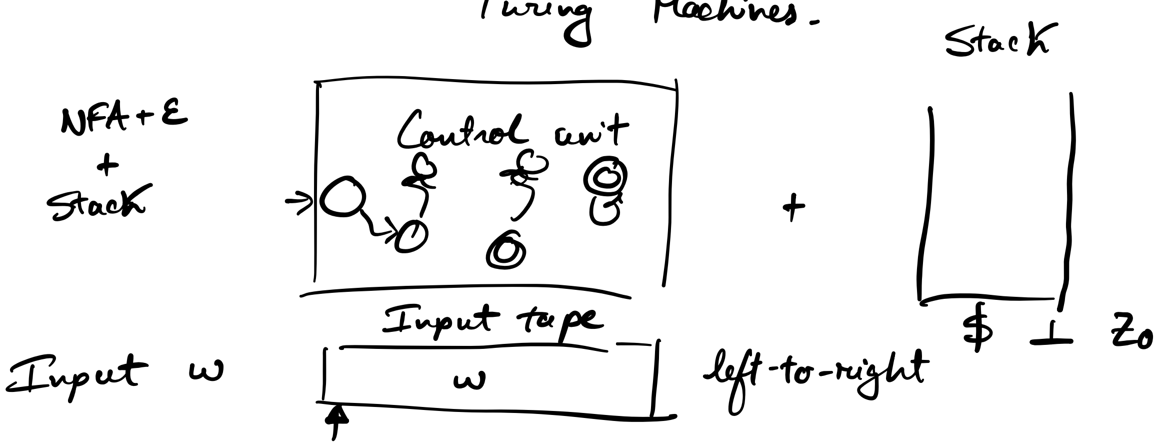
# Comp 330 - Lecture 16 - Oct 26th

IE: Tutto fumo e niente arrosto
All smoke and no roast
All talk & no action

→ UCORE Fri 3PM to 6PM
Trottier 2nd Floor
→ Prakash Nov 7th - 6PM - 7:30PM
- MC 204

## Pushdown Automata (PDA)

NPDA ↔ Equivalent to CFG
1. New tool to prove that L is CF
2. Working our way towards the most expressive comp. model: Turing Machines.



(*) In **N**PDAs, NPDA can have choices in its transition function => NPDA guesses & verifies.

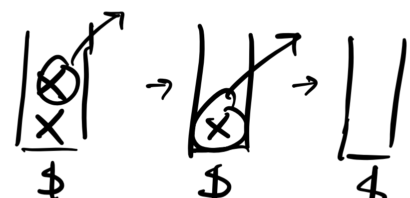
(*) DPDA are not equivalent in comp. power to NPDAs
(D = deterministic)

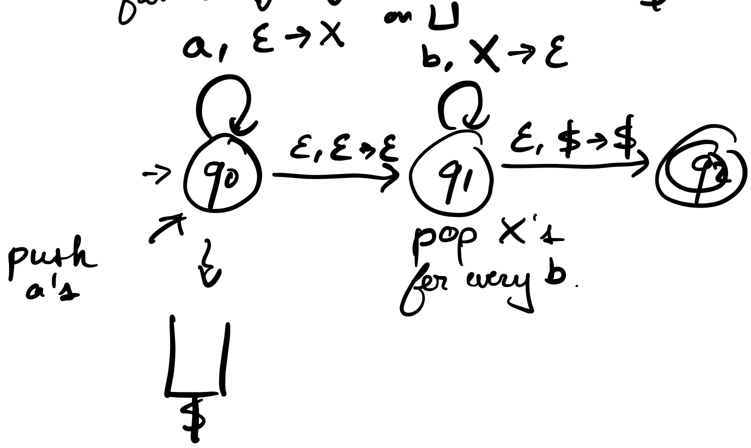
**Ex** Consider the following NPDA which accepts $L = \{a^n b^n : n \in \mathbb{N}\}$

Input: aa bb □

pop nothing from ⊔

push on X on ⊔




Is the stack of infinite size? Unbounded!

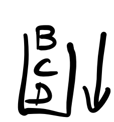
If the stack was bounded, then you would have an NFA+ε

Transitions are labelled

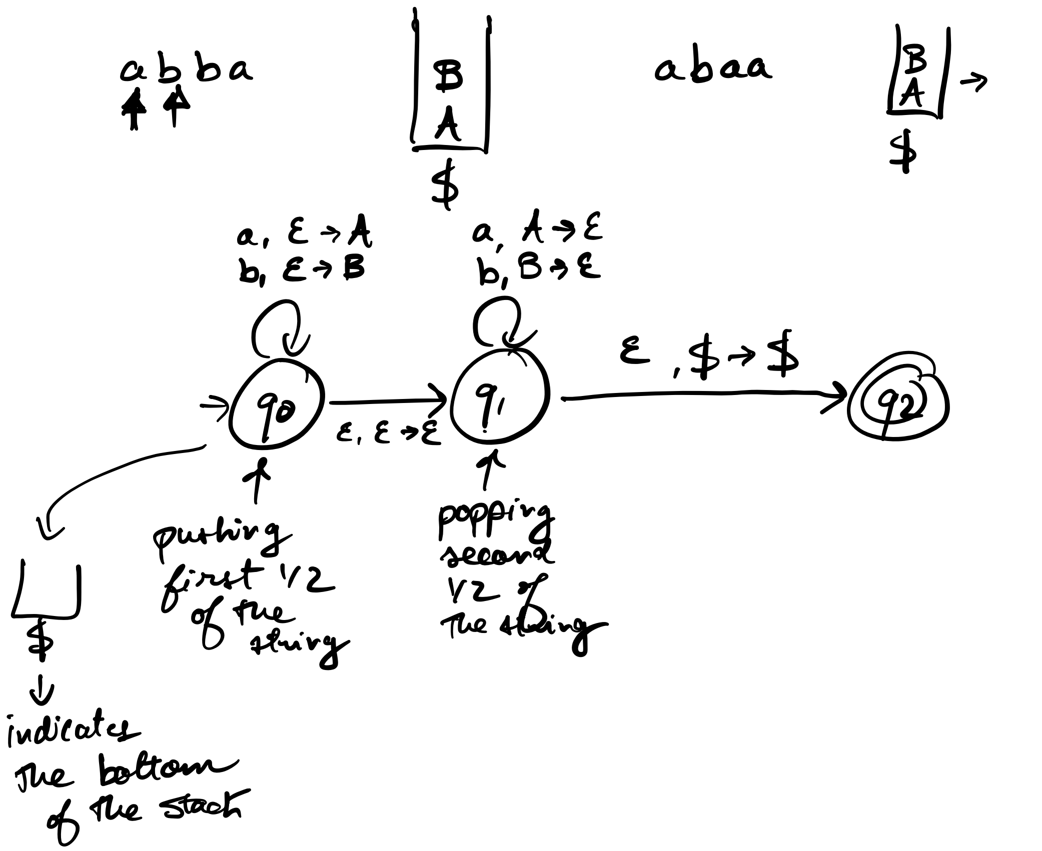
$\sigma, A \to \gamma$

σ: letter you read; A: what you pop; γ: what you push

$\gamma = \overrightarrow{BCD}$



**Ex** Design NPDA which accepts the language $L = \{ ww^R : w \in \{a,b\}^* \}$.




**Def** (PDA) A PDA is a 7-tuple $P = (Q, \Sigma, \Gamma, \delta, s, Z, F)$ where

$Q$ is the finite set of states

$\Sigma$ is the input alphabet, $\varepsilon \notin \Sigma$

$\Gamma$ is the stack alphabet, $\varepsilon \notin \Gamma$

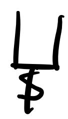
$\delta$ is the transition function

$$\delta : Q \times \underbrace{(\Sigma \cup \{\varepsilon\})}_{= \Sigma_\varepsilon} \times \underbrace{(\Gamma \cup \{\varepsilon\})}_{= \Gamma_\varepsilon}$$

$\to \mathcal{P}_{fin}(Q \times \Gamma^*)$    if $A \in \mathcal{P}_{fin}(B)$ then $A \subseteq B$ & $|A| < \infty$

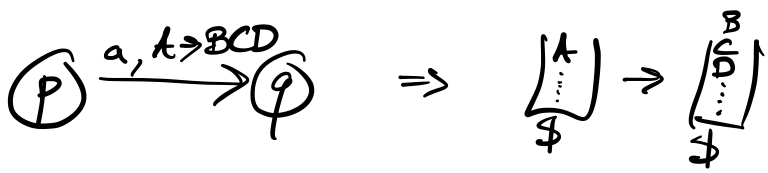
$s \in Q$ is the start state

$Z \in \Gamma$ is the special bottom of the stack marker $\Rightarrow$ Assume that $P$ starts with




$F \subseteq Q$ is the set of accept states.

