# Grammar

- Ambiguity
- Precedence and associativity

## Equivalent

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

- Simplify
- Chomsky normal form
- Push down automata

# Parsing (Terminal) Left to right

| - | Leftmost | Rightmost Derivation |
|---|---|---|
| | Top-down | Bottom-up |
| | Recursive Descent, LL(1) | LR(0), SLR(1), LR(1) |
| | Predict-match | Shift-reduce [deterministic] |

### LL(1)

- Why M?
    - why First Set, Follow Set?
- And how to write the above formally?
- Why it has non-determinism for some Grammar?
    - Left recursion

$$A \Rightarrow^{+} A\alpha$$

    - Add EOF to S
    - Add extra `D = then C | epsilon`

### LR(0)

- Why it has non-determinism for some Grammar?
    - shift-reduce conflict
        - both *shift* $A \to \alpha \bullet a\beta$ and *reduce* $B \to \beta \bullet$, $A, B \in N$
        - and no matter whether $a \in FOLLOW(B)$ in an NFA state set.

$$S \to L\bullet := R$$

$$R \to L\bullet$$

- reduce-reduce conflicts
  - see below
- NFA

### SLR(1)

- LR(0) table structure
  - same parser operation (shift/ reduce)
  - **one token of lookahead**
    - to arbitrate among shift-reduce conflicts
  - **DFA (less non-determinism allowed)**
- 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 has non-determinism for some Grammar?
  - shift-reduce
    - both *shift* $A \to \alpha\bullet a\beta$ and *reduce* $B \to \beta\bullet$, $A, B \in N$
    - and $a \in FOLLOW(B)$ in an NFA state set.

$$S \to L\bullet := R$$

$$R \to L\bullet, (:=) \in FOLLOW(R)$$

  - reduce-reduce
    - both *reduce* $A \to \alpha\bullet$ and *reduce* $B \to \beta\bullet$ , $A, B \in N$
    - and $\exists a. a \in FOLLOW(A)$ and $a \in FOLLOW(B)$ in an NFA state.
    - In particular, hold if $A = B$.

### LR(1)

- *shift* $A \to \alpha\bullet a\beta, b$ and *reduce* $B \to \beta\bullet, b$
  - **reduce** only if the next token is exactly terminal $b$ as $b \in first(B)$ rather than any of those in $Follow(B)$ in SLR(1) parsing.

$$R \to L\bullet, b$$

$$S \to L\bullet := R, b$$

- but causing more complex DFA

## More in Formal Model of Language

- Earley Parser
- Chart Parsing

- Dependency Grammar/Parsing
- Categorical Grammar/Parsing