Generating LL Translation Grammars for Arithmetic Expressions

Bc. Radek Vít   Bc. Tomáš Ženčák

October 31, 2017

Abstract

Translation grammars are a formal device for defining translation. As an alternative to context-free grammars in compilers, they generate intermediate code along with parsing input, separating syntactic structure from semantic meaning where possible. Unlike LL(1) grammars, LL(1) translation grammars are well-suited for defining arithmetic expressions.

In this lecture, we define the notion of LL(1) translation grammars. We discuss the limitations of arithmetic expressions defined by LL(1) grammars and how we can avoid some of them by using LL(1) translation grammars instead. Then, we present an algorithm for generating an LL(1) translation grammar that translates expressions with unary and binary operators, and generates a postfix notation version of these expressions, which reflect the precedence and associativity of the individual operators.