

# Apiladores versus gramáticas

Clase 26

IIC2223 / IIC2224

Prof. Cristian Riveros

# Outline

Apiladores (clase anterior)

Desde CFG a PDA

Desde PDA a CFG

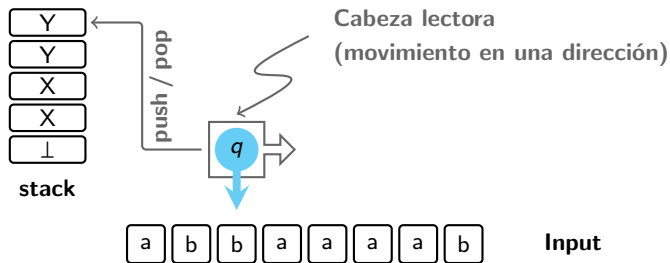
# Outline

Apiladores (clase anterior)

Desde CFG a PDA

Desde PDA a CFG

# Recordatorio: Autómatas apiladores



# Recordatorio: Autómatas apiladores

## Definición

Un **autómata apilador** (PushDown Automata, PDA) es una estructura:

$$\mathcal{P} = (Q, \Sigma, \Gamma, \Delta, q_0, \perp, F)$$

- $Q$  es un conjunto finito de **estados**.
- $\Sigma$  es el alfabeto de **input**.
- $q_0 \in Q$  es el estado **inicial**.
- $F$  es el conjunto de estados **finales**.

+

- $\Gamma$  es el alfabeto de **stack**.
- $\perp \in \Gamma$  es el símbolo **inicial de stack**.
- $\Delta \subseteq (Q \times (\Sigma \cup \{\epsilon\}) \times \Gamma) \times (Q \times \Gamma^*)$  es una relación finita de transición.

# Recordatorio: Autómatas apiladores alternativos

## Definición

Un **PDA alternativo** es una estructura:

$$\mathcal{D} = (Q, \Sigma, \Delta, q_0, F)$$

- $Q$  es un conjunto finito de **estados**.

- $\Sigma$  es el alfabeto de **input**.

- $q_0 \in Q$  es el estado **inicial**.

- $F$  es el conjunto de estados **finales**.

+

- $\Delta \subseteq Q^+ \times (\Sigma \cup \{\epsilon\}) \times Q^*$  es una **relación finita de transición**.

# Equivalencia entre modelos

## Teorema

Para todo autómata apilador  $\mathcal{P}$   
existe un autómata apilador alternativo  $\mathcal{D}$ , y viceversa, tal que:

$$\mathcal{L}(\mathcal{P}) = \mathcal{L}(\mathcal{D})$$

Desde ahora, usaremos ambos modelos de manera **equivalente**.

# Equivalencia entre modelos

## Demostración: de $\mathcal{P}$ a $\mathcal{D}$

Sea  $\mathcal{P} = (Q, \Sigma, \Gamma, \Delta, q_0, \perp, F)$  un PDA.

Construimos un **PDA alternativo**  $\mathcal{D}_{\mathcal{P}} = (Q', \Sigma, \Delta', q'_0, F')$  tal que:

- $Q' = Q \cup \Gamma \cup \{q'_0\}$
- $F' = F$
- $\Delta' = \{ (q'_0, \epsilon, q_0 \perp) \} \cup \{ (pA, a, q\gamma) \mid (p, a, A, q, \gamma) \in \Delta \}$

Ejercicio: demuestre que  $\mathcal{L}(\mathcal{P}) = \mathcal{L}(\mathcal{D}_{\mathcal{P}})$ .



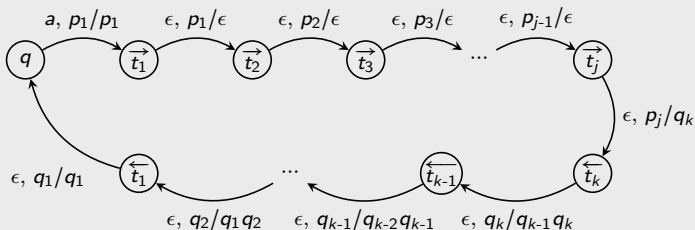
# Equivalencia entre modelos

## Demostración: de $\mathcal{D}$ a $\mathcal{P}$

Sea  $\mathcal{D} = (Q, \Sigma, \Delta, q_0, F)$  un PDA alternativo.

Construimos un **PDA**  $\mathcal{P}_{\mathcal{D}} = (Q', \Sigma, \Gamma', \Delta', q'_0, \perp', F')$  tal que ...

... para cada  $t = (p_1 \dots p_j, a, q_1 \dots q_k) \in \Delta$ , haremos lo siguiente:



# Equivalencia entre modelos

## Demostración: de $\mathcal{D}$ a $\mathcal{P}$

Sea  $\mathcal{D} = (Q, \Sigma, \Delta, q_0, F)$  un PDA alternativo.

Construimos un **PDA**  $\mathcal{P}_{\mathcal{D}} = (Q', \Sigma, \Gamma', \Delta', q'_0, \perp', F')$  tal que:

- $Q' = \{q, q_f\} \cup \bigcup_{t: (\alpha, a, \beta) \in \Delta} \{\vec{t}_i \mid 1 \leq i \leq |\alpha|\} \cup \{\overleftarrow{t}_i \mid 1 \leq i \leq |\beta|\}$
- $\Gamma' = Q$
- $\perp' = q_0$
- $q'_0 = q$
- $F = \{q_f\}$

# Equivalencia entre modelos

## Demostración: de $\mathcal{D}$ a $\mathcal{P}$

Sea  $\mathcal{D} = (Q, \Sigma, \Delta, q_0, F)$  un PDA alternativo.

Construimos un **PDA**  $\mathcal{P}_{\mathcal{D}} = (Q', \Sigma, \Gamma', \Delta', q'_0, \perp', F')$  tal que:

- $\Delta'$ : para cada  $t = (p_1 \dots p_j, a, q_1 \dots q_k) \in \Delta$  tenemos:

$$(q, a, p_1, \overrightarrow{t_1}, p_1) \in \Delta'$$

$$(\overrightarrow{t_i}, \epsilon, p_i, \overrightarrow{t_{i+1}}, \epsilon) \in \Delta' \quad \text{para todo } 1 \leq i < j$$

$$\text{si } k = 0: \quad (\overrightarrow{t_j}, \epsilon, p_j, q, \epsilon) \in \Delta'$$

$$\text{si } k > 0: \quad (\overrightarrow{t_j}, \epsilon, p_j, \overleftarrow{t_k}, q_k) \in \Delta'$$

$$(\overleftarrow{t_i}, \epsilon, q_i, \overleftarrow{t_{i-1}}, q_{i-1}q_i) \in \Delta' \quad \text{para todo } 1 < i \leq k$$

$$(\overleftarrow{t_1}, \epsilon, q_1, q, q_1) \in \Delta'$$

$$\text{para todo } p \in F: \quad (q, \epsilon, p, q_f, \epsilon) \in \Delta'$$

Ejercicio: demuestre que  $\mathcal{L}(\mathcal{D}) = \mathcal{L}(\mathcal{P}_{\mathcal{D}})$ .

# ¿en qué se parecen CFG a PDA?

## Teorema

Todo **lenguaje libre de contexto** puede ser descrito equivalentemente por:

- Una gramática libre de contexto (**CFG**).
- Un autómata apilador (**PDA**).

# Outline

Apiladores (clase anterior)

**Desde CFG a PDA**

Desde PDA a CFG

# Desde CFG a un PDA

## Teorema

Para toda gramática libre de contexto  $\mathcal{G}$ ,  
existe un **autómata apilador alternativo**  $\mathcal{D}$ :

$$\mathcal{L}(\mathcal{G}) = \mathcal{L}(\mathcal{D})$$

¿cómo **construimos**  $\mathcal{D}$  desde  $\mathcal{G}$ ?

## CFG $\rightarrow$ PDA: Construcción

Sea  $\mathcal{G} = (V, \Sigma, P, S)$  una CFG.

Construimos un PDA alternativo  $\mathcal{D}$  que acepta  $\mathcal{L}(\mathcal{G})$ :

$$\mathcal{D} = (V \cup \Sigma \cup \{q_0, q_f\}, \Sigma, \Delta, q_0, \{q_f\})$$

La relación de transición  $\Delta$  se define como:

$$\begin{aligned} \Delta = & \{ (q_0, \epsilon, S \cdot q_f) \} && \cup \\ & \{ (X, \epsilon, \gamma) \mid X \rightarrow \gamma \in P \} && \cup \text{ (Expandir)} \\ & \{ (a, a, \epsilon) \mid a \in \Sigma \} && \text{ (Reducir)} \end{aligned}$$

Ejemplo en clases

$$\mathcal{G}: \quad S \quad \rightarrow \quad SS \mid aSb \mid \epsilon$$

Intuitivamente, ¿qué está haciendo el autómata apilador  $\mathcal{D}$ ?

# CFG $\rightarrow$ PDA: Demostración

Por demostrar:

$$\mathcal{L}(\mathcal{G}) = \mathcal{L}(\mathcal{D})$$

Dos direcciones:

1.  $\mathcal{L}(\mathcal{G}) \subseteq \mathcal{L}(\mathcal{D})$
2.  $\mathcal{L}(\mathcal{D}) \subseteq \mathcal{L}(\mathcal{G})$



## CFG $\rightarrow$ PDA: Demostración $\mathcal{L}(\mathcal{G}) \subseteq \mathcal{L}(\mathcal{D})$

Para cada  $w \in \mathcal{L}(\mathcal{G})$  debemos encontrar una ejecución de aceptación de  $\mathcal{D}$  sobre  $w$ .

¿cómo encontramos una ejecución sobre  $w$ ?

### Idea

Para cada árbol de derivación  $\mathcal{T}$  de  $\mathcal{G}$  sobre  $w$ , construimos una ejecución de  $\mathcal{D}$  sobre  $w$  que recorre el árbol  $\mathcal{T}$  **en profundidad**.

**Inducción** sobre la altura del árbol  $\mathcal{T}$

# CFG $\rightarrow$ PDA: Demostración $\mathcal{L}(\mathcal{G}) \subseteq \mathcal{L}(\mathcal{D})$

## Hipótesis de inducción

Para todo árbol de derivación  $\mathcal{T}$  de  $\mathcal{G}$  con **altura**  $h$  tal que:

- la raíz de  $\mathcal{T}$  es  $X$ , y
- $\mathcal{T}$  produce la palabra  $w$

entonces  $(X \cdot \gamma, w) \vdash_{\mathcal{D}}^* (\gamma, \epsilon)$  para todo  $\gamma \in Q^+$ .

Si demostramos esta hipótesis habremos demostrado que  $\mathcal{L}(\mathcal{G}) \subseteq \mathcal{L}(\mathcal{D})$

¿por qué?

## CFG $\rightarrow$ PDA: Demostración $\mathcal{L}(\mathcal{G}) \subseteq \mathcal{L}(\mathcal{D})$

Caso base:  $h = 1$

Si  $\mathcal{T}$  tiene altura 1 entonces:

- $\mathcal{T}$  produce la palabra  $w = a$  para algún  $a \in \Sigma$  y
- $\mathcal{T}$  consiste de un nodo  $X$  y un hijo  $a$  con  $X \rightarrow a$ .

Entonces para todo  $\gamma \in Q^+$ :

$$(X \cdot \gamma, a) \vdash_{\mathcal{D}} (a \cdot \gamma, a) \vdash_{\mathcal{D}} (\gamma, \epsilon)$$

es una ejecución de  $\mathcal{D}$  sobre  $a$ .



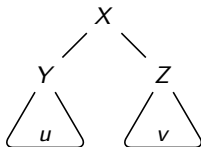
# CFG $\rightarrow$ PDA: Demostración $\mathcal{L}(\mathcal{G}) \subseteq \mathcal{L}(\mathcal{D})$

Caso inductivo:  $h = n$

Suponemos que el árbol de derivación  $\mathcal{T}$  de  $\mathcal{G}$  tiene **altura**  $n$  tal que:

- la raíz de  $\mathcal{T}$  es  $X$ , y
- $\mathcal{T}$  produce la palabra  $w$ .

