

Autómatas de pila



UNIVERSIDAD
DE CHILE

Arturo Merino



Recuerdo / Plan

Recuerdo/Plan

- **Recuerdo:**

- Gramáticas libres de contexto
- Lenguajes libres de contexto
- Relación con los lenguajes regulares
- Propiedades de cerradura

- **Plan:**

- Autómatas de pila
- ¿Cómo computan?
- Relación con los lenguajes regulares

Autómata de pila

Autómatas de pila

- AF $\leftarrow$ un computador sin memoria.
- **Extendamos** dichos modelos.
 - Acceso a alguna estructura de datos $\rightarrow$ pila
 - Demos a un AFND acceso a una pila
- Pila: estructura LIFO con las operaciones:
 - push(x)
 - pop()
 - peek()? replace(a,b)? is_empty()?
- ¿Reconocer $0^n 1^n$?
- Modelo más limpio: un string como pila, con acceso por la derecha.
 - push(x) : $\varepsilon \rightarrow x$.
 - pop() == x : $x \rightarrow \varepsilon$
 - pop() == x, push(y) : $x \rightarrow y$.

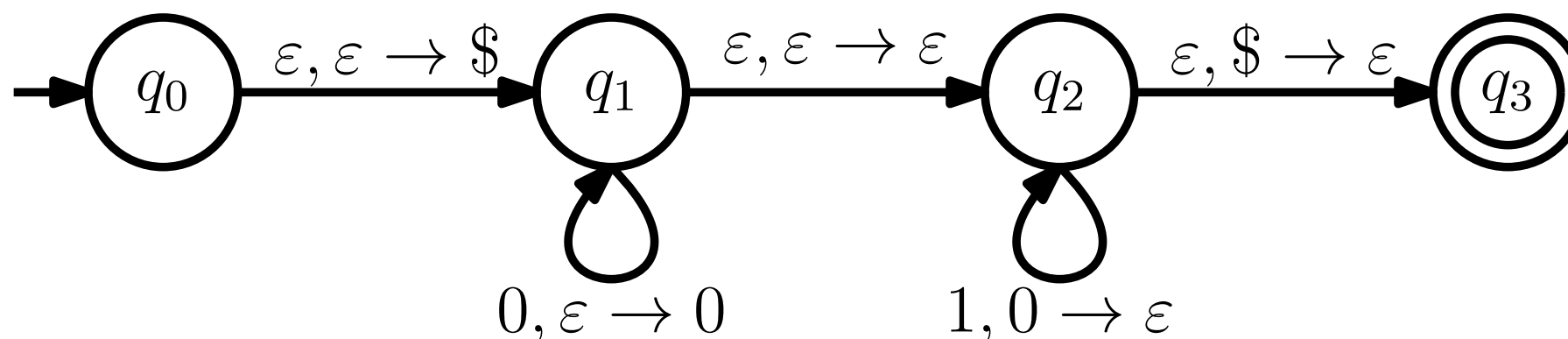
Autómata de pila

Def.

Un *autómata de pila (AP)* es una 6-tupla $(Q, \Sigma, \Gamma, \delta, q_0, F)$, donde

- Q es un conjunto finito de *estados*,
- Σ es un conjunto finito llamado *alfabeto (de entrada)*,
- Γ es un conjunto finito llamado el *alfabeto de pila*,
- $\delta: Q \times (\Sigma \cup \{\varepsilon\}) \times (\Gamma \cup \{\varepsilon\}) \rightarrow 2^{Q \times (\Gamma \cup \{\varepsilon\})}$ es la *f. de transición*,
- $q_0 \in Q$ es el *estado inicial*, y
- $F \subseteq Q$ son los *estados de aceptación (o finales)*.

• Ej:



