COMP 302 Lecture 10 03 October 2016

Backus-Naur Form Grammars

Set of rules.

- LHS ::= RHS.
- LHS is composed of a single (context-free) non-terminal
- RHS is composed of non-terminals, terminals (tokens that the scanner returns), and operators. '|' and 'ε'.

ex: *see 10c302*

- There are multiple legal parse trees of this grammar for this assign_stnt → ambiguous grammar.
- It would be nice if anything could only be divided up one single way. We should not leave this ambiguity. We try and avoid it.
- We do understand that ' * > + '.
- We can fix this in the grammar.
    - *see 10c302-1*

2 grammar subsets (parsing algorithms) used.

- LL, LR. The L always means 'left-to-right'.
- L is leftmost, R is rightmost.
- They will apply rules to a stream of tokens in two different ways.
- One produces the parse as it descends, the other produces the parse as it comes back up.
- LL is easier to implement by hand. LR is a little bit more general but also more complex.
- LR uses tools.

LL

- LL parsers are also known as top-down parsers. We parse it as we recursively descend.
- At each stage, we will 'predict' the next rule to apply. *see 10c302-2*
- The above is an LL parser.
- We may notice we looked ahead a token each time to decide which rule to apply.
- It is an LL (1) parser, with 1 being the number of tokens to look ahead.

LR-parsing

- Shift-reduced parser *see 10c302-3*

LL (1) parser

- Every RHS of a rule should start with a terminal.
- Avoid left-recursion. *see 10c302-4*

To build a parser

see 10c302-5