Grammar

- Ambiguity
- Precedence and associativity

Equivalent

How to prove L(G1) = L(G2)?

- Simplify
- Chomsky normal form
- Push down automata

Parsing

Leftmost Derivation

- Top-down
- Recursive Descent, LL(1)

Recursive Descent

LL(1)

- How to LL(1) with M
- Why M?
  - why First Set?
  - why Follow Set?
- And how to write the above formally?
- Why it doesn't work for all Grammar? [non-deterministic]

LR

- Bottom-up
- Shift-reduce [deterministic]
- LR(0), SLR(1), LR(1)

LR(0)

- Why it's [non-deterministic]?
- shift / reduce conflict ; NFA

SLR(1)

- SLR(1) parsers use the same LR(0) configuration sets and have the same table structure and parser operation
- the difference comes in assigning table actions, where we are going to use one token of lookahead to help arbitrate among the shift-reduce conflicts.
- How to construct a Full DFA?
    - Do it directly with epsilon closure.
    - Power set / subset construction for converting NFAs to DFAs. (Time consuming)
- Why it's [non-deterministic]?
    - shift-reduce / reduce-reduce conflicts

LR(1)

- Solved the above problem by looking ahead 1 symbol, but more complex DFA

More in Formal Model of Language

- Earley Parser
- Chart Parsing
- Dependency Grammar/Parsing
- Categorical Grammar/Parsing