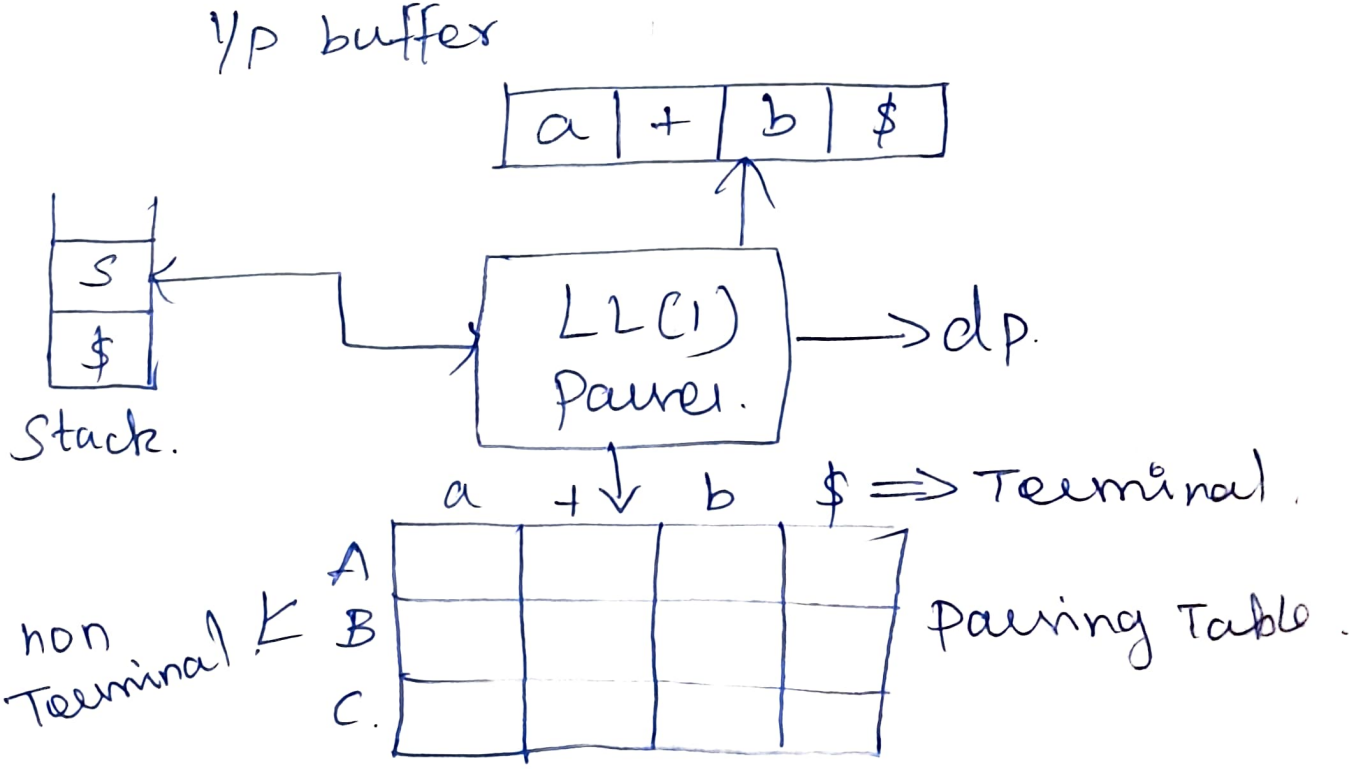
(2) At most one of α & β can derive empty string.

(3) If β → ∈, then 'a' does not derive any string begnning with a teeminal in Follow(A).

→ LL(1) use data structure are,

- L> I/p buffer
- L> stack
- L> parsing table.

Structure of LL(1):



Construction of