# Parsing LL(K) Grammar

- In input string we read 1 symbol at a time
- leftmost derivation is used ( leftmost non terminal will be deduced first )
- We read input string from left to right

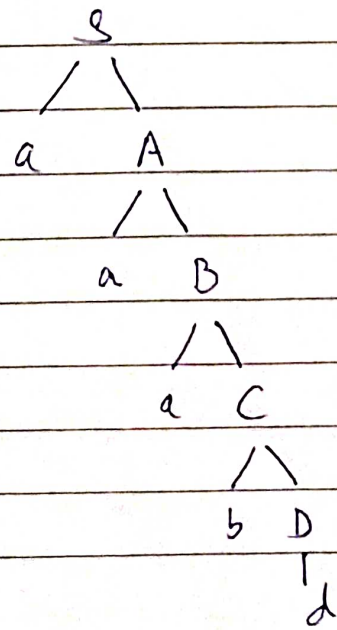
LL grammar is used for Top Down Parsing

eg. Determine if given grammar is LL(1) or not
Consider $\omega$ = a a a b d

$S \rightarrow aA \mid bB$
$A \rightarrow aB \mid cB$
$B \rightarrow bC \mid aC$
$C \rightarrow bD$
$D \rightarrow d$

Note: If it is not deterministic to select a single production then that is not LL(1) grammar

Ans)




Yes, it is LL(1) grammar

eg. $S \to abB \mid aaA$

$B \to d$

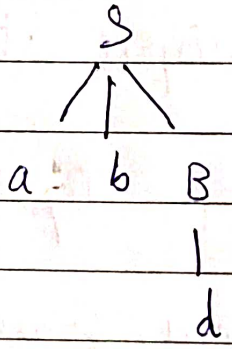
$A \to c \mid d$

here $w = abd$

Ans) It is not deterministic to read single symbol in input string.

So, It is not LL(1) grammar.

We read two symbols at a time




So, it is LL(2) grammar

Note: $LL(K) \subset LL(K+1)$

Means, if a grammar is LL(K), then it is also LL(K+1), LL(K+2), LL(K+3), LL(K+4), ... and so on.

If a grammar is not LL(K) then it can be LL(K+1)