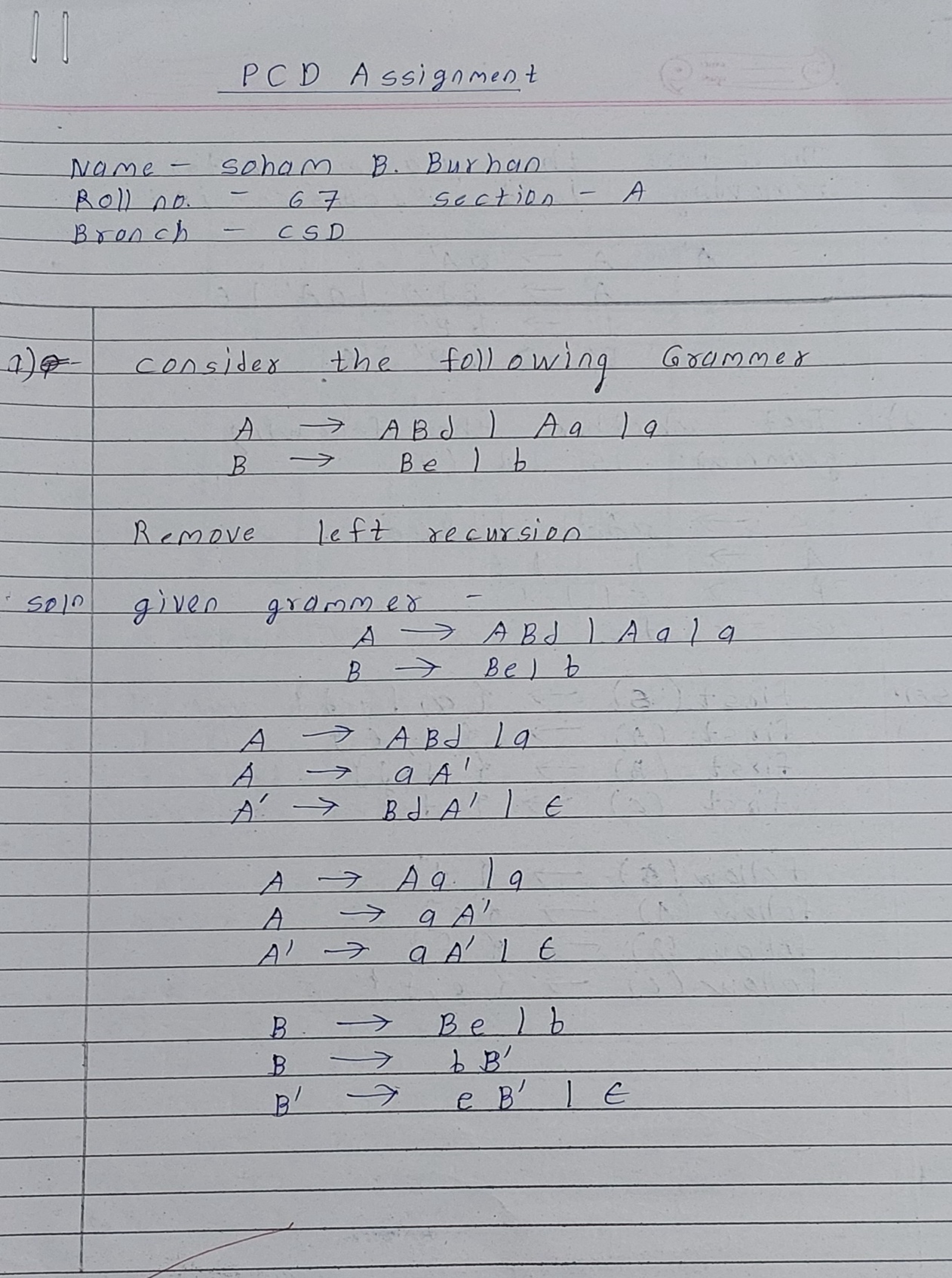
