The Overall Procedure I

- Given

\[ P = (Q, \Sigma, \Gamma, \delta, q_0, \{q_{accept}\}) \]

- Construct a CFG \( G \)

\[ \text{var}(G) = \{A_{pq} \mid p, q \in Q\} \]

- Start variable:

\[ A_{q_0,q_{accept}} \]

- Rules: see earlier slides
Needed modifications of PDA I

- Recall we need PDA to satisfy
  1. Single accept state
  2. Stack empty before accepting
  3. Each transition push or pop, but not both

- Let’s handle the first two together: single accept and stack empty before accepting:
- A new start $q_s \rightarrow q_{s'}$ with $\epsilon, \epsilon \rightarrow \$\$
- For any $q \in F$, we have $\epsilon, a \rightarrow \epsilon$ back to $q$, $\forall a$. This pops things out before accepting a string
- Then from any $q \in F$, we do $\epsilon, \$ \rightarrow \epsilon$ to $q_a$. 
Needed modifications of PDA II

- $q \in F$ are no longer accept states
- See the illustration in the following figures
- Original PDA:

```plaintext
\[ q_s' \rightarrow \ldots \rightarrow q_1 \rightarrow \ldots \rightarrow q_2 \]
```
Needed modifications of PDA III

New:

\[
\begin{align*}
q_s &\quad \epsilon, \epsilon \rightarrow \$ \\
q_s' &\quad \epsilon, \epsilon \rightarrow \$ \\
q_1 &\quad \epsilon, \$, \epsilon, b \rightarrow \epsilon \\
q_2 &\quad \epsilon, \$, \epsilon, a \rightarrow \epsilon \\
q_a &\quad \epsilon, \$, \epsilon, b \rightarrow \epsilon
\end{align*}
\]
Needed modifications of PDA IV

To have each transition push or pop, but not both, change

\[ q_1 \rightarrow q_2 \text{ with } a, a \rightarrow b \]

to

\[ q_1 \rightarrow q_3, a, a \rightarrow \epsilon \]
\[ q_3 \rightarrow q_2, \epsilon, \epsilon \rightarrow b \]

and change

\[ q_1 \rightarrow q_2, a, \epsilon \rightarrow \epsilon \]

to

\[ q_1 \rightarrow q_3, a, \epsilon \rightarrow ? \]
\[ q_3 \rightarrow q_2, \epsilon, ? \rightarrow \epsilon \]
Regular language is context Free I

- We roughly know this but didn’t give a formal proof. Here are the steps
- Regular language \(\Rightarrow\) recognized by DFA (in Chapter 1)
- DFA is a PDA
- Thus regular language recognized by PDA
- Then any regular language is context free (by the proof in this chapter)
Non-context free languages I

- There are such languages
- We omit the discussion