Homework 2

1. Chapter 3 #3

We have the following BNF:

$\langle assign \rangle \rightarrow \langle id \rangle = \langle expr \rangle$
$\quad \langle id \rangle \rightarrow A \mid B \mid C$
$\quad \langle expr \rangle \rightarrow \langle expr \rangle * \langle term \rangle \mid \langle term \rangle$
$\quad \langle term \rangle \rightarrow \langle factor \rangle + \langle term \rangle \mid \langle factor \rangle$
$\quad \langle factor \rangle \rightarrow (\langle expr \rangle) \mid \langle id \rangle$

2. Chapter 3 #6(b)

Leftmost derivation:

$\langle assign \rangle \implies \langle id \rangle = \langle expr \rangle$
$\quad \implies B = \langle expr \rangle$
$\quad \implies B = \langle id \rangle * \langle expr \rangle$
$\quad \implies B = C * \langle expr \rangle$
$\quad \implies B = C * (\langle expr \rangle)$
$\quad \implies B = C * (\langle id \rangle * \langle expr \rangle)$
$\quad \implies B = C * (A * \langle expr \rangle)$
$\quad \implies B = C * (A * \langle id \rangle + \langle expr \rangle)$
$\quad \implies B = C * (A * C + \langle id \rangle)$
$\quad \implies B = C * (A * C + B)$

Parse tree:

3. Chapter 3 #7(a)

$\langle assign \rangle \implies \langle id \rangle = \langle expr \rangle$
$\quad \implies A = \langle term \rangle * \langle factor \rangle$
$\quad \implies A = \langle term \rangle * \langle id \rangle$
$\quad \implies A = \langle term \rangle * C$
$\quad \implies A = \langle factor \rangle * C$
$\quad \implies A = (\langle expr \rangle) * C$
$\quad \implies A = (\langle expr \rangle + \langle term \rangle) * C$
$\quad \implies A = (\langle expr \rangle + \langle factor \rangle) * C$
$\quad \implies A = (\langle expr \rangle + \langle id \rangle) * C$
$\quad \implies A = (\langle expr \rangle + B) * C$
$\quad \implies A = (\langle term \rangle + B) * C$
$\quad \implies A = (\langle factor \rangle + B) * C$
$\quad \implies A = (A + B) * C$

Parse tree:

4. Chapter 3 #11

The strings `baab` and `bbaab` are valid, but `bbbab` and `bbaaaaa` are not.

5. Chapter 3 #15

We have the following EBNF grammar:

$\langle program \rangle \rightarrow$ `begin` $\langle stmt\_list \rangle$ `end`

    $\langle stmt\_list \rangle \rightarrow \langle stmt \rangle \; \{; \; \langle stmt\_list \rangle\}$

    $\langle stmt \rangle \rightarrow \langle var = \textit{¡expression} \rangle$

    $\langle var \rangle \rightarrow$ A | B | C

    $\langle expression \rangle \rightarrow \langle var \rangle \; \{(+ \; | \; \text{-}) \; \langle var \rangle \; \}$