

Ours

$\text{boolExp} \rightarrow \text{boolExp} \mid \langle \text{op} \rangle \mid \langle \text{op} \rangle$
 $\langle \text{op} \rangle \rightarrow \langle \text{op} \rangle \&\& \langle \text{op2} \rangle \mid \langle \text{op2} \rangle$
 $\langle \text{op2} \rangle \rightarrow \langle \text{op2} \rangle \mid \langle \text{op3} \rangle \mid \langle \text{op3} \rangle$
 $\langle \text{op3} \rangle \rightarrow (\langle \text{op3} \rangle) \mid \langle \text{id} \rangle$
 $\langle \text{id} \rangle \rightarrow \text{variable} \mid \langle \text{boolRel} \rangle$

Kooshes's

$\text{BoolExp} \rightarrow \text{BoolTerm} \text{ OR } \text{BoolExp} \mid \text{BoolTerm}$
 $\text{BoolTerm} \rightarrow \text{BoolFactor} \text{ AND } \text{BoolTerm} \mid \text{BoolFactor}$
 $\text{BoolFactor} \rightarrow \langle \text{id} \rangle \mid \text{not } \langle \text{BoolExp} \rangle \mid (\langle \text{BoolExp} \rangle)$
 $\langle \text{id} \rangle \rightarrow \text{variable name}$
 $\text{BoolFactor} \rightarrow ! \text{BoolFactor} \mid \text{BoolPrimary}$
 $\text{BoolPrimary} \rightarrow (\text{BoolExp}) \mid \text{id}$
 $\text{id} \rightarrow \text{variable}$

Level

Operator \rightarrow ~~L1~~ L2

L1 \rightarrow $> | < | > = | < =$

L2 \rightarrow $= = | ! = |$ L1

L3

$\langle \text{Bool Exp} \rangle \rightarrow \langle \text{Bool Exp} \rangle | \langle \text{Bool Exp} \rangle \text{ Operator } \langle \text{Bool Exp} \rangle$

$((a+b-c) \text{ ~~and~~ } (d+e)) \&\& (c < d)$

Combine

Rel Op Bool ^{Connectors} ~~Exp~~ Arith Exp

\hookrightarrow Bool Primary \rightarrow Arith Exp

Quiz on parsing program

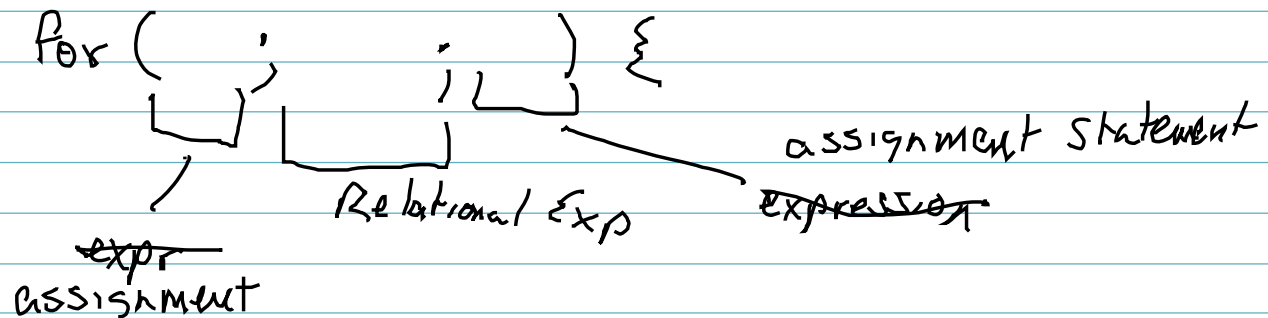
code for parser

Read / understand this code!

Today:

Extend interpreter.

First step add a for



expr that contains assignment, still an expr

$i = j = 2 ; \quad j = 2 \text{ then } i = j$
not in our language

Write production Rules

$$\begin{aligned}
\langle \text{arith_expr} \rangle &\rightarrow \langle \text{arith_term} \rangle \{ (+, -) \langle \text{arith_term} \rangle \} \\
\langle \text{arith_term} \rangle &\rightarrow \langle \text{arith_factor} \rangle \{ (*, /, \%) \langle \text{arith_factor} \rangle \} \\
\langle \text{arith_factor} \rangle &\rightarrow (\langle \text{arith_expr} \rangle) \mid - \langle \text{arith_expr} \rangle \mid \langle \text{id} \rangle \mid \langle \text{number} \rangle \\
\langle \text{id} \rangle &\rightarrow [-a-zA-Z][-a-zA-Z0-9]^* \\
\langle \text{number} \rangle &\rightarrow [0-9]^+
\end{aligned}$$

Fix

Look @ handout
 write grammar rules
 for
 assign

460 2/13/19

Semicolon to terminate statement- (not separate)

Proj 1 posted today

Write a few expressions
and use rules to parse them

$a >= b * c < d + 2$

~~$\langle \text{rel. expr} \rangle \rightarrow \langle \text{rel. term} \rangle$~~

~~$\langle \text{rel. term} \rangle \rightarrow \langle \text{rel. primary} \rangle >= \langle \text{rel. primary} \rangle$~~

~~$\text{rel. primary} \rightarrow \text{arith}$~~

rel expr
{

$(id) \rightarrow a$ $>=$ rel. primary $() = \langle \text{rel. primary} \rangle$
/
 ~~$\langle \text{arith term} \rangle \langle \text{mult op} \rangle \langle \text{arith primary} \rangle \langle \text{arith primary} \rangle$~~
arith term
 ~~$\langle \text{arith primary} \rangle \langle \text{mult op} \rangle \langle \text{arith primary} \rangle$~~
/
arith factor
id
b
*
arith factor
id
c
+
2