Plan for Today

Expression evaluation (PA3 sneak peek)

Syntax-directed expression evaluation
– example
– but what about grammar ambiguity?

Ambiguity and examples of removing it
– using JavaCUP precedence and left and right keywords

Start covering lexical analysis with JLex

Subset of MiniJava Expression Grammar

Subset

Expression ::=  
Expression ( "&&" | "<" | "+" | "-" | "+" ) Expression  
| <INTEGER_LITERAL>  | (" Expression ")

Full Expression Grammar

Expression ::=  
Expression ( "&&" | "<" | "+" | "-" | "+" ) Expression  
| Expression "[" Expression "]"  
| Expression "." "length"  
| Expression "." Identifier ") ( Expression ( "," Expression )* )? ")"  
| <INTEGER_LITERAL>  | "true"  | "false"  | Identifier  
| "this"  | "new" "int" "[" Expression "]"  | "new" Identifier ")"  
| "." Expression  | (" Expression ")
Parse Tree Example

```
expr
  __/      \\
   /      /
  expr    +  expr
  |       /
NUM(42)  /  
       \\
expr    *
       /
NUM(7)  

expr  
  __/  \\
   /   
  expr  expr
  |    |
NUM(7) NUM(6)
```

Semantic Rules for Expression Example

```
expr
  __/        \\
   /        
  expr     expr
  |        |
expr  +  expr
  |    |
expr  *
  |  |
NUM(42) expr  expr
  |    |
NUM(7)  

expr  
  __/  \\
   /   
  expr  expr
  |    |
NUM(7)  

expr  
  __/  \\
   /   
  expr  expr
  |    |
expr  *
  |  |
NUM(6)
```

```
Another valid parse tree