Computación

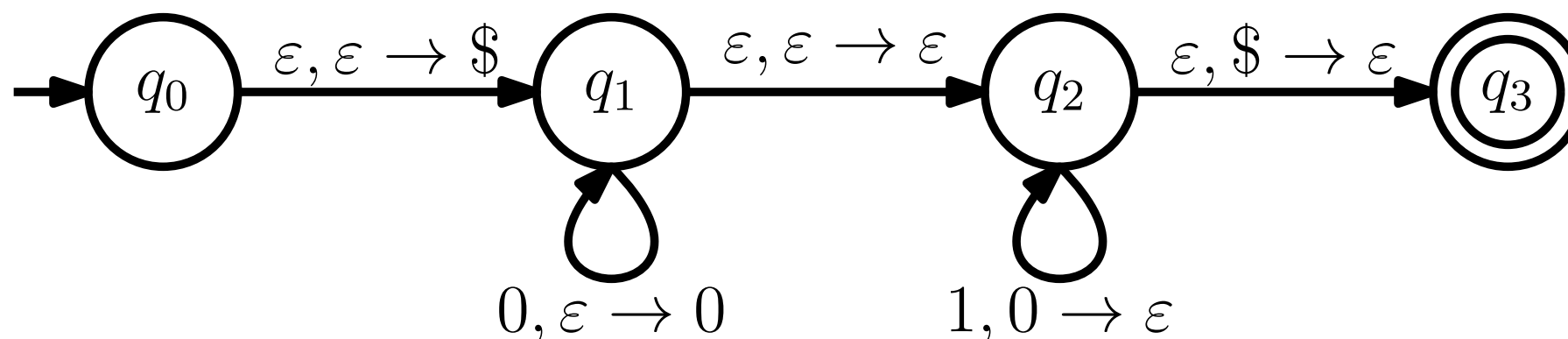
Def.

Sean $M = (Q, \Sigma, \delta, q_0, F)$ un AP y $w \in \Sigma^*$ un string.

Diremos que M *acepta* w si podemos escribir $w = x_1x_2 \dots x_m$ donde cada $x_i \in \Sigma \cup \{\varepsilon\}$, existen estados $r_0, \dots, r_m \in Q$ y existen strings $s_0, \dots, s_m \in \Gamma^*$ tales que

- $r_0 = q_0$ y $s_0 = \varepsilon$ (partimos en la config. inicial)
- para todo $i \in \{1, \dots, m\}$, tenemos que $(r_i, b) \in \delta(r_{i-1}, x_i, a)$
y $s_{i-1} = ta$ y $s_i = tb$ para algún $t \in \Gamma^*$ (transiciones válidas)
- $r_m \in F$ (terminamos en aceptación)

- **Obs:** distintas nociones: pila vacía/estado final y pila vacía.
- **Ej:** ¿01?



