Chapter 15: CFG = PDA

I. Theory of Automata

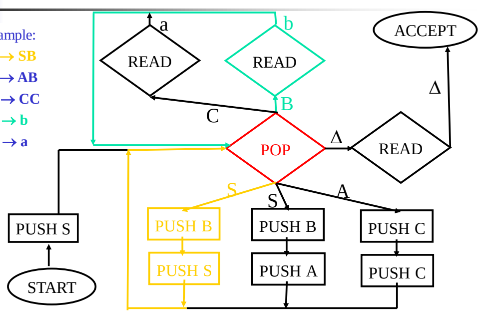
→ II. Theory of Formal Languages

III. Theory of Turing Machines …

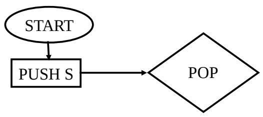
- Theorem1. Given a CFG that generates the language L, there is a PDA that accepts the language L.

- Theorem2. Given a PDA that accepts the language L, there is a CFG that generates the language L.

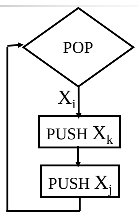
Example:
$S \to SB$
$S \to AB$
$A \to CC$
$B \to b$
$C \to a$



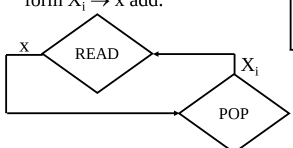
- Proof: by constructive algorithm (CFG → PDA)
  1. First, transform the grammar to one in Chomsky normal form. All productions are of the form:
     $X_i \rightarrow X_j X_k$
     $X_i \rightarrow x$
  2. Add



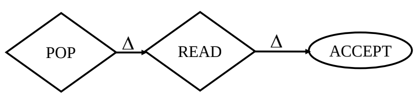
3. For each production of the form $X_i \rightarrow X_j X_k$ add:



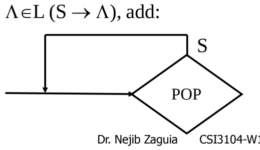
4. For each production of the form $X_i \rightarrow x$ add:



5. Add:

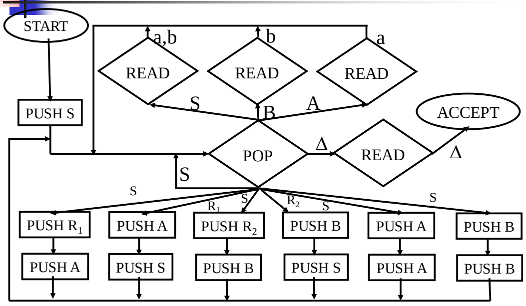


6. If $\Lambda \in L$ ($S \rightarrow \Lambda$), add:



$S \to AR1 \mid BR2 \mid AA \mid BB \mid a \mid b \mid \Lambda$

$A \to a \quad B \to b \quad R1 \to SA \quad R2 \to SB$



- Theorem. Given a PDA that accepts the language L, there is a CFG that generates the language L.
  - See the book for the proof

PDA and CFG are equivalent