**Sin pérdida de generalidad** suponga que  $\mathcal{T}$  es de la forma:

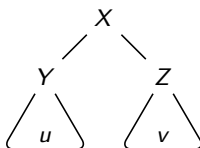


donde  $w = u \cdot v$  y  $X \rightarrow YZ$ .

# CFG $\rightarrow$ PDA: Demostración $\mathcal{L}(\mathcal{G}) \subseteq \mathcal{L}(\mathcal{D})$

Caso inductivo:  $h = n$

**Sin perdida de generalidad** suponga que  $\mathcal{T}$  es de la forma:



donde  $w = u \cdot v$  y  $X \rightarrow YZ$ .

Por HI, se tiene que para todo  $\gamma_1, \gamma_2 \in Q^+$ :

$$(Y \cdot \gamma_1, u) \vdash_{\mathcal{D}}^* (\gamma_1, \epsilon)$$

$$(Z \cdot \gamma_2, v) \vdash_{\mathcal{D}}^* (\gamma_2, \epsilon)$$

## CFG $\rightarrow$ PDA: Demostración $\mathcal{L}(\mathcal{G}) \subseteq \mathcal{L}(\mathcal{D})$

Caso inductivo:  $h = n$

Por HI, se tiene que para todo  $\gamma_1, \gamma_2 \in Q^+$ :

$$(Y \cdot \gamma_1, u) \vdash_{\mathcal{D}}^* (\gamma_1, \epsilon)$$

$$(Z \cdot \gamma_2, v) \vdash_{\mathcal{D}}^* (\gamma_2, \epsilon)$$

Para  $\gamma \in Q^+$  **construimos** la siguiente ejecución de  $\mathcal{D}$  sobre  $w = uv$ :

$$(X \cdot \gamma, uv) \vdash_{\mathcal{D}} (YZ \cdot \gamma, uv) \vdash_{\mathcal{D}}^* (Z \cdot \gamma, v) \vdash_{\mathcal{D}}^* (\gamma, \epsilon)$$



## CFG $\rightarrow$ PDA: Demostración $\mathcal{L}(\mathcal{D}) \subseteq \mathcal{L}(\mathcal{G})$

Para cada  $w \in \mathcal{L}(\mathcal{D})$  debemos encontrar un árbol de derivación  $\mathcal{G}$  para  $w$ .

¿cómo encontramos un árbol de derivación para  $w$ ?

Idea

Si tenemos una ejecución de  $\mathcal{D}$  sobre  $w$  de la forma:

$$(X \cdot q_f, w) \vdash_{\mathcal{D}}^* (q_f, \epsilon)$$

entonces  $X \xRightarrow[\mathcal{G}]^* w$

**Inducción** en la cantidad de pasos de la ejecución.

## CFG $\rightarrow$ PDA: Demostración $\mathcal{L}(\mathcal{D}) \subseteq \mathcal{L}(\mathcal{G})$

### Hipótesis de inducción

Para toda ejecución de  $\mathcal{D}$  sobre  $w$  de largo  $k$  de la forma:

$$(X \cdot q_f, w) = (\gamma_0, w_0) \vdash_{\mathcal{D}} (\gamma_1, w_1) \vdash_{\mathcal{D}} \cdots \vdash_{\mathcal{D}} (\gamma_k, w_k) = (q_f, \epsilon)$$

entonces  $X \xRightarrow[\mathcal{G}]{}^* w$ .

**Ejercicio:** terminar la demostración.



# Outline

Apiladores (clase anterior)

Desde CFG a PDA

Desde PDA a CFG

# Desde PDA a un CFG

## Teorema

Para todo autómata apilador  $\mathcal{P}$ , existe una gramática libre de contexto  $\mathcal{G}$ :

$$\mathcal{L}(\mathcal{P}) = \mathcal{L}(\mathcal{G})$$

## Estrategia de la demostración

Sea  $\mathcal{P} = (Q, \Sigma, \Gamma, \Delta, q_0, \perp, F)$  un PDA (normal).

