The FIRST condition for an LL grammar is violated by the RHS rules for <stmt_list> and <expression> in Example 3.1. An LL grammar can be obtained by left-factoring these rules and replacing the right recursion in <stmt_list> with repetition. Using EBNF, the modified rules for Example 3.1 are:

<program> → **begin** <stmt_list> **end**
<stmt_list> → <stmt> { ; <stmt> }
<stmt> → <var> = <expression>
<var> → A | B | C
<expression> → <var> [ ( + | - ) <var> ]

The corresponding syntax graphs are shown below: