Programming Languages: Lecture 6
Lexical Analysis

Rishabh Dhiman

15 January 2022

1 Lexical Analysis

A source program consists of a stream of characters.

Given a stream of characters that make up a source program the compiler must first break up this stream into a sequence of “lexemes” and other symbols.

Lexemes are often separated by non-lexemes.

Certain sequences of characters are *not* tokens (Eg.: comments)

1.1 Erroneous Lexemes

Some lexemes violare all rules of tokens.

- 12ab would not be an identifier or a number in most languages, 0x2ab may be hex.
- 127.0.1 probably won’t be a number but 127.0.0.1 is a valid IP.

1.2 Tokens

Common examples

- Constants
- Identitfiers – Name of variables, constants, procedures, functions, etc.
- Keywords/Reserved words – void, public, main
- Operators – +, *, /
- Punctuation – „ :, .
- Brackets – (,), [,], begin, end, case, esac

1.3 Scanning

During the scanning phase the compiler/interpreter

- takes a stream of characters and identifies tokens from the lexemes
- eliminiates comments and redundant whitespace
- keeps track of line numbers and columb numbers and passes them as parameters to the other phases to enable error-reporting and handling to the user.

2 Regular Expressions Language

- Any set of strings built up from the symbols of A is called a language. A^* is the set of all finite strings buillt up form A.
- Each regex is a finite sequence of symbols made up of symbols from the alphabet and other symbols called operators.
- A regular expression may be used to describe an *infinite* collection of strings.

3 Language

Any collection of finite strings is a language.

4 Simple Language of Regular Expressions

We consider a simple language of regular expressions. Assume a (finite) alphabet A of symbols. Each regular expression r denotes a set of strings $\mathcal{L}(r)$. $\mathcal{L}(r)$ is also called the *language* spexified by the regular expression r.

- Symbol, for $a \in A$, $\{a\}$ refers to the single element a.
- Concatenation. $\mathcal{L}(rs) = \mathcal{L}(r)\mathcal{L}(s)$.
- Epsilon ε denotes the language with a single element the *empty* string, “ ”.
$$L(\varepsilon) = \{\varepsilon\}.$$
- Alternation. Given two regex r, s; $r \mid s$ is the set of union of the languages specified by r and s.
$$\mathcal{L}(r \mid s) = \mathcal{L}(r) \cup \mathcal{L}(s).$$
- Kleene Closure $r^* = r^0 \mid r^1 \mid \cdots$ denotes an infinite union of languages.
$$\mathcal{L}(r^*) = \bigcup_{n=0}^{\infty} \mathcal{L}(r^n).$$