BU CS320 Assignment 5: Context Free Grammars

October 30, 2023

1. Given the following grammar where $\langle expr \rangle$ is the starting symbol

$$
\begin{aligned}
\langle digit \rangle &::= 0 \mid 1 \mid 2 \mid 3 \mid 4 \mid 5 \mid 6 \mid 7 \mid 8 \mid 9 \\
\langle nat \rangle &::= \langle digit \rangle \mid \langle digit \rangle \langle nat \rangle \\
\langle int \rangle &::= \langle nat \rangle \mid \text{-}\langle nat \rangle \\
\langle expr \rangle &::= \langle int \rangle \\
&\quad \mid \ (\ \langle expr \rangle\ ) \\
&\quad \mid \ \langle expr \rangle + \langle expr \rangle \\
&\quad \mid \ \langle expr \rangle * \langle expr \rangle
\end{aligned}
$$

Derive the sentence using *rightmost derivation*.

12 + 2 * -07

<expr> ::= <expr> * <expr>

=> <expr> * <int>

=> <expr> * - <nat>

=> <expr> * - <digit> <nat>

=> <expr> * - <digit> <digit>

=> <expr> * - <digit> 7

=> <expr> * - 07

=> <expr> + <expr> * -07

=> <expr> + <int> * -07

=> <expr> + <nat> * -07

=> <expr> + <digit> * -07

=> <expr> + 2 * -07

=> <int> + 2 * -07

=> <nat> + 2 * -07

=> <digit> <nat> + 2 * -07

=> <digit> <digit> + 2 * -07

=> <digit> 2 + 2 * -07

=> 12 + 2 * -07

2. Given the following grammar where ⟨*stmt*⟩ is the starting symbol.

⟨*digit*⟩ ::= 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9

⟨*letter*⟩ ::= a | b | c | . . . | z

⟨*nat*⟩ ::= ⟨*digit*⟩ | ⟨*digit*⟩⟨*nat*⟩

⟨*int*⟩ ::= ⟨*nat*⟩ | -⟨*nat*⟩

⟨*expr*⟩ ::= ⟨*int*⟩
| (⟨*expr*⟩)
| ⟨*expr*⟩ + ⟨*expr*⟩
| ⟨*expr*⟩ * ⟨*expr*⟩

⟨*id*⟩ ::= ⟨*letter*⟩ | ⟨*letter*⟩⟨*id*⟩

⟨*stmt*⟩ ::= ⟨*id*⟩ = ⟨*expr*⟩
| for ⟨*id*⟩ = ⟨*expr*⟩ to ⟨*expr*⟩ do ⟨*stmt*⟩
| { ⟨*stmts*⟩ }
| pass

⟨*stmts*⟩ ::= ⟨*stmt*⟩ | ⟨*stmt*⟩ ; ⟨*stmts*⟩

Derive the sentence using *leftmost derivation*.

```
for x = -12 to 10 do { y = 0; pass }
```

<stmt> ::= for <id> = <expr> to <expr> do <stmt>

=> for <letter> = <expr> to <expr> do <stmt>

=> for x = <expr> to <expr> do <stmt>

=> for x = <int> to <expr> do <stmt>

=> for x = -<nat> to <expr> do <stmt>

=> for x = -<digit><nat> to <expr> do <stmt>

=> for x = -1<nat> to <expr> do <stmt>

=> for x = -1<digit> to <expr> do <stmt>

=> for x = -12 to <expr> do <stmt>

=> for x := -12 to <int> do <stmt>

=> for x := -12 to <nat> do <stmt>

=> for x := -12 to <digits> <nat> do <stmt>

=> for x := -12 to 1 <nat> do <stmt>

=> for x := -12 to 1 <digits> do <stmt>

=> for x := -12 to 10 do <stmt>

=> for x := -12 to 10 do { <stmts> }

=> for x := -12 to 10 do { <stmt> ; <stmts> }

=> for x := -12 to 10 do { <id> := <expr> ; <stmts> }

=> for x := -12 to 10 do { y := <expr> ; <stmts> }

=> for x := -12 to 10 do { y := <int> ; <stmts> }

=> for x := -12 to 10 do { y := <nat> ; <stmts> }

=> for x := -12 to 10 do { y := <digits> ; <stmts> }

=> for x := -12 to 10 do { y := 0 ; <stmts> }

$\Rightarrow$ for x := -12 to 10 do { y := 0; <stmt> }

$\Rightarrow$ for x := -12 to 10 do { y := 0; pass }