Parsing techniques – LL(k) Parsing

František Hanák, xhanak01
Martin Kletzander, xkletz00

November 9, 2011

Chapters 8.2 through 8.5 from the book Parsing techniques: A Practical Guide describe several parsing techniques used in compilers based on LL(k) grammars. The book introduces the main questions of LL table constructions, table conflicts solving, handling ε-rules and describes different kinds of grammars, e.g., linear approximation of LL(k), LL-regular, ELL(1) (Extended LL(1)), etc.

Chapter 8.2 describes the main aspect of deterministic top down LL(1) parsing. It deals with construction of LL tables for grammars without ε-rules using $FIRST$ sets. In addition the introduction to $FOLLOW$ set is made and thanks to these sets the principle of LL table construction for grammars with ε-rules is described. It also discusses the conflicts in LL(1) parsing and techniques for solving the common problems such as left-factoring and left-recursion elimination. At the end of this chapter an implementation approach of recursive descent is described.

Chapter 8.3 shows many ways of increasing the power of deterministic LL parsing. The first main approach is to use LL(k) grammars for $k > 1$ that are being compared to LL(1) grammars and the main benefits are shown. For the use of longer look-ahead, the construction of $FIRST$ and $FOLLOW$ sets is redefined. The second way of improving deterministic LL(1) parsing is to use linear-approximate LL(2) that is described with the usage of $FIRST$ and $SECOND$ sets. The strength and memory consumption is compared to LL(2) parser. The last approach shown is LL-Regular parsing that is used for unbounded look-ahead technique.

Chapter 8.4 describes how to get a parse tree during LL(1) parsing. It discusses the creation of a new grammar rule for each prediction using global counter for nonterminals. Also, the prediction stack is mentioned as an addition to the usual stack.

Chapter 8.5 handles extended LL(1) grammars and a principle of transformation of these grammars into ordinary LL(1) grammars. In this chapter, the author also describes main benefits of extended LL(1) grammars as a foundation for recursive descent parsers.