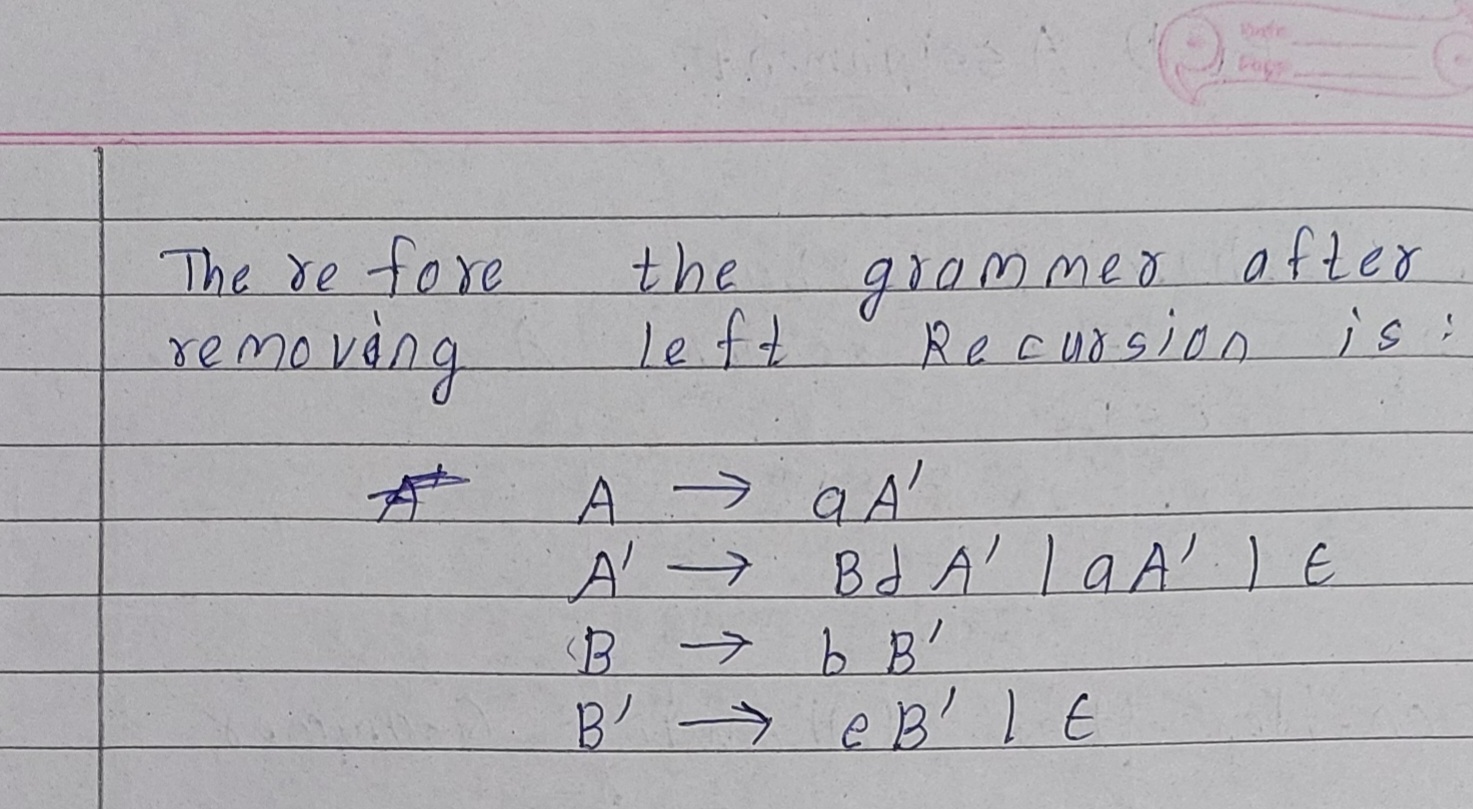
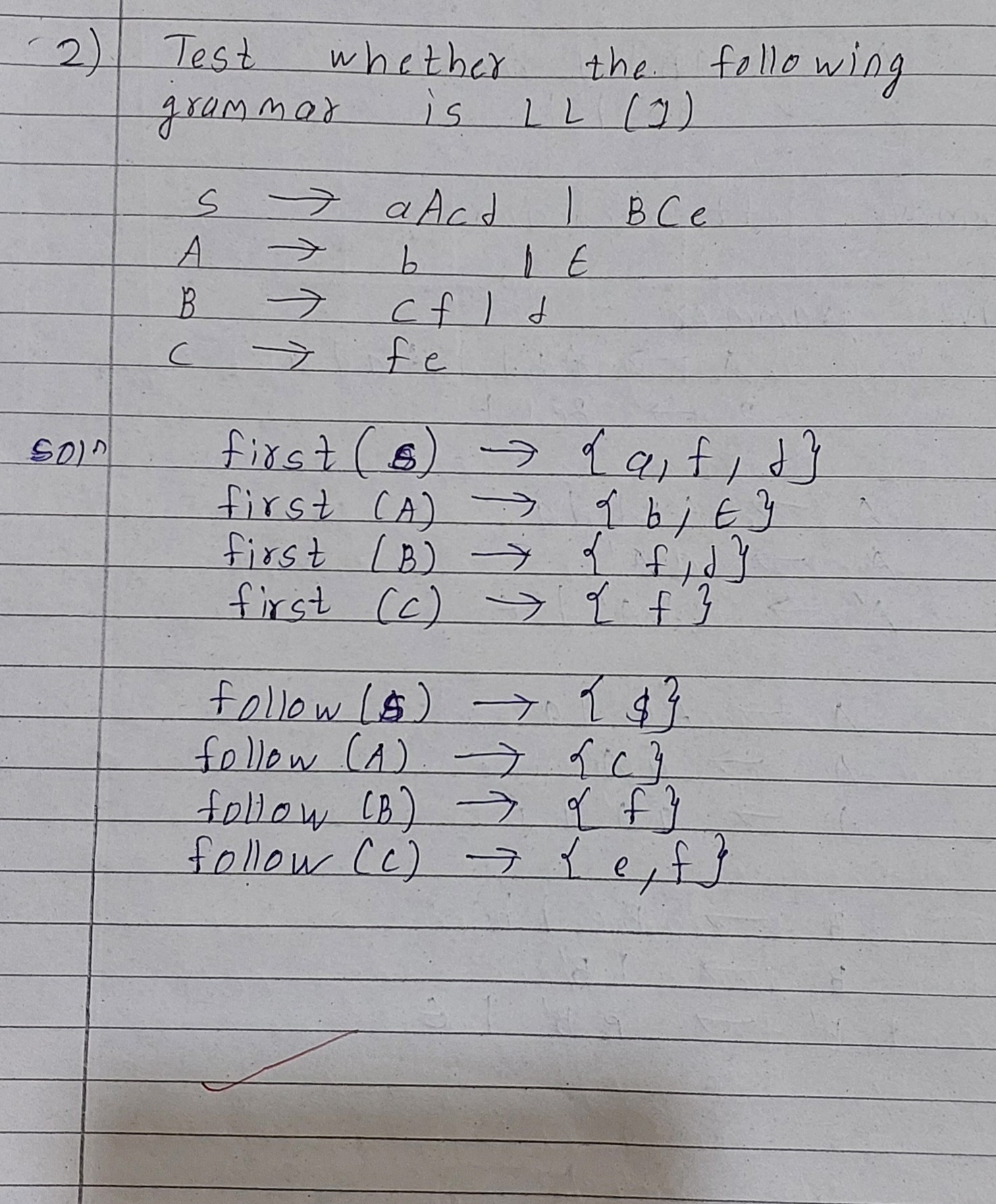
