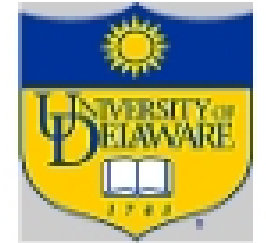




Parsing III

Bottom-up Parsing



Parsing Techniques

Top-down parsers (LL(1), recursive descent)

- Start at the root of the parse tree and grow toward leaves
- Pick a production & try to match the input
- Bad "pick" \Rightarrow may need to backtrack
- Some grammars are backtrack-free *(predictive parsing)*

Bottom-up parsers (LR(1), operator precedence)

- Start at the leaves and grow toward root
- As input is consumed, encode possibilities in an internal state
- Start in a state valid for legal first tokens
- Bottom-up parsers handle a large class of grammars



Bottom-up Parsing

(definitions)

The point of parsing is to construct a derivation

A derivation consists of a series of rewrite steps

$$S \Rightarrow \gamma_0 \Rightarrow \gamma_1 \Rightarrow \gamma_2 \Rightarrow \dots \Rightarrow \gamma_{n-1} \Rightarrow \gamma_n \Rightarrow \textit{sentence}$$

- Each γ_i is a sentential form
 - If γ contains only terminal symbols, γ is a **sentence** in $L(G)$
 - If γ contains ≥ 1 non-terminals, γ is a **sentential form**
- To get γ_i from γ_{i-1} , expand some NT $A \in \gamma_{i-1}$ by using $A \rightarrow \beta$
 - Replace the occurrence of $A \in \gamma_{i-1}$ with β to get γ_i
 - In a leftmost derivation, it would be the first NT $A \in \gamma_{i-1}$

*A **left-sentential form** occurs in a leftmost derivation.*

*A **right-sentential form** occurs in a rightmost derivation.*