

Honor Code Notice Collaboration is a great way to learn, therefore you are encouraged to help one another out, but you must not cheat. When you cheat you only harm yourself. When collaborating with other students you may get help understanding course content, but may not get help completing course requirements unless specifically stated on the assignment. *See syllabus for the full honor statement.*

Answer the following questions. Turn in a single pdf file that contains all of your answers.

1. Start with the following grammar:

$$\begin{aligned} \langle \text{exp} \rangle &::= \langle \text{exp} \rangle + \langle \text{mulexp} \rangle \mid \langle \text{mulexp} \rangle \\ \langle \text{mulexp} \rangle &::= \langle \text{mulexp} \rangle * \langle \text{rootexp} \rangle \mid \langle \text{rootexp} \rangle \\ \langle \text{rootexp} \rangle &::= (\langle \text{exp} \rangle) \mid \mathbf{a} \mid \mathbf{b} \mid \mathbf{c} \end{aligned}$$

Modify as follow:

- (a) Add subtraction and division operators (- and /) with the customary precedence and associativity.

$$\begin{aligned} \langle \text{exp} \rangle &::= \langle \text{exp} \rangle - \langle \text{addexp} \rangle \mid \langle \text{addexp} \rangle \\ \langle \text{addexp} \rangle &::= \langle \text{addexp} \rangle + \langle \text{divexp} \rangle \mid \langle \text{divexp} \rangle \\ \langle \text{divexp} \rangle &::= \langle \text{divexp} \rangle / \langle \text{mulexp} \rangle \mid \langle \text{mulexp} \rangle \\ \langle \text{mulexp} \rangle &::= \langle \text{mulexp} \rangle * \langle \text{rootexp} \rangle \mid \langle \text{rootexp} \rangle \\ \langle \text{rootexp} \rangle &::= (\langle \text{exp} \rangle) \mid \mathbf{a} \mid \mathbf{b} \mid \mathbf{c} \mid \mathbf{d} \mid \mathbf{e} \end{aligned}$$

- (b) After performing a., add a left-associative operator % between + and * in precedence.

$$\begin{aligned} \langle \text{exp} \rangle &::= \langle \text{exp} \rangle - \langle \text{addexp} \rangle \mid \langle \text{addexp} \rangle \\ \langle \text{addexp} \rangle &::= \langle \text{addexp} \rangle + \langle \text{divexp} \rangle \mid \langle \text{divexp} \rangle \\ \langle \text{divexp} \rangle &::= \langle \text{divexp} \rangle / \langle \text{modexp} \rangle \mid \langle \text{modexp} \rangle \\ \langle \text{modexp} \rangle &::= \langle \text{modexp} \rangle \% \langle \text{mulexp} \rangle \mid \langle \text{mulexp} \rangle \\ \langle \text{mulexp} \rangle &::= \langle \text{mulexp} \rangle * \langle \text{rootexp} \rangle \mid \langle \text{rootexp} \rangle \\ \langle \text{rootexp} \rangle &::= (\langle \text{exp} \rangle) \mid \mathbf{a} \mid \mathbf{b} \mid \mathbf{c} \mid \mathbf{d} \mid \mathbf{e} \end{aligned}$$

- (c) After performing c., add a right-associative operator = at lower precedence than

any of the other operators.

$$\begin{aligned}
 \langle exp \rangle &::= \langle subexp \rangle = \langle exp \rangle \mid \langle subexp \rangle \\
 \langle subexp \rangle &::= \langle subexp \rangle - \langle addexp \rangle \mid \langle addexp \rangle \\
 \langle addexp \rangle &::= \langle addexp \rangle + \langle divexp \rangle \mid \langle divexp \rangle \\
 \langle divexp \rangle &::= \langle divexp \rangle / \langle modexp \rangle \mid \langle modexp \rangle \\
 \langle modexp \rangle &::= \langle modexp \rangle \% \langle mulexp \rangle \mid \langle mulexp \rangle \\
 \langle mulexp \rangle &::= \langle mulexp \rangle * \langle rootexp \rangle \mid \langle rootexp \rangle \\
 \langle rootexp \rangle &::= (\langle exp \rangle) \mid \mathbf{a} \mid \mathbf{b} \mid \mathbf{c} \mid \mathbf{d} \mid \mathbf{e} \mid \mathbf{f}
 \end{aligned}$$

2. Show that the following grammar is ambiguous:

$$\langle exp \rangle ::= \langle exp \rangle + \langle exp \rangle \mid \langle exp \rangle * \langle exp \rangle \mid (\langle exp \rangle) \mid \mathbf{a} \mid \mathbf{b} \mid \mathbf{c}$$

The grammar of the above statement is ambiguous as the repetitive use of the $\langle exp \rangle$ makes the associativity and precedence of the expression undetermined and the expression ambiguous.

3. State the binding time for each of the following for both C++ as well as Java. Feel free to include Python if you're curious:

- (a) The location in memory of a local variable in a function.
Java: Run time C++: Run time
- (b) The meaning of the keyword while.
Java: Language definition time. C++: Language definition time.
- (c) The size in memory of a variable of type int.
C++: Language implementation time. Java: Language implementation time.
- (d) The location in memory of a global static variable.
Java: Run time. C++: Load time.
- (e) The code for the printf (or equivalent) function.
Java: Link time. C++: Link time.
- (f) The type of a local variable in a function.
Java: Compile time. C++: Compile time.
- (g) The value(s) assigned to a variable.
Java: Compile time. (if constant var, run time) C++: Compile time.
- (h) The size in memory of a pointer.
Java: Language implementation time. C++: Run time.