Lexical Analysis - Scanner

66.648 Compiler Design Lecture 2 (01/14/98)

Computer Science

Rensselaer Polytechnic

Lecture Outline

- Scanners/ Lexical Analyzer
- Regular Expression NFA/DFA
- Administration

Introduction

- Lexical Analyzer reads source text and produces tokens, which are the basic lexical units of the language.

Example: System.out.println("Hello Class");

has tokens System, dot, out, dot, println, left paren, String Hello Class, right paren and a semicolon.

Lexical Analyzer/Scanner

- Lexical Analyzer also keeps track of the source-coordinates of each token - which file name, line number and position. This is useful for debugging purposes.
- Lexical Analyzer is the only part of a compiler that looks at each character of the source text.

Tokens - Regular Expressions

Qn: How are tokens defined and recognized?

Ans: By using regular expressions to define a token as a formal regular language.

Formal Languages --
Alphabet - a finite set of symbols, ASCII is a computer alphabet.
String - finite sequence of symbols from the alphabet.

Formal Lang. Contd

Empty string = special string of length 0

Language = set of strings over a given alphabet (e.g., set of all programs)

Regular Expressions:
A reg. expression E denotes a language L(E)

Regular Expressions

An alphabet symbol,a, is a regular expression.
An empty symbol is also a regular expression.

If E1 and E2 are regular expressions denoting languages L(E1) and L(E2), then

- E1 | E2 is a regular expression denoting a language L(E1) union L(E2).
- E1 E2 is a regular expression denoting a language L(E1) followed by L(E2).
- E* (E star) is a regular expression denoting L(E star) = Kleene closure of L(E).

Examples

- Specify a set of unsigned numbers as a regular expression.

Examples: 1997, 19.97

Solution: Note use of regular definitions as intermediate names that define regular subexpressions.

digit → 0 | 1 | 2| 3| … | 9
digit → digit digit* (often written as digit+) This is the Kleene star. Means 1 or more digits.

Example Contd

optional_fraction ⟶ . digits | epsilon

num ⟶ digits optional_fraction

Note that we have used all the definitions of a regular expression.
One can define similar regular expression(s) for identifier comments, Strings, operators and delimiters.
Qn: How to write a regular expression for identifiers?
(identifiers are letters followed by a letter or a digit).

Identifiers contd

letter ⟶ a|A|b|B| … |z|Z

digit ⟶ 0|1|2| … | 9

letter_or_digit ⟶ letter | digit

identifier ⟶ letter | letter letter_or_digit*

Building a recognizer

A General Approach

- Build Nondeterministic Finite Automaton (NFA) from Regular Expression E.
- Simulate execution of NFA to determine whether an input string belongs to L(E).
- The simulation can be much simplified if
- you convert your NFA to Deterministic Finite Automaton (DFA).

NFA

A transition graph represents a NFA.

- Nodes represent states. There is a distinguished start state and one or more final states.
- Edges represent state transitions.
- An edge can be labeled by an alphabet or an empty symbol

NFA contd

From a state(node), there may be more than one edge labeled with the same alphabet and there may be no edge from a node labeled with an input symbol.

- NFA accepts an input string iff (if and only if) there is a path in the transition graph from the start node to some final state such that the labels along the edge spell out the input string.

Deterministic Finite Automaton (DFA)

A finite automaton is deterministic if

- It has no edges/transitions labeled with epsilon.
- For each state and for each symbol in the alphabet, there is exactly one edge labeled with that symbol.

Such a transition graph is called a state graph.

DFA's Counted

- NFAs are quicker to build but slower to simulate.
- DFAs are slower to build but quicker to simulate.
- The number of states in a DFA may be exponential in the number of states in a DFA.

Administration

- We are in Chapter 3 of Aho, Sethi and Ullman's book. Please read that chapter and chapter 1 which we covered in Lecture 1.
- Work out the first few exercises of chpater 3.
- Lex and Yacc Manuals will be handed out on Monday along with first project.

Where to get more information

- Newsgroup comp.compilers
- There are a lot of resources on Java in the internet. Please browse through www.java.sun.com and www.gamelan.com. Please familiarize with this language as quickly as possible.
- As a warmup, write a few (at least two) java programs and try to compile and run.

Feedback

- Please let me know whether by Monday whether you are able to look at these things and work out some problems.