Syntax Analysis
Ch 4 p 159-195
February 23, 2009
CS310 Problems

• Describe (in English) the language denoted by the regular expression 
  \(((\varepsilon|0)1^*)^*\)

• Write regular definitions for:
  – all strings that begin with an aa
  – all strings that contain aa
  – all strings that do not contain aa
  – All are over the alphabet \{a,b\}.

• Construct an NFA for the regular expression 
  \(((\varepsilon|a)b^*)^*\)
CFGs

expr -> expr op expr | ( expr ) | number | id

op -> + | - | *

• Backus-Naur Form

< expr > ::= < expr > < op > < expr > | (< expr >) | NUMBER

< op > ::= + | - | *

CS 480 – Spring 2009
Pacific University
Notation from the Book

• Terminals

• Nonterminals

• String of terminals

• Greek Letters

• Alternate Forms

• Start production
Derivations

- $\Rightarrow$
  - can derive with one application of a production

- $\Rightarrow^*$
  - can derive with zero or more applications of any productions

$E \rightarrow (E) \mid a$

Does $E \Rightarrow^* ((a))$?

Does $E \Rightarrow ((a))$?

Does $E \Rightarrow^* (a)(a)$?
Grammars

- \( G_1: A \rightarrow Aa \mid a \)
- \( G_2: B \rightarrow aB \mid a \)

- Do \( G_1 \) and \( G_2 \) describe the same language?
- Are both \( G_1 \) and \( G_2 \) equivalent to \( a^* \)?
- Are they ambiguous?
  - How fix?
- Right or Left recursive?
  - What problems could arise?
- Does \( A \rightarrow \epsilon \)
More...

• Give a CFG which generates sequences of one or more statements (s) separated by ;

  – (i.e. $L(G) = \{s \ s; s \ s; s \ s; \ldots\}$)

• Give a CFG which generates sequences of one or more statements where the semicolon is a terminator and not a separator (i.e. $L(G) = \{s; s; s; s; s; \ldots\}$)
Parsing!

\[
\begin{align*}
\text{expr} & \rightarrow \text{expr op expr} \mid (\text{expr}) \mid \text{number} \\
\text{op} & \rightarrow + \mid - \mid \ast \mid / 
\end{align*}
\]

• Problem?
  
  \[1 + 3 \ast 8\]
  
  Left most? Right most?

• Ambiguity:
  
  – Get rid of it \textbf{OR}
  
  – Use rules to limit its impact
More..

\[ \text{expr} \rightarrow \text{expr op expr | term} \]
\[ \text{op} \rightarrow + | - | * \]
\[ \text{term} \rightarrow \text{number} \]

• Ambiguous?
  – Why or why not?
  – Precedence?
stmt -> ifstmt | other
ifstmt -> if ( expr ) stmt |
       if ( expr ) stmt else stmt
expr -> T | F

• Thoughts?

• Fixes?
Immediate Left Recursion

• Immediate Left Recursion

\[
\begin{align*}
E & \rightarrow E + T \mid T \\
T & \rightarrow T * F \mid F \\
F & \rightarrow (E) \mid \text{id}
\end{align*}
\]

• Nonimmediate Left Recursion:

\[
\begin{align*}
S & \rightarrow Aa \mid b \\
A & \rightarrow Ac \mid Sd \mid e
\end{align*}
\]

Differences? Why is this important?

How do you remove each type?
Practice

S → Ba | b
B → Sa | a

• What is the language?
• Eliminate all the left recursion
  – Algorithm 4.1 on p 177