Lexical Analysis

Source code (character stream)

if (b == 0) a = b;

Token stream

if (b == 0) a = b;

- Identifiers: x y11 elsen _i00
- Integers: 2 1000 -500 5L
- Floating point: 2.0 .02 1. 1e5 0.e-10
- Strings: "x" "He said, \"moo\""
- Comments: /** don't change this **/
- Keywords: if else while break
- Symbols: + * { } ++ < << [ ] >=
Regular Expressions

• A language is a set of words: \{ moo, cow \}, \{ a,b,c,d,... \}

• Regular expressions describe languages

\[ abab \quad a|b \quad (a|b)^* \quad [1-9][0-9]^* \quad [a-z][a-z0-9]^* \]

• Definition

<table>
<thead>
<tr>
<th>Symbol</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>a</td>
<td>ordinary character stands for itself</td>
</tr>
<tr>
<td>(\varepsilon)</td>
<td>the empty string</td>
</tr>
<tr>
<td>R</td>
<td>S</td>
</tr>
<tr>
<td>RS</td>
<td>R followed by S (concatenation)</td>
</tr>
<tr>
<td>R*</td>
<td>R repeated 0 or more times</td>
</tr>
</tbody>
</table>

• \(L(R)\) = the language defined by regular expression \(R\)
  
  – \(L(a (moo | cow))\) = \{ amoo, acow \}
  
  – \(L([1-9][0-9]^*)\) = \{ 1,2,3,4,5,6,7,8,9,10,11,12,13,....\}
Acceptors: (a.k.a. recognizers)

• Abstract machines that determine if an input string belongs to a language, answering Yes/No.

• Finite Automata: acceptors for languages described by regular expressions
Finite Automata

• Regular Expression: \((-|\epsilon)[0-9][0-9]^*\)

• Non-deterministic Finite Automata:

• Deterministic Finite Automata:
Building an acceptor for a regular expression:

Input String $w$ is checked against the language $L(R)$ of the regular expression $R$.

- If $w \in L(R)$, the string is accepted.
- If $w \notin L(R)$, the string is rejected.