Assignment 6

1. The grammer is ambiguous since it can evaluate one string in multiple ways.

For example, the string 'let x = 1 in x; x'

Parse Deviation 1

<expr> → let <id> = <expr> in <expr>
→ let x = <expr> in <expr>
→ let x = <dig> in <expr>
→ let x = 1 in <expr>
→ let x = 1 in <expr>; <expr>
→ let x = 1 in <id> ; <expr>
→ let x = 1 in x ; <expr>
→ let x = 1 in x ; <id>
→ let x = 1 in x ; x

Parse Deviation 2

<expr> → <expr>; <expr>
→ let <id> = <expr> in <expr>; <expr>
→ let x = <expr> in <expr>; <expr>
→ let x = <dig> in <expr>; <expr>
→ let x = 1 in <expr>; <expr>
→ let x = 1 in <id>; <expr>
→ let x = 1 in x ; <expr>
→ let x = 1 in x ; <id>
→ let x = 1 in x ; x

2. <id> ::= a | b | c | ... | z

<dig> ::= 0 | 1 | 2 | ... | 9

<expr> ::= <term>
| let <id> = <expr> in <expr>
| <expr>; <expr>
| begin <expr> end

<term> ::= ()
| <dig>
| <id>
| <term>, <term>

I made is unambiguous by adding the nonterminal 'term' which takes care of the simplification of a digit, identifier, or semicolon