

Parsing Strong LL(k) Languages with a Modified LL(1) Parsing Table and a Lookahead Table

Traditionally, we shall not construct a parsing table for a strong LL(k) grammar for $k > 1$, because a canonical parsing table for such a grammar is too space-consuming to construct a real one. But we observe that parsing a string in the strong LL(k) language is not always necessary to construct a huge parsing table. In many cases, a string in the strong LL(k) language can be parsed successfully with a modified parsing table whose size is the same as that of an LL(1) and a small lookahead table. In this paper, we present a parsing technique to parse the strong LL(k) languages with a modified LL(1) parsing table and a lookahead table. Two algorithms are given to show the canonical parsing method for a strong LL(k) grammar. We present another two algorithms to illustrate how to construct a modified LL(1) parsing table and a lookahead table and how to parse strong LL(k) languages with these two tables. Our method can reduce the parsing table size considerably, of course, will reduce the ability of handling error recovery, too. An error recovery scheme is proposed to remedy this weakness. Several practical methods that can be used to parse strong LL(k) languages are also compared. Finally, we discuss how to apply this method to parse LR(k) languages with a similar modified LR(1) parsing table and a lookahead table.