ANNOUNCEMENTS and REMINDERS

• Recitations have started.

• This Friday class will be taught by Prof. Zhang

• TA office hours have been posted. You can go to any 314 office hour.

• My office hours for next Monday are moved to next Tuesday, 12:30 - 2:00pm.

• First homework has been posted on our class website.

  **Deadline:** Friday, September 22

  Use **piazza** tool on sakai to post questions / discuss homework (tag/folder: hw1)
Review - Front end of a compiler

Front End: syntax & (static) semantics analyzer, *il* code generator (syntax-directed translation)

Front End Responsibilities:

- recognize legal programs
- report errors
- produce *il* (intermediate language / representation)
- preliminary storage map
- shape the code for the back end

Much of front end construction can be automated
Review: Syntax and Semantics of Prog. Languages

The syntax of programming languages is often defined in two layers: **tokens** and **sentences**.

- **tokens** – basic units of the language
  
  **Question:** How to spell a token (word)?
  
  **Answer:** regular expressions

- **sentences** – legal combination of tokens in the language
  
  **Question:** How to build correct sentences with tokens?
  
  **Answer:** (context-free) grammars (CFG) E.g., Backus-Naur form (BNF) is a formalism used to express the syntax of programming languages.
Lexical Analysis (Scott 2.1, 2.2)

character sequence

\[ i \_ f \_ a \_ \leq \_ b \_ t \_ h \_ e \_ n \_ c \_ = \_ 1 \_ \]

\[ \downarrow \]

scanner

\[ \text{id} \_ <a> \_ \leq \_ \text{id} \_ <b> \]

\[ \text{then} \_ \text{id} \_ <c> \_ := \_ \text{num} \_ <1> \]

token sequence

Tokens (Terminal Symbols of CFG, Words of Lang.)

- Smallest “atomic” units of syntax
- Used to build all the other constructs
- Example, Pascal:
  
  **keywords:**  program begin if then ...  
  = * / - < > = <= >= <>  
  ( ) [ ] ; := . , ...

  **number** (Example: 3.14 28 ... )

  **identifier** (Example: b square addEntry ... )
Lexical Analysis (cont.)

Identifiers

• Names of variables, etc.
• Sequence of terminals of restricted form;
  Example, Pascal: \texttt{A31}, but not \texttt{1A3}
• Upper/lower case sensitive?

Keywords

• Special identifiers which represent tokens in the language
• May be reserved (\textit{reserved words}) or not
  – E.g., Pascal: “if” is reserved.
  – E.g., FORTRAN: “if” is not reserved.

Delimiters – When does character string for token end?

• Example: identifiers are longest possible character sequence that does not include a delimeter
• Few delimiters in Fortran (not even ‘\texttt{'}’)
  – \texttt{DO I = 1.5} same as \texttt{DOI=1.5}
• Most languages have more delimiters such as ‘\texttt{'}’, new line, keywords, ...
## Regular Expressions

A syntax (notation) to specify regular languages.

<table>
<thead>
<tr>
<th>RE $r$</th>
<th>Language $L(r)$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$a$</td>
<td>{a}</td>
</tr>
<tr>
<td>$\epsilon$</td>
<td>{\epsilon}</td>
</tr>
<tr>
<td>$r \mid s$</td>
<td>$L(r) \cup L(s)$</td>
</tr>
<tr>
<td>$rs$</td>
<td>{rs \mid r \in L(r), s \in L(s)}</td>
</tr>
<tr>
<td>$r^+$</td>
<td>$L(r) \cup L(rr) \cup L(rrr) \cup \ldots$ (any number of r’s concatenated)</td>
</tr>
<tr>
<td>$r^*$</td>
<td>{\epsilon} \cup L(r) \cup L(rr) \cup L(rrr) \cup \ldots</td>
</tr>
<tr>
<td>($r^* = r^+</td>
<td>\epsilon$)</td>
</tr>
<tr>
<td>( s )</td>
<td>$L(s)$</td>
</tr>
</tbody>
</table>

(all left-assoc. in order of increasing precedence.)

$\Rightarrow$ **Note:** Inductive definition!
### Examples of Expressions

<table>
<thead>
<tr>
<th>RE</th>
<th>Language</th>
</tr>
</thead>
<tbody>
<tr>
<td>a</td>
<td>bc</td>
</tr>
<tr>
<td>(a</td>
<td>b)c</td>
</tr>
<tr>
<td>a$\epsilon$</td>
<td></td>
</tr>
<tr>
<td>a*$</td>
<td>b</td>
</tr>
<tr>
<td>ab*</td>
<td></td>
</tr>
<tr>
<td>ab*$</td>
<td>c^+</td>
</tr>
<tr>
<td>(a</td>
<td>b)*</td>
</tr>
<tr>
<td>(0</td>
<td>1)*1</td>
</tr>
</tbody>
</table>
Examples of Expressions - Solution

RE Language

\( a|bc \) \( \{a, bc\} \)

\((a|b)c\) \( \{ac, bc\} \)

\( a\varepsilon \) \( \{a\} \)

\( a^*|b \) \( \{\varepsilon, a, aa, aaa, aaaa, \ldots\} \cup \{b\} \)

\( ab^* \) \( \{a, ab, abb, abbb, abbb, \ldots\} \)

\( ab^*|c^+ \) \( \{a, ab, abb, abbb, abbbb, \ldots\} \cup \{c, cc, ccc, \ldots\} \)

\((a|b)^* \) \( \{\varepsilon, a, b, aa, ab, ba, bb, aab, \ldots\} \)

\((0|1)^*1 \) binary numbers ending in 1
More on regular expressions

For convenience, names may be introduced to represent particular regular expressions. Examples:

\[ \text{digit} \rightarrow (0 \mid 1 \mid 2 \mid 3 \mid 4 \mid 5 \mid 6 \mid 7 \mid 8 \mid 9) \]

\[ \text{letter} \rightarrow (a \mid b \mid c \mid d \mid e \mid f \mid g \mid h \mid i \mid j \mid k \mid l \mid m \mid n \mid o \mid p \mid q \mid r \mid s \mid t \mid u \mid v \mid w \mid x \mid y \mid z) \]

These names are just shorthands (macros) for regular expressions.
Recognizers for Regular Expressions

Example 1: integer constant
RE: digit+
FSA:

Example 2: identifier
RE: letter ( letter | digit )* 
FSA:

Example 3: Real constant
RE: digit*.digit+
FSA:
Next Lecture

Things to do:

• First homework is due Friday, September 22.
  Electronic submission of PDF files ONLY