Parsing and Semantic Actions
Interlude: Ambiguous Grammars

Grammar: \[ S \rightarrow S + S \mid \text{<int>} \]
String: \[ 1+2+3 \]
Two possible parse trees:

Disambiguated: \[ S \rightarrow S + \text{<int>} \mid \text{<int>} \]
Grammar of Roman Numerals

\[
S \rightarrow xTU \mid IX \mid X \\
T \rightarrow c \mid l \\
X_1 \rightarrow xX_2 \mid U \\
U \rightarrow iY \mid vI \mid I \\
Y \rightarrow x \mid v \\
I_1 \rightarrow ili_2 \mid \epsilon
\]

• Is the grammar ambiguous? Why not?
• Give a leftmost derivation of xlii
  • Always expand leftmost non-terminal
• Draw a parse tree
• Write semantic actions to evaluate the value
  • Start by assigning values to the terminals
• Show an evaluation trace
LL(1) Parsers
• Is this grammar LL(1)?
  • LL(1) parsers pick a production based on the next terminal

• No:
  • If we’re expanding S and see a 1, multiple productions match
  • A is left-recursive. If we expand the second production, we end up in the same state. Infinite loop!

• Make it LL(1)
  • First problem is solved with left-factoring
  • Second problem is solved with left-recursion elimination
Now that we think we have an LL(1) grammar, we move on to the next question:

- Write the *First* and *Follow* sets
  - *Fixpoint iteration*: Repeat the rules until nothing changes
  - *Follow* Tip: Make a column of $\subseteq$ relations
- Construct the LL(1) parsing table
- Show all the steps to parse: 1 1 0 2 0 3 0 1 0 3 3
We’re done!

Any questions?

Good luck with the assignment.