PRACTICAL COMPILER CONSTRUCTION

PRACTICAL COMPILER CONSTRUCTION

Building a Language, One Pass at a Time

by Marcus Webb

PRACTICAL COMPILER CONSTRUCTION

This is a sample starter document generated from the `nostarch` LaTeX class. It is not an official No Starch Press publication.

For everyone who has ever stared at a shift-reduce conflict.

CONTENTS IN DETAIL

FOREWORD **ix**

1
LEXICAL ANALYSIS **1**

Regular Expressions to Automata 1

2
PARSING **3**

Handling Ambiguity 3

BIBLIOGRAPHY **5**

INDEX **7**

FOREWORD

This book grew out of a compilers course taught over several years. It assumes you can already write ordinary programs, but have never written a program that reads programs.

1

LEXICAL ANALYSIS

The first stage of any compiler turns a stream of characters into a stream of *tokens*. A lexer recognizes patterns — identifiers, numbers, keywords — using regular expressions compiled down to a finite automaton.

Regular Expressions to Automata

Every regular expression can be converted to a nondeterministic finite automaton (NFA) via Thompson's construction, and every NFA to an equivalent deterministic finite automaton (DFA) via the subset construction.

Note. Most real lexer generators skip the NFA step entirely and construct the DFA directly from the regular expression's syntax tree.

2

PARSING

A parser consumes the token stream and builds a parse tree according to a grammar. This chapter covers recursive-descent parsing for LL(1) grammars and shift-reduce parsing for LR(1) grammars.

Tip. Recursive-descent parsers are easier to write by hand; LR parsers are easier to generate from a grammar specification and handle a strictly larger class of grammars.

Handling Ambiguity

The classic dangling-else problem is a textbook example of grammar ambiguity: an `if-then-else` grammar resolves it by convention, binding each `else` to the nearest unmatched `if`, per Aho et al. [2006]. A rigorous treatment of parser generator theory appears in Grune and Jacobs [2008].

BIBLIOGRAPHY

Alfred V. Aho, Monica S. Lam, Ravi Sethi, and Jeffrey D. Ullman. *Compilers: Principles, Techniques, and Tools*. Addison-Wesley, 2 edition, 2006.

Dick Grune and Ceriel J. H. Jacobs. *Parsing Techniques: A Practical Guide*. Springer, 2 edition, 2008.

INDEX

D

dangling else, 3

F

finite automaton
  deterministic, 1
  nondeterministic, 1

G

grammar
  LL(1), 3
  LR(1), 3

L

lexer, 1

P

parser, 3

T

tokens, 1