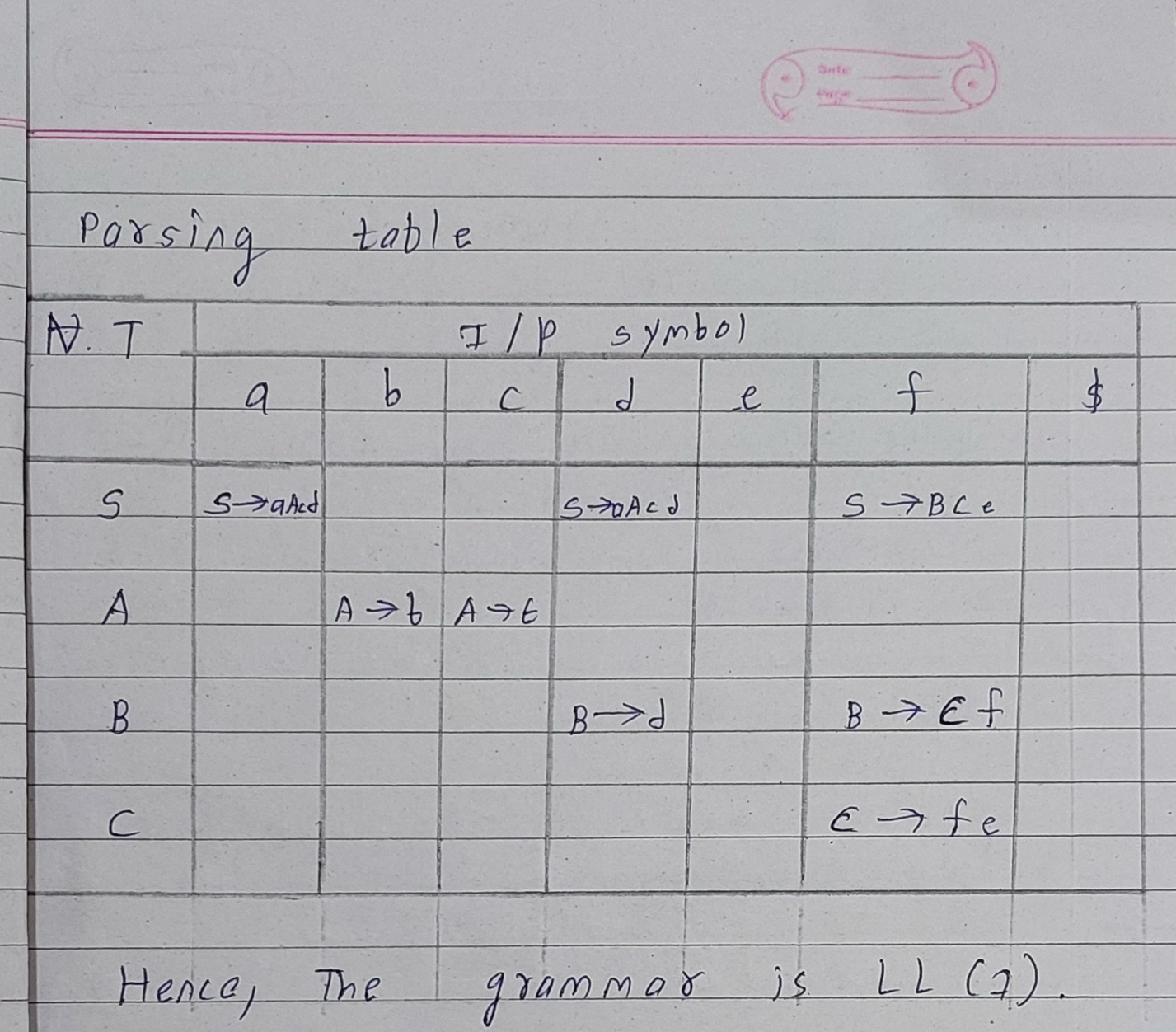
**PCD ASSIGNMENT**