**Def** (NPDA) A NPDA is a PDA with no restrictions on $\delta$.



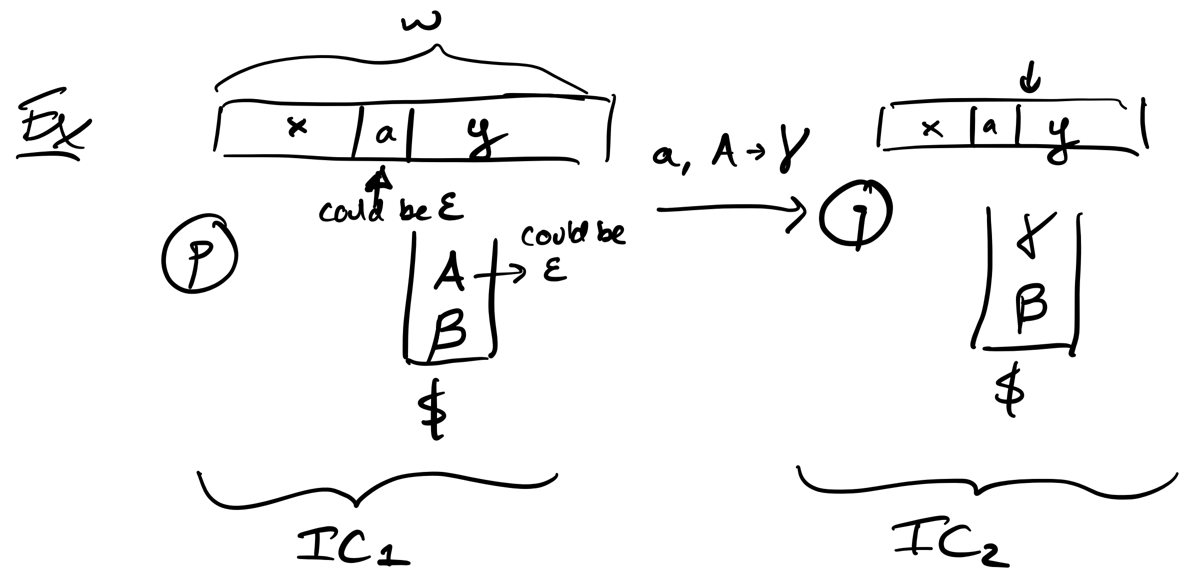

## Instantaneous configurations

**Def** (IC) An instantaneous config of a PDA $P$ is an element of

$$\underbrace{Q}_{\text{state } P \text{ is at}} \times \underbrace{\Sigma^*}_{\text{what's left to read}} \times \underbrace{\Gamma^*}_{\text{what's on stack}}$$

It completely describes $P$ at any given moment of its computation.

**Ex**




$IC_1 = (p, ay, A\beta\$) \overset{1}{\underset{P}{\vdash}} IC_2 = (q, y, \gamma\beta\$)$

What is the start IC of a PDA?

$(s, \underbrace{\underline{w}}_{\text{input string}}, \$)$

Acceptance config:

$(\underset{\in F}{f}, \varepsilon, \underset{\in \Gamma^*}{\alpha})$

**Def** ($\overset{1}{\underset{P}{\vdash}}$, next-one instant. config relation).

Given a PDA $P$, $p, q \in Q$, $a \in \Sigma_\varepsilon$, $\alpha \in \Gamma_\varepsilon$, $\gamma \in \Gamma^*$

If $(q, \gamma) \in \delta(p, a, \alpha)$ then $\forall y \in \Sigma^*, \beta \in \Gamma^*$

$(p, ay, \alpha\beta) \overset{1}{\underset{P}{\vdash}} (q, y, \gamma\beta)$

**Def** ($\vdash_P^n$) Given a PDA $P$, $n \in \mathbb{N}$, and two instant. configs. $C$ & $D$ Then

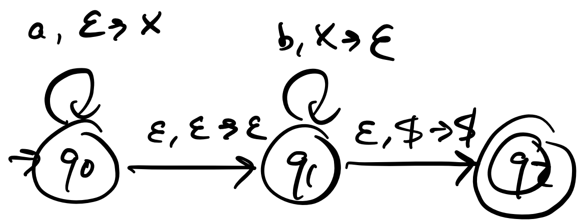
<u>Base case</u> $\quad C \vdash_P^0 C \quad \forall C$

<u>Inductive case</u> $\quad C \vdash_P^{n+1} D$ if $C \vdash_P^n E$ & $E \vdash_P^1 D$

**Def** ($\vdash_P^*$) Given PDA $P$ & ICs $C, D$,
$C \vdash_P^* D$ if $\exists n \in \mathbb{N}$ s.t. $C \vdash_P^n D$.

<u>Ex</u>

$P$:




One possible sequence of instant. configs:

$w = ab$

$$(q_0, ab, \$) \vdash_P^1 (q_0, b, X\$)$$

$$\vdash_P^1 (q_1, b, X\$)$$

$$\vdash_P^1 (q_1, \varepsilon, \$)$$

$$\vdash_P^1 (q_2, \varepsilon, \$)$$

acceptance config
=> ab is accepted

by P.

