**Theory of Computation, CSCI 438 spring 2022**
**Pushdown automaton to a context-free grammar, pages 119-123**
**Feb. 28**

A language is context-free iff some pushdown automaton recognizes it (Theorem 2.20, page 117)

(if, $\Leftarrow$)
If a PDA recognizes some language, then the language is context-free (Lemma 2.27, page 121)

Given a PDA translate it to a CFG.

Step 1: Translate the PDA to an equivalent PDA with:
1. Only 1 accept state, $q_{accept}$
2. Empties the stack before accepting
3. Every transition either pushes something onto the stack or pops something from it. No transition both pushes and pops

Step 2: Create a grammar for the new PDA as follows:
Let $P=(Q, \Sigma, \Gamma, \delta, q_0, \{q_{accept}\})$ and construct a grammar with variables $\{V_{pq} \mid p,q \in Q\}$, terminal symbols $\Sigma$, start variable $V_{q0,qaccept}$ and the following rules:

1. For each $p, q, r, s \in Q$, $t \in \Gamma$, and $a,b \in \Sigma_\varepsilon$, if $\delta(p,a,\varepsilon)$ contains $(r,t)$ and $\delta(s,b,t)$ contains $(q, \varepsilon)$, add the rule $V_{pq} \rightarrow aV_{rs}b$.
2. For each $p, q, r \in Q$, add the rule $V_{pq} \rightarrow V_{pr}V_{rq}$.
3. For each $p \in Q$, add the rule $V_{pp} \rightarrow \varepsilon$.