****

****





**3)** **Every unambiguous grammar is LL(1). Comment on the Truth/Falsehood of the statement and justify it.**

Ans. :

The statement "Every unambiguous grammar is LL(1)" is False.

Justification:

An LL(1) grammar is a context-free grammar where the next symbol to be parsed can be uniquely determined by looking at the top symbol on the stack and the next input symbol. This restriction makes LL(1) grammars relatively easy to parse using a top-down parsing technique.

However, not all unambiguous grammars are LL(1).

Left Recursion: LL(1) grammars cannot handle left-recursive productions directly.

Ambiguity Due to Common Prefixes: Even if a grammar is not left-recursive, it can still be non-LL(1) due to common prefixes.

example:

The grammar A-> aB l aC unambiguous but not LL(1) due to the common prefix a.

In Conclusion:

While LL(1) grammars are a subset of unambiguous grammars, they have specific limitations related to left recursion and common prefixes. Therefore, not all unambiguous grammars can be parsed using an LL(1) parser.

**4) Write the comparison among SLR Parser, LALR parser and Canonical LR Parser**.

Ans :

|  |  |  |  |
| --- | --- | --- | --- |
| **Feature** | **SLR PARSER** | **LALR PARSER** | **CLR PARSER** |
| Augumented  Grammar | Yes | Yes | Yes |
| Construction | Based on follow sets | Based on lookahead sets | Based on goto and action function |
| Table Size | Smaller | Smaller than CLR but larger than SLR | Largest |
| Parsing speed | Fastest | Faster than CLR but slower than SLR | Slowest |
| Error recovery | Good | Good | Poor |
| Handling Ambiguity | Cannot handle | Can handle some ambiguities | Can handle all ambiguities |
| Example | Simple  grammars | More complex grammars than SLR | Most complex grammars |