1. Convertir  $\mathcal{P}$  a un PDA  $\mathcal{P}'$  con **UN solo estado**.
2. Convertir  $\mathcal{P}'$  a una gramática libre de contexto  $\mathcal{G}$ .

¿cómo hacemos cada paso?

# PDA $\rightarrow$ CFG: Demostración

Paso 2: Convertir  $\mathcal{P}'$  a una CFG  $\mathcal{G}$

Sea  $\mathcal{P}' = (\{q\}, \Sigma, \Gamma, \Delta, q, \perp, \{q\})$  con **UN solo estado**.

Construimos la gramática:

$$\mathcal{G} = (V, \Sigma, P, \perp)$$

- $V = \Gamma$ .
- Si  $qA \xrightarrow{\epsilon} q\alpha \in \Delta$  entonces:  $A \rightarrow \alpha \in P$
- Si  $qA \xrightarrow{a} q\alpha \in \Delta$  entonces:  $A \rightarrow a\alpha \in P$

Demostración: ejercicio.

# PDA $\rightarrow$ CFG: Demostración

Paso 1: Convertir  $\mathcal{P}$  a un PDA  $\mathcal{P}'$  con UN solo estado

Sea  $\mathcal{P} = (Q, \Sigma, \Gamma, \Delta, q_0, \perp, F)$  un PDA.

¿cómo guardamos la información de los estados en el stack?

Pregunta principal

*“Si el PDA esta en el estado  $p$  y en el tope del stack hay una  $A$  entonces, ¿a cuál estado llegaré al remover  $A$  del stack?”*

Solución

Podemos **adivinar** (no-determinismo) el estado que vamos a llegar cuando removamos  $A$  del stack.

# PDA $\rightarrow$ CFG: Demostración

Paso 1: Convertir  $\mathcal{P}$  a un PDA  $\mathcal{P}'$  con UN solo estado

Sea  $\mathcal{P} = (Q, \Sigma, \Gamma, \Delta, q_0, \perp, F)$  un PDA.

**Sin perdida de generalidad** podemos asumir que:

1. Todas las transiciones son de la forma:

$$qA \xrightarrow{c} pB_1B_2 \quad \text{o} \quad qA \xrightarrow{c} p\epsilon$$

con  $c \in (\Sigma \cup \{\epsilon\})$ .

¿por qué?

2. Existe  $q_f \in Q$  tal que si  $w \in \mathcal{L}(\mathcal{P})$  entonces:

$$(q_0\perp, w) \vdash_{\mathcal{D}}^* (q_f, \epsilon)$$

¿por qué?

Siempre llegamos al **mismo estado**  $q_f$ .

# PDA $\rightarrow$ CFG: Demostración

Paso 1: Convertir  $\mathcal{P}$  a un PDA  $\mathcal{P}'$  con UN solo estado

Sea  $\mathcal{P} = (Q, \Sigma, \Gamma, \Delta, q_0, \perp, F)$  un PDA.

Construimos el autómata apilador  $\mathcal{P}'$  con **un solo estado**:

$$\mathcal{P}' = (\{q\}, \Sigma, \Gamma', \Delta', \{q\}, \perp', \{q\})$$

■  $\Gamma' = Q \times \Gamma \times Q.$

*" $(p, A, q) \in \Gamma'$  si desde  $p$  leyendo  $A$  en el tope de stack llegamos a  $q$  al hacer pop de  $A$ "*

■  $\perp' = (q_0, \perp, q_f)$

*"El autómata parte en  $q_0$  y al hacer pop de  $\perp$  llegará a  $q_f$ "*

# PDA $\rightarrow$ CFG: Demostración

Paso 1: Convertir  $\mathcal{P}$  a un PDA  $\mathcal{P}'$  con UN solo estado

Sea  $\mathcal{P} = (Q, \Sigma, \Gamma, \Delta, q_0, \perp, F)$  un PDA.