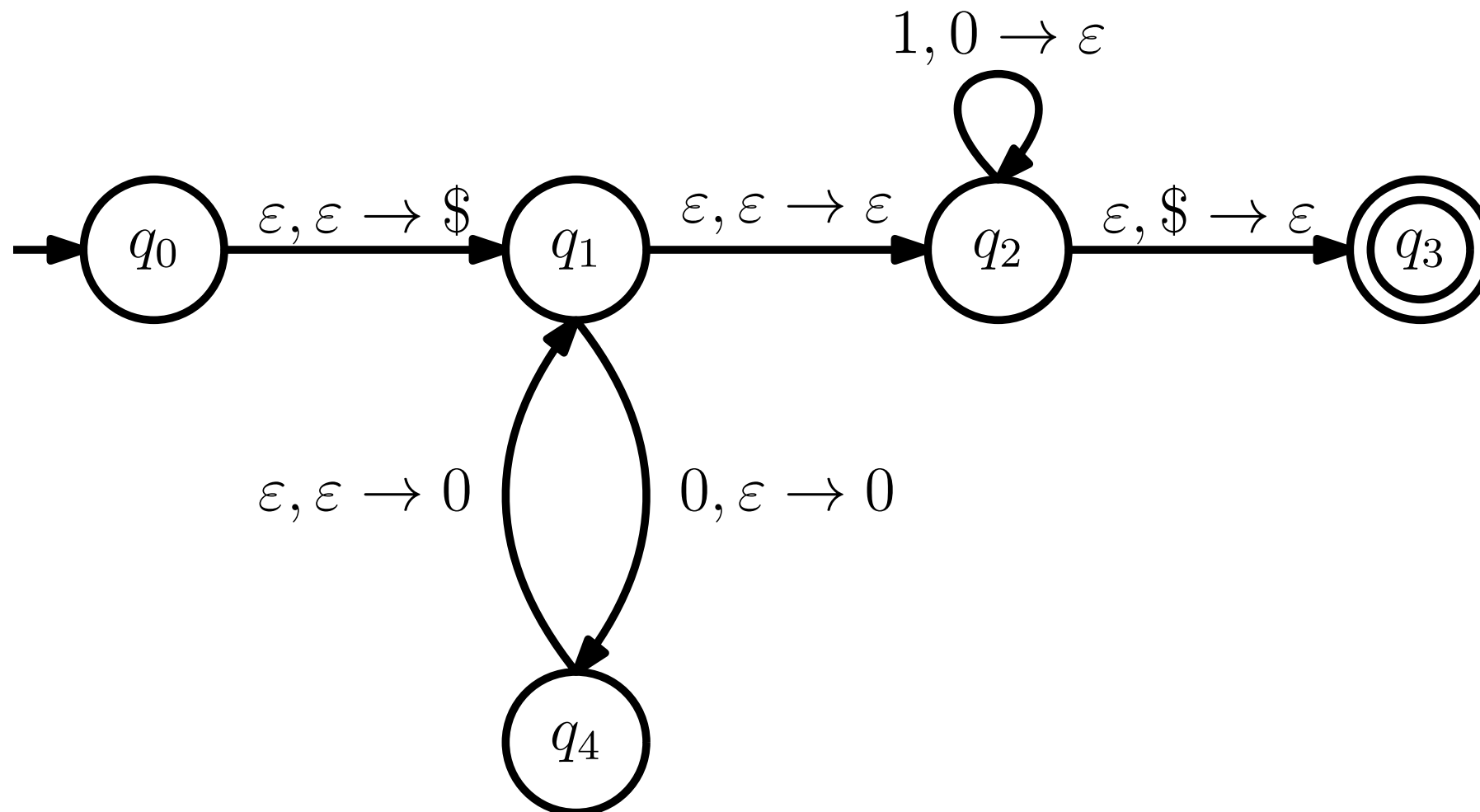
Autómatas vía sus lenguajes

Def.

Sea $M = (Q, \Sigma, \delta, q_0, F)$ un AP. El *lenguaje aceptado* por M es

$$\mathcal{L}(M) = \{w \in \Sigma^* : M(w) = 1\}$$

- **Ej:** Lenguaje aceptado por



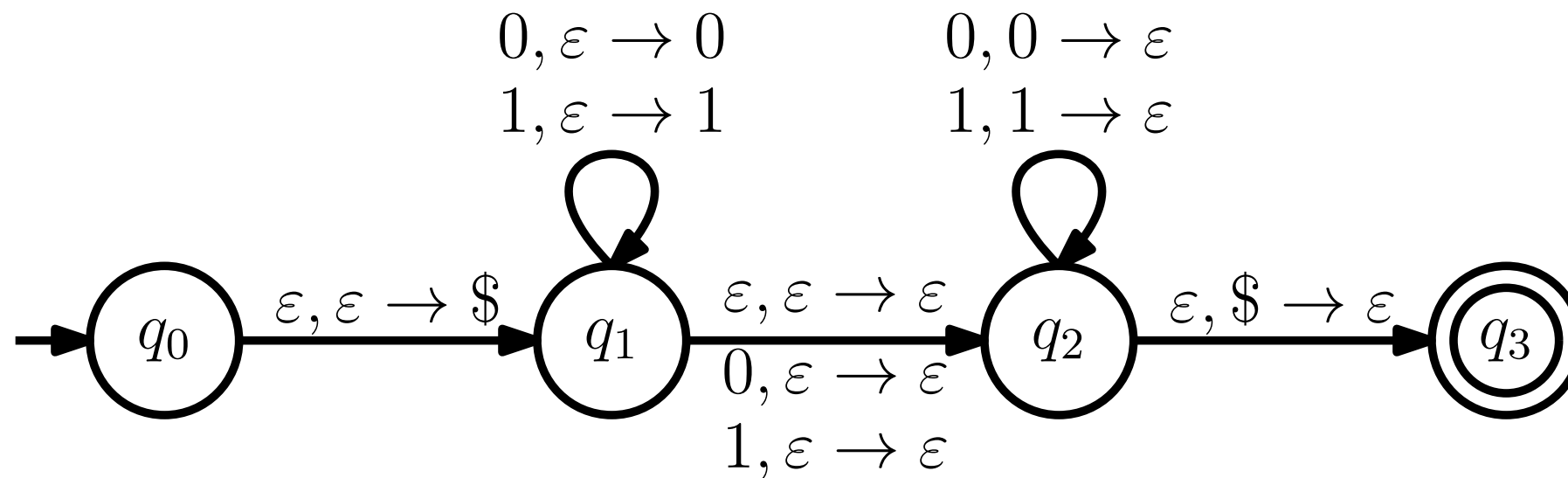
Autómatas vía sus lenguajes

Def.

