Lecture 15:

- MyPL AST Overview
- Associativity and Precedence (intro)

Announcements:

- HW-3 out
**MyPL AST Classes**

See the course website for diagram ...

Trickier parts:

- simple vs complex terms (latter for parenthesized expressions)
- if statements with (reusable) “If Parts”
- lvalues as path (list) of VarRef (var name and optional array expr)
- variable rvalues with similar path of VarRef
More on Context Free Grammars

With recursive descent parsers, it can be hard to ...

• define grammars with appropriate operator **associativity**
• define grammars with appropriate operator **precedence**
• ... and these are important for semantic analysis (and evaluation)

**Operator associativity**

• many operators are **left associative** ... e.g., ×, ÷, +, −
• For example ... 40 ÷ 10 ÷ 2 ≡ (40 ÷ 10) ÷ 2
• Can be captured by the grammar rule:
  \[ e \rightarrow e \div n \]  
  \( \ldots \) \( n \) a number

• and the "AST":

```
  /\  
 /   \ 
/ 2  
\40\10
```

• But notice this requires ulineleft recursion! \( \ldots \) so not \( LL(k) \)