Construimos el autómata apilador  $\mathcal{P}'$  con **un solo estado**:

$$\mathcal{P}' = (\{q\}, \Sigma, \Gamma', \Delta', \{q\}, \perp', \{q\})$$

- Si  $pA \xrightarrow{c} p'B_1B_2 \in \Delta$  con  $c \in (\Sigma \cup \{\epsilon\})$ , entonces **para todo**  $p_1, p_2 \in Q$ :

$$q(p, A, p_2) \xrightarrow{c} q(p', B_1, p_1)(p_1, B_2, p_2) \in \Delta'$$

- Si  $pA \xrightarrow{c} p' \in \Delta$  con  $c \in (\Sigma \cup \{\epsilon\})$ , entonces:

$$q(p, A, p') \xrightarrow{c} q \in \Delta'$$

## PDA $\rightarrow$ CFG: Demostración $\mathcal{L}(\mathcal{P}) = \mathcal{L}(\mathcal{P}')$

Hipótesis de inducción (en el número de pasos  $n$ )

Para todo  $p, p' \in Q$ ,  $A \in \Gamma$ , y  $w \in \Sigma^*$  se cumple que:

$$(pA, w) \vdash_{\mathcal{P}}^n (p', \epsilon) \quad \text{si, y solo si,} \quad (q(p, A, p'), w) \vdash_{\mathcal{P}'}^n (q, \epsilon)$$

donde  $\vdash_{\mathcal{P}}^n$  es la relación de **siguiente-paso** de  $\mathcal{P}$   $n$ -veces.

Si demostramos esta hipótesis habremos demostrado que  $\mathcal{L}(\mathcal{P}) = \mathcal{L}(\mathcal{P}')$

¿por qué?



# PDA $\rightarrow$ CFG: Demostración $\mathcal{L}(\mathcal{P}) = \mathcal{L}(\mathcal{P}')$

Caso base:  $n = 1$

Para todo  $p, p' \in Q$  y  $A \in \Gamma$  se cumple que:

$$(pA, c) \vdash_{\mathcal{P}} (p', \epsilon) \quad \text{si, y solo si,} \quad (q(p, A, p'), c) \vdash_{\mathcal{P}'} (q, \epsilon)$$

para todo  $c \in (\Sigma \cup \{\epsilon\})$ .

¿por qué?

# PDA $\rightarrow$ CFG: Demostración $\mathcal{L}(\mathcal{P}) = \mathcal{L}(\mathcal{P}')$

Caso inductivo:

**Sin perdida de generalidad,**

suponga que  $pA \xrightarrow{a} p_1A_1A_2$  y  $w = auv$ , entonces:

$$(pA, \underbrace{auv}_w) \vdash_{\mathcal{P}}^n (p', \epsilon) \quad \text{ssi} \quad (pA, auv) \vdash_{\mathcal{P}} (p_1A_1A_2, uv) \vdash_{\mathcal{P}}^i (p_2A_2, v) \vdash_{\mathcal{P}}^j (p', \epsilon)$$

$$\text{ssi} \quad (p_1A_1, u) \vdash_{\mathcal{P}}^i (p_2, \epsilon) \quad \text{y} \quad (p_2A_2, v) \vdash_{\mathcal{P}}^j (p', \epsilon)$$

$$\text{ssi} \quad (q(p_1, A_1, p_2), u) \vdash_{\mathcal{P}'}^i (q, \epsilon) \quad \text{y} \quad (q(p_2, A_2, p'), v) \vdash_{\mathcal{P}'}^j (q, \epsilon)$$

$$\text{ssi} \quad (q(p, A, p'), auv) \vdash_{\mathcal{P}} (q(p_1, A_1, p_2)(p_2, A_2, q)), uv) \vdash_{\mathcal{P}}^{i+j} (q, \epsilon)$$



# Cierre de clase

En esta clase vimos:

1. Equivalencia entre gramáticas y apiladores.
2. Apiladores pueden recorrer los árboles de derivación en profundidad.
3. Gramáticas pueden simular apiladores al remover el uso de estados.

**Próxima clase:** First y follow.