Compiler Design
Syntax Analysis
Overview

conf. dr. ing. Ciprian-Bogdan Chirila

chirila@cs.upt.ro

http://www.cs.upt.ro/~chirila

Introduction

- The Role of the Parser
- Representative Grammars
- Syntax Error Handling
- Error Recovery Strategies

Context-Free Grammars

- The Formal Definition of a Context Free Grammar
- Notational Conventions
- Derivations
- Parse Trees and Derivations
- Ambiguity
- Verifying the Language Generated by a Grammar
- Context Free Grammars versus Regular Expressions

Writing a Grammar

- Lexical versus Syntactic Analysis
- Eliminating Ambiguity
- Elimination of Left Recursion
- Left Factoring
- Non-Context-Free Language Constructs

Top-Down Parsing

- Recursive Descendant Parsing
- FIRST and FOLLOW
- LL(1) Grammars
- Nonrecursive Predicting Parsing
- Error Recovery in Predictive Parsing

Bottom-Up Parsing

- Reductions
- Handle Pruning
- Shift-Reduce Parsing
- Conflicts During Shift-Reduce Parsing

Introduction to LR Parsing

- Why LR Parsers ?
- Items and LR(0) Automaton
- The LR-Parsing Algorithm
- Constructing the SLR-Parsing Tables
- Viable Prefixes

More Powerful LR Parsers

- Canonical LR(1) Items
- Constructing LR(1) Set of Items
- Canonical LR(1) Parsing Tables
- Constructing LALR Parsing Tables
- Efficient Construction of LALR Parsing Tables
- Compaction of LR Parsing Tables

Using Ambiguous Grammars

- Precedence and Associativity to Resolve Conflicts
- The "Dangling-Else" ambiguity
- Error Recovery in LR Parsing

Parser Generator

- The Parser Generator Yacc
- Using Yacc with Ambiguous Grammars
- Creating Yacc Lexical Analyzers with Lex
- Error Recovery in Yacc

Bibliography

- Alfred V. Aho, Monica S. Lam, Ravi Sethi, Jeffrey D. Ullman – Compilers, Principles, Techniques and Tools, Second Edition, 2007