Compiler Design
Lexical Analysis
Overview

conf. dr. ing. Ciprian-Bogdan Chirila

chirila@cs.upt.ro

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

The Role of Lexical Analyzer

- Lexical Analysis vs. Parsing
- Tokens, Patterns and Lexemes
- Attributes for Tokens
- Lexical Errors

Input Buffering

- Buffer Pairs
- Sentinels

The Lexical-Analyzer Generator Lex

- Use of Lex
- Structure of Lex Programs
- Conflict Resolution in Lex
- The Lookahead Operator

Finite Automata

- Nondeterministic Finite Automata
- Transition Tables
- Acceptance of Input Strings by Automata
- Deterministic Finite Automata

From Regular Expressions to Automata

- Conversion of an NFA to DFA
- Simulation of a NFA
- Efficiency of NFA Simulation
- Construction of an NFA from a Regular Expression

Optimization of DFA-Based Pattern Matchers

- Important States of an NFA
- Functions Computed from the Syntax Tree
- Computing nullable, firstpos and lastpos
- Computing followpos
- Converting a Regular Expression Directly to a DFA
- Minimizing the Number of States of a DFA
- State Minimization of a Lexical Analyzer
- Trading Time for Space in DFA Simulation

Bibliography

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