Computer Language Processing

Exercise Sheet 03

October 11, 2022

Welcome to the third exercise session of CS320! We have selected exercises from the 2019 edition of this class (ex. 3, 4), as well as exercises 3.11, 3.9 from Basics of Compiler Design.

As usual, you can find the solutions of exercises 3.11 and 3.9 on the textbook's website, while solutions of exercises 3 and 4 will be provided to you.

Exercises from the book

Here are a couple of selected exercises from Basics of Compiler Design to work on grammar.

- **3.11 | Nullable, First**
- **3.9 | Nullable, First, Follow**

Exercise 3

Consider the following grammar for balanced parentheses:

S ::= B **EOF**

B ::= ε | B **(** B **)**

1. Compute NULLABLE, FIRST and FOLLOW for each non-terminal
2. Build the LL(1) parsing table for the grammar. Is the grammar LL(1)?
3. Build the LL(1) parsing table for the following grammar. Is the grammar LL(1)?

   S ::= B **EOF**

   B ::= ε | **(** B **)** B

4. Run the LL(1) parsing algorithm on the following input strings, . Show the derivation you obtain, or the derivation up until the error.

   (( ) ( )) EOF

   ( )) (EOF

5. Show that the second grammar describes the language of balanced parentheses. To do so, show that:

   (a) Every parse tree of grammar yields a balanced parenthesis string.

   (b) For every balanced parenthesis string there exists a parse tree.

Exercise 4

1. Build an LL(1) grammar for expressions consisting of variables, infix binary addition and multiplication (with usual priorities), and parentheses. Note that we do not care about associativity of binary operations at the level of parse tree.

   Build the LL1 parsing table.

2. Parse the following input strings using your parsing table. Show the derivation and parse trees.

   x + y * (z + x) EOF

   x * y * z EOF

   ((x)) EOF