

Homework 4

1. (a) Consider the following grammar:

$$\begin{aligned} E &\rightarrow E + T \mid E - T \mid T \\ T &\rightarrow T * F \mid T / F \mid F \\ F &\rightarrow (E) \mid \text{id} \end{aligned}$$

Why is this grammar not LL? Give a rightmost derivation and parse tree for the following expression:

$$7 - 8 * (3 + 2) / 4 + 11$$

Note that all the numbers are ids, as in class, and I'm asking for BOTH the derivation and the parse tree (which are not always identical).

- (b) Now consider an equivalent LL grammar:

$$\begin{aligned} E &\rightarrow TE' \\ E' &\rightarrow +TE' \mid -TE' \mid \epsilon \\ T &\rightarrow FT' \\ T' &\rightarrow *FT' \mid /FT' \mid \epsilon \\ F &\rightarrow (E) \mid \text{id} \end{aligned}$$

Give a leftmost derivation and parse tree for the same expression:

$$7 - 8 * (3 + 2) / 4 + 11$$

Again, note that all the numbers are ids, as in class, and I'm asking for BOTH the derivation and the parse tree (which are not always identical).

2. Consider the following LL grammar:

$$\begin{aligned} S &\rightarrow aB \mid bA \mid \epsilon \\ A &\rightarrow bAA \mid aS \\ B &\rightarrow aBB \mid bS \end{aligned}$$

- (a) Compute the FIRST and FOLLOW sets for each nonterminal.
- (b) Using the FIRST and FOLLOW sets, generate the predictive parsing table.
- (c) Show the parsing action (including the matches, stack, input and action columns) for the string: baab\$. Note that if your parsing does not work (which it should for this one, unless I've made a mistake), you should simply show the parsing action up to the point where it gets stuck.
3. (a) Write a context free grammar to accept the set of all regular expressions. (Hint: you just need to make rules for each of the operations that a regular expression is built from.) Is your grammar LL?
- (b) **Extra credit:** Now re-write your grammar for all regular expressions that will give Kleene closure the highest precedence (so lowest in the tree) and alternation (ORs) the lowest precedence.