REG = \{ L \mid L \text{ is regular, i.e.} \}
\begin{align*}
L &= L(M) \text{ for some DFA } M \\
CFL &= \{ L \mid L \text{ is context-free, i.e.} \}
\begin{align*}
L &= L(G) \text{ for some CFG } G
\end{align*}
\end{align*}

THM \quad REG \subseteq CFL

IDEA:

DFA: \begin{array}{c}
a \rightarrow q_j \\
B_i \rightarrow q_j
\end{array}

CFG:
\begin{align*}
A_i \rightarrow aA_j \\
A_k \rightarrow \varepsilon \mid bA_m
\end{align*}

To accept: the symbol that corresponds to the accept state generates \( \varepsilon \). (It can also generate other things, if it has transitions going out of it.)

PR Let \( L \in \text{REG} \)
\begin{align*}
\exists \text{DFA } M \text{ s.t. } L(M) &= L \\
M &= (Q, \Sigma, \delta, q_0, F)
\end{align*}
Create CFG \( G_M (V, \Sigma, R, S) \)

\begin{align*}
V &= \{ A_0, A_1, \ldots, A_l | q_1 \}
\Sigma &= \Sigma \\
S &= A_0
\end{align*}

\begin{align*}
R &= \bigcup \{ A_i \rightarrow \varepsilon \mid \delta(q_i, a) = q_j \} \\
&\cup \{ A_k \rightarrow \varepsilon \mid q_k \in F \}
\end{align*}

\begin{align*}
L &= L(M) = L(G_M) \implies \text{REG} \subseteq \text{CFL}
\end{align*}
CFL $\subseteq$ REG

CFL $\subseteq$ REG

Question: Do there some kind of machine that can recognize CFL's?

Can we add some data structure to DFA/NFA to accept $L$?

How about a stack?

Example: $L = \{0^n1^n | n \geq 0\}$

Let all the zeros on the stack. For every 1, pop a 0. If stack empty at end (and you didn't run out too soon), accept.

DFA/NFA

Add a Stack (PUSHDOWN AUTOMATA)

Post/Push (LIFO)
NFA
\[ \delta(q_i, a \lor \varepsilon) = \{ q_j, q_k, \ldots \} \]

PDA
\[ \delta(q_i, a \lor \varepsilon, b \lor \varepsilon) = \{ (q_j, a \lor \varepsilon), (q_k, d \lor \varepsilon), \ldots \} \]

Each state has its own copy of the stack.

Graph:
- State transitions:
  - \( q_0 \) to \( q_1 \) on input 'c'
  - \( q_1 \) to \( q_3 \) on input 'a'
  - \( q_3 \) to \( q_5 \) on input 'a'
- Final state: \( q_6 \) with empty stack

Input Notation:
- Transition graph:
  - \( q_i \rightarrow q_j \) on input 'a'
  - \( q_i \rightarrow q_j \) on input 'b'
  - Transition functions:
    - \( \text{pop} \rightarrow \text{push} \)
- Transition rules:
  - \( \varepsilon, \varepsilon \rightarrow \) on state \( q_0 \)
  - \( 0, \varepsilon \rightarrow 0 \) (read and push REJECT)
  - \( 1, 0 \rightarrow E \) (read 1 and pop 0)
  - \( \varepsilon, \varepsilon \rightarrow E \) on state \( q_3 \)

Example:
- \( L = \{ 0^n 1^n \mid n \geq 0 \} \)