Name SOLUTIONS

Tuesday, September 15

Email 6.818-www@mit.edu

6.818 Fall 2020 Miniquiz #6 5 Minutes

1. Consider the following grammar, which has no left recursion but is right-associative:

$$T \rightarrow x\ T'$$
$$T' \rightarrow *\ x\ T'$$
$$T' \rightarrow \varepsilon$$

What can you do to implement a top-down parser that constructs left-associative parse trees for the grammar above?

Option 1: First construct a right-associative concrete parse tree using the top-down parser, then rewrite the parse tree to make it left-associative.

Option 2: Have T' construct an incomplete parse tree with the leftmost child missing, and have the caller fill in the missing child (with x if the caller is T, or with a similarly incomplete subtree if the caller is also T').

2. Consider the following grammar:

$$E \rightarrow E + T \mid T$$
$$T \rightarrow T * F \mid F$$
$$F \rightarrow (\ E\ ) \mid x$$

What are the first five steps of a shift-reduce parser for the input $(\ x * x\ ) + x$? (Assume the parser won't take a step that would lead to an invalid string.)

1) SHIFT (
2) SHIFT x
3) REDUCE F
4) REDUCE T
5) SHIFT *