**Def** Given a PDA $P$, the language accepted by $P$ is

$$L(P) = \{ w \in \Sigma^* : \exists f \in F, \gamma \in \Gamma^* \quad (s, w, \$) \vdash^* (f, \varepsilon, \underline{\gamma}) \}$$

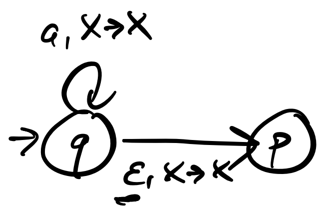
don't care what's on stack at the end.

**Def** A DPDA is a PDA with the following restrictions on $\delta$:

$\forall q \in Q, \ a \in \Sigma_\varepsilon, \ X \in \Gamma_\varepsilon$

1. $|\delta(q, a, X)| \leq 1$
2. if $\delta(q, a, X) \neq \emptyset$ for some $a \in \Sigma$ then $\delta(q, \varepsilon, X) = \emptyset$.

Why 2.




$IC = (q, a \ldots, X \ldots)$

$(q, \ldots, X \ldots)$ OR $(p, \ldots, X \ldots)$

DPDA are strictly less expressive than NPDAs

e.g. $L = \{ w w^R : w \in \Sigma^* \}$

<u>Exercise</u> Make a DPDA which accepts $\{a^n b^n : n \in \mathbb{N}\}$

**Thm** $\Sigma \neq \emptyset$, $\mathcal{L}_{NPDA} = \{ L \subseteq \Sigma^* :$
$\exists$ NPDA $P$ s.t.
$L(P) = L \}$

Then $\mathcal{L}_{NPDA} = \mathcal{L}_{CFL}$.

Implication: To prove that $L$ is CF
1. Give a CFG $G$ $L(G) = L$
+ Prove correctness

NEW! 2. Give a PDA $P$ $L(P) = L$
+ Prove correctness
↳ Make a statement

① if $(s, w, \$) \vdash^* (q_1, x, z\$)$ then
$x = \ldots$ & $z = \ldots$
$\wedge$ $(s, w, \$) \vdash^* (q_2, x, z\$)$ then
$x = \ldots$ & $z = \ldots$

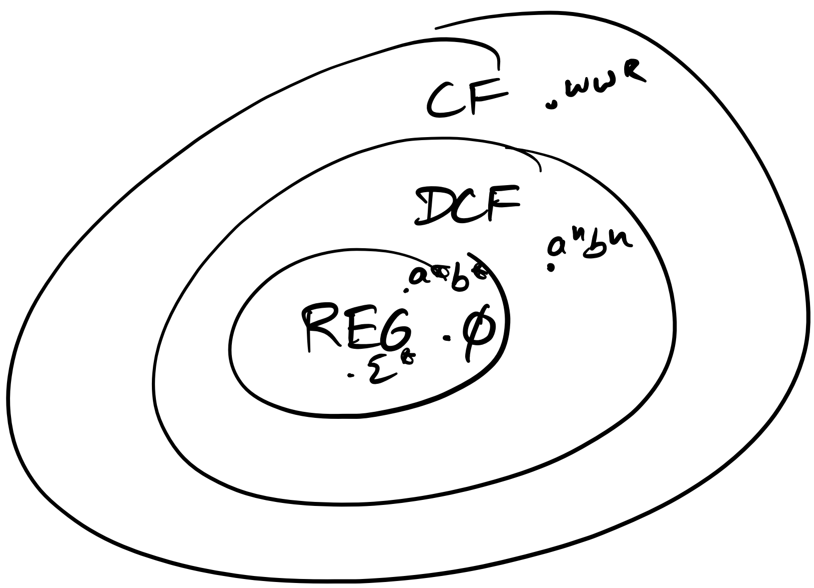
$\mathcal{L}_{CFL} = \mathcal{L}_{NPDA}$

Pf. $\mathcal{L}_{CFL} \subseteq \mathcal{L}_{NPDA} \Rightarrow$ Create a PDA which using its stack simulates the leftmost derivation of a string in $L(G)$.

$\mathcal{L}_{NPDA} \subseteq \mathcal{L}_{CFL} \Rightarrow$ For every pair $S^{(p,q)}$ of states in NPDA $P$, design a variable which generates all the strings that can transition from p to q.

Aside $L_{DPDA} = \{ L \subseteq \Sigma^*: \exists \text{ DPDA } P \text{ s.t. } L(P) = L \}$

Def The family of deterministic CF languages is defined as $L_{DPDA}$.




There is a connection between ambiguity & DCF. $L_{DCF} \subseteq L_{NOT\ INHERENTLY\ AMBIG}$

It's only a subset b/c $ww^R$ has an unambiguous grammar $S \to 0S0 \mid 1S1 \mid \varepsilon$