LL(1) : short cut trick

(i) Necessary conditions for grammar to be LL(1) are:

(a) Left recursion absent

(b) unambiguous grammar

(c) Left-factored grammar.

Failing any of the above will lead to not LL(1).

(ii) Parse table for LL(1) must not contain any multiple entries.

or

A grammar will be LL(1) iff:

⇒ For a non-terminal X,

$$X \to \alpha_1 \mid \alpha_2 \mid \dots \mid \alpha_n \mid \epsilon$$

First(α_1), First(α_2) .... First(α_n), Follow(X)

are all mutually exclusive.

(doesn't contain any same element)

[this will be corresponding to "ε" production]

Example:-

(a) $S \to aSbS \mid bSaS \mid \epsilon$

⇒ First($aSbS$) = {a}

First($bSaS$) = {b}

Follow(S) = {\$, a, b}

} ⇒ Follow(S) ∩ First(bSaS) ≠ φ

⇓

Hence, grammar is not LL(1).

Parse table will surely have multiple entry.

(b) $S \rightarrow iCtSS_1 \mid a$ $\Rightarrow \{i\} \cap \{a\} = \phi$

$S_1 \rightarrow eS \mid \epsilon$ $\Rightarrow \{e\}, \{e, \$\} \neq \phi$

$C \rightarrow b$

Hence grammar is not LL(1)

(c) $S \rightarrow aSa \mid bS \mid c$

$\Rightarrow$ $First(aSa) = \{a\}$
$First(bS) = \{b\}$
$First(c) = \{c\}$

No element repeating or no two will have $\neq \phi$ intersection and hence LL(1).