Sea $M = (Q, \Sigma, \delta, q_0, F)$ un AP. El *lenguaje aceptado* por M es

$$\mathcal{L}(M) = \{w \in \Sigma^* : M(w) = 1\}$$

- **Ej:** Lenguaje aceptado por



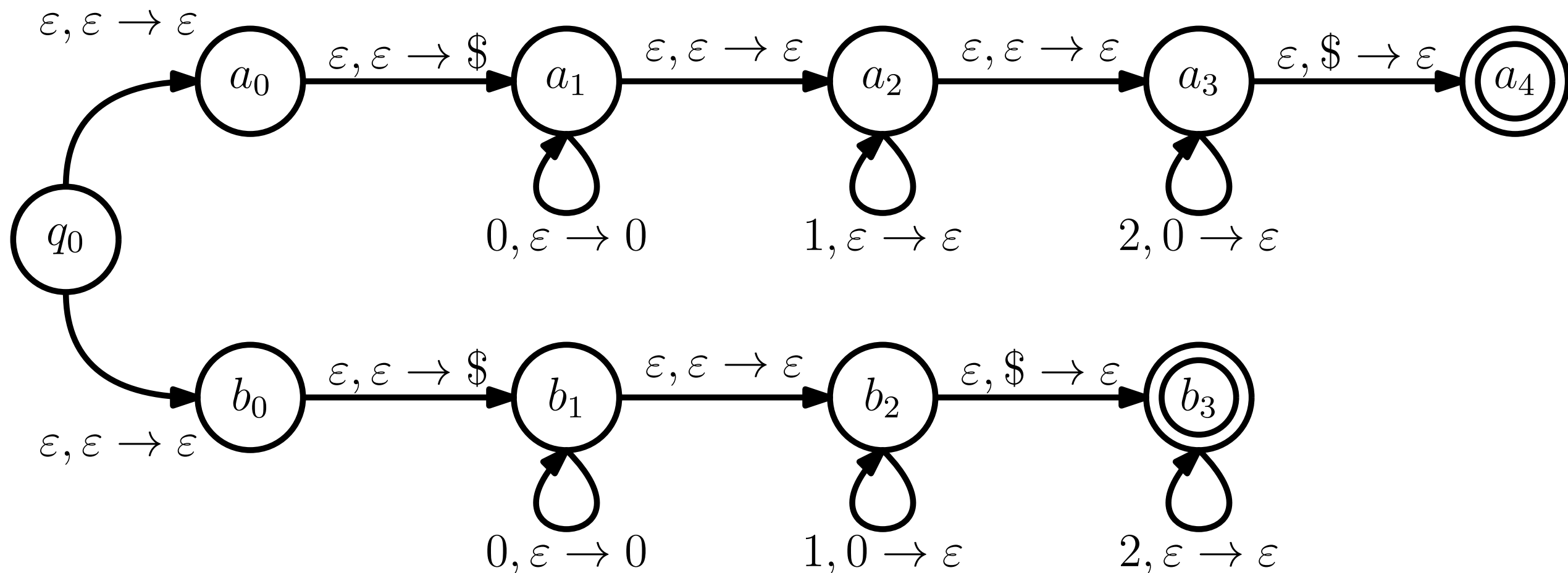
Autómatas vía sus lenguajes

Def.

Sea $M = (Q, \Sigma, \delta, q_0, F)$ un AP. El *lenguaje aceptado* por M es

$$\mathcal{L}(M) = \{w \in \Sigma^* : M(w) = 1\}$$

- **Ej:** Lenguaje aceptado por



Diseño de AP

- ¿Cómo diseño autómatas de pila?
- Tips sobre el no determinismo: “adivinar” partes clave de la computación.
- Usar la pila para recordar información
- Diseñar autómatas para:
 - $L_1 = \{a^i b^j \mid i \geq j\}$
 - $L_2 = \{a^i b^j c^j d^i \mid i, j \geq 0\}$
 - $L_3 = \{a^i \# a^j \# a^{i+j} \mid i, j \geq 0\}$
 - $L_4 = \{w \in \{ (,) \}^* : w \text{ está bien balanceado}\}$
 - $L_5 = \{w \in \{0, 1\}^* : w \text{ tiene la misma cantidad de ceros que de unos}\}$

Lenguajes libres de contexto

- ¿Relación con las GLC?

Teo.

Sea $L \subseteq \Sigma^*$.

L es libre de contexto si y sólo si existe un AP M tal que $\mathcal{L}(M) = L$.

- **Dem:** próxima clase.
- AP y GLC son *equivalentes*.
- ¿AP **deterministas**?
 - Más débiles
 - GLC determinista / GLC con *endmarks*.
 - ¿Realista?
 - Algoritmo de Valiant.

Dudas/Preguntas

Wrap-Up

- **Hoy:**

- Autómatas de pila
- ¿Cómo computan?
- Relación con los lenguajes regulares

- **Próx. clase:**

- Equivalencia AP-GLC