UNIVERSITY OF AMSTERDAM
Informatics Institute
Systems and Networking Lab
Parallel Computing Systems Group

Compiler Construction
2019/20
Dr. Clemens Grelck
February 18, 2020

Assignment 3

Syntactic Analysis

Consider the following context-free grammar of expressions constructed from identifiers using binary addition, unary negation, post-decrement and post-increment:

$$
\begin{array}{lcl}
\mathit{Expr} & \Rightarrow & \mathbf{Id} \\
 & | & \mathit{Expr}\ +\ \mathit{Expr} \\
 & | & -\ \mathit{Expr} \\
 & | & \mathit{Expr}\ -- \\
 & | & \mathit{Expr}\ ++
\end{array}
$$

Assignment 3.1: Precedence and Associativity

Rewrite the above grammar such that it properly expresses precedence and associativity according to the C standard: http://en.wikipedia.org/wiki/Operators_in_C_and_C++

Assignment 3.2: Left- and Right-recursive Grammars

Convert the (left-recursive) grammar developed for Assignment 3.1 into a right-recursive grammar.

Assignment 3.3: Predictive Grammars

Convert the right-recursive grammar of Assignment 3.2 into a start-separated, predictive grammar.

Assignment 3.4: Recursive-descent Parsing

Derive pseudo code for a top-down recursive-descent parser from the start-separated, predictive grammar of Assignment 3.3.

Assignment due date: Friday, February 28, 2020