Theory of Computation, CSCI 438 spring 2022
Pushdown automaton, pages 111-116, Feb. 23

Nondeterministic PDA is a 6-tuple $M=(Q, \Sigma, \Gamma, \delta, q_0, F)$ where Q, Σ, $\Gamma$and F are all finite sets, and

1. Q is the set of states
2. Σ is the input alphabet
3. Γ is the stack alphabet
4. $\delta: Q x \Sigma_\varepsilon x \Gamma_\varepsilon \rightarrow P(Q x \Gamma_\varepsilon)$ is the transition function
5. $q_0 \in Q$ is the start state
6. $F \subseteq Q$ is the set of accept states. (Definition 2.13, page 113)

For the transition function δ: $Q x \Sigma_\varepsilon x \Gamma_\varepsilon \rightarrow P(Q x \Gamma_\varepsilon)$

- Γ_ε in the domain - what is seen on top of the stack. ε indicates not to look at the stack. In order to look at the top of the stack the stack must be popped.
- Γ_ε in the range - what to push onto the stack. ε indicates not to push anything.
- The stack is just temporary storage. Once the string is consumed and the PDA is in the accept state, the string is accepted regardless of the state of the stack.
- Your program should never try to look at an empty stack.

Review:

Regular Languages:
Finite automata

```
  input tape
 |__|__|__|__...
 ____↑_______
|  finite control |
|                 |
|_________________|
```

$M=(Q, \Sigma, \delta, q_0, F)$

DFA $\delta: Q x \Sigma \rightarrow Q$
NFA $\delta: Q x \Sigma_\varepsilon \rightarrow P(Q)$

Context Free Languages:
Nondeterministic pushdown automata

```
           input tape
|__|__|__|__...
___↑_________
| finite control  |      |___|
|                 |→     |___|  stack (can be infinite)
|_________________|      |___|
```

$M=(Q, \Sigma, \Gamma, \delta, q_0, F)$
↑
Finite set of stack symbols

Deterministic PDA $\delta: Q x \Sigma x \Gamma_\varepsilon \rightarrow Q x \Gamma_\varepsilon$
Nondeterminisistic PDA $\delta: Q x \Sigma_\varepsilon x \Gamma_\varepsilon \rightarrow P(Q x \Gamma_\varepsilon)$

Deterministic PDAs define a proper subset of context free languages. For now we'll discuss nondeterministic PDAs.

What it means for a machine to "accept" a string:
M=(Q, Σ, Γ, δ, q_0, F) accepts string w, where w = $w_1w_2 \ldots w_n$ and each $w_i \in \Sigma_\varepsilon$ and sequences of states r_0 r_1 r_2 …, r_m ∈Q and strings s_0, s_1, s_2, …, s_m ∈Γ* exists such that

1. $r_0 = q_0$ and $s_0 = \varepsilon$ i.e. M starts out properly
2. For i=0, 1, …, m-1 $(r_{i+1}, b) \in \delta(r_i, w_{i+1}, a)$ where s_i =at and s_{i+1} =bt for some $a,b \in \Gamma_\varepsilon$ and $t \in \Gamma^*$
3. $r_m \in F$.

(I won't ask you to give this on an exam.)