CFG => PDA

Sipser 2 (pages 99-118)

Formally...

• A pushdown automaton is a sextuple $M = (Q, \Sigma, \Gamma, \delta, q_0, F)$, where
  - $Q$ is a finite set of states
  - $\Sigma$ is a finite alphabet (the input symbols)
  - $\Gamma$ is a finite alphabet (the stack symbols)
  - $\delta: (Q \times \Sigma \times \Gamma) \rightarrow P(Q \times \Gamma)$ is the transition function
  - $q_0 \in Q$ is the initial state, and
  - $F \subseteq Q$ is the set of accept states
Balanced brackets

- Let $M = (Q, \Sigma, \Gamma, \delta, q_0, F)$, where
  - $Q = \{q_1, q_2, q_3\}$
  - $\Sigma = \{[, ]\}$
  - $\Gamma = \{[, $\}$
  - $q_0 = q_1$
  - $F = \{q_1, q_3\}$
  - $\delta$ is given by the transition diagram:

Finite automata and Pushdown automata

- Regular languages
- Pushdown languages
- ?
Regular => Pushdown

• Proposition: Every finite automaton can be viewed as a pushdown automaton that never operates on its stack.

• Proof:
Let \( M = (Q, \Sigma, \delta, q_0, F) \) be a finite automaton.
Define \( M' = (Q, \Sigma, \Gamma, \delta', q_0, F) \), where...

finite automata and Pushdown automata

? pushdown languages

regular languages
Pushdown automata are nondeterministic

- Build a machine to recognize
  
  \( PAL = \{ww^R \mid w \in \{0, 1\}^*\} \)

Recognizing context-free languages

- Lemma 2.21: If a language is context-free, then some pushdown automaton recognizes it.
- Proof:
For example...

- Let's use this construction on:
- \( G = (V, \Sigma, R, S) \), where
  - \( V = \{S\} \)
  - \( \Sigma = \{[, ]\} \)
  - \( R = \{ S \rightarrow_G \varepsilon, S \rightarrow_G SS, S \rightarrow_G [S]\} \)