

The provided grammar is left-recursive, because of the production D → D e b.

I replaced it with D → e D b. This new production does allow to parse the provided string. The new productions are: P = {

1. S → L d X

2. X → D

3. L → c a

4. L → a L

5. D → b

6. D → e D b

}

1. Sets FIRST and FOLLOW

|  |  |  |
| --- | --- | --- |
|  | FIRST | FOLLOW |
| S | c,a | $ |
| X | b,e | $ |
| L | c,a | d |
| D | b,e | $,b |

2. Constructing the LL(1) Parsing Table

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | a | b | c | d | e | $ |
| S | LdX |  | LdX |  |  |  |
| X |  | D |  |  | D |  |
| L | aL |  | ca |  |  |  |
| D |  | b |  |  | eDb |  |

3. Analyzing the string **aaaacadeebbb**

|  |  |
| --- | --- |
| Production | Derivation |
|  | S |
| S→LdX | LdX |
| L→aL | aLdX |
| L→aL | aaLdX |
| L→aL | aaaLdX |
| L→aL | aaaaLdX |
| L→ca | aaaacadX |
| X→D | aaaacadD |
| D→eDb | aaaacadeDb |
| D→eDb | aaaacadeeDbb |
| D→b | aaaacadeebbb |

