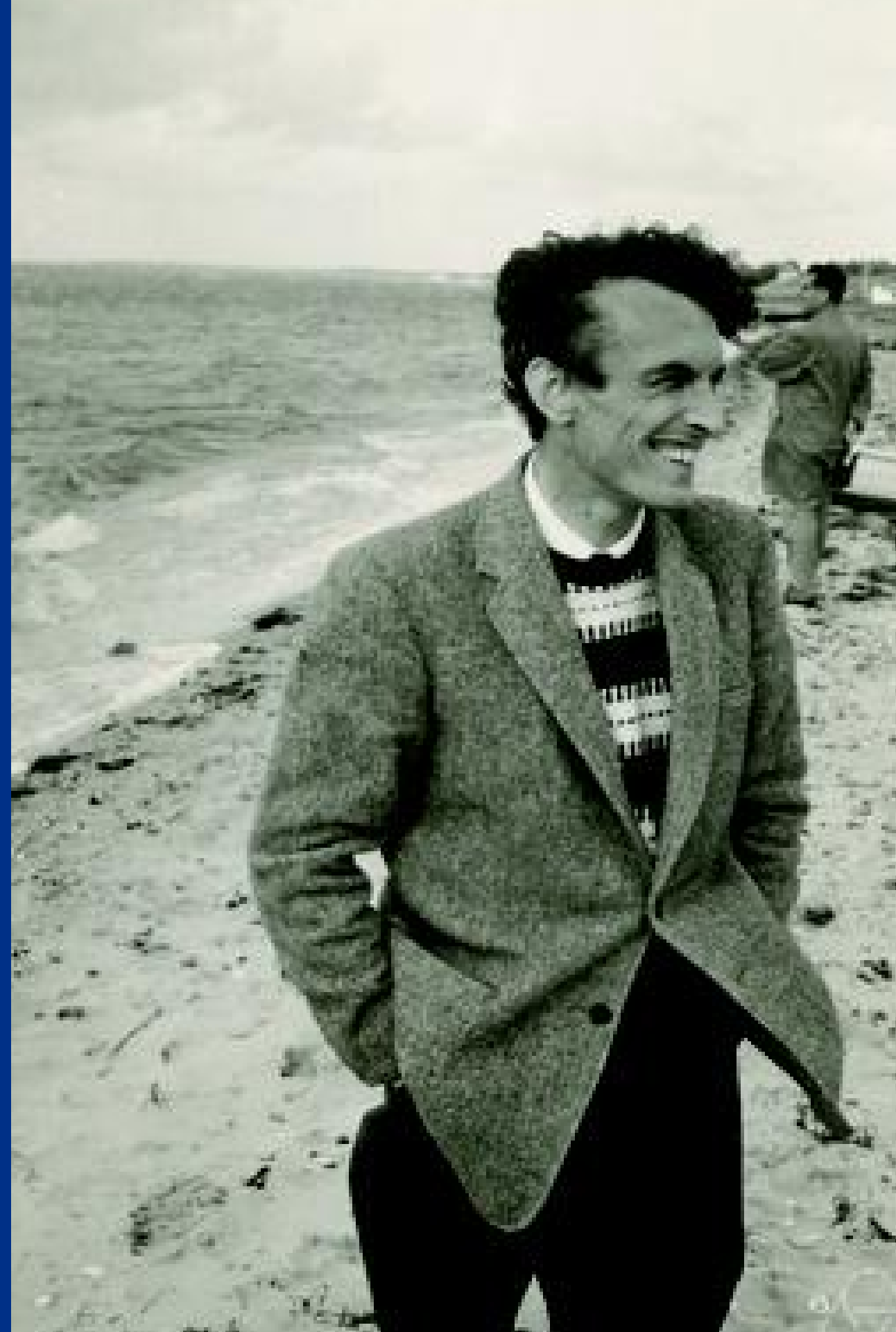


Equivalencia entre AP y GLC



UNIVERSIDAD
DE CHILE

Arturo Merino



Recuerdo / Plan

■ Recuerdo/Plan

- **Recuerdo:**

Recuerdo/Plan

- **Recuerdo:**
 - Autómatas de pila

Recuerdo/Plan

- **Recuerdo:**
 - Autómatas de pila
 - ¿Cómo computan?

Recuerdo/Plan

- **Recuerdo:**
 - Autómatas de pila
 - ¿Cómo computan?
 - ¿Cómo diseñarlos?

Recuerdo/Plan

- **Recuerdo:**

- Autómatas de pila
- ¿Cómo computan?
- ¿Cómo diseñarlos?

- **Plan:**

Recuerdo/Plan

- **Recuerdo:**

- Autómatas de pila
- ¿Cómo computan?
- ¿Cómo diseñarlos?

- **Plan:**

- Equivalencia AP - GLC

Equivalencia

Equivalencia AP - GLC

Teo.

Sea $L \subseteq \Sigma^*$.

Equivalencia AP - GLC

Teo.

Sea $L \subseteq \Sigma^*$.

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

Equivalencia AP - 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:** dos direcciones.

Equivalencia AP - 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:** dos direcciones.
- Dada una GLC, construir un AP equivalente.

Equivalencia AP - 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:** dos direcciones.
- Dada una GLC, construir un AP equivalente.
- Dado un AP, construir una GLC equivalente.

De GLC a AP

De GLC a AP

Lem.

Sea $G = (V, \Sigma, R, S)$ una GLC.

De GLC a AP

Lem.

Sea $G = (V, \Sigma, R, S)$ una GLC.

Existe un AP equivalente a G .

De GLC a AP

Lem.

Sea $G = (V, \Sigma, R, S)$ una GLC.

Existe un AP equivalente a G .

- Una GLC genera; un AP computa

De GLC a AP

Lem.

Sea $G = (V, \Sigma, R, S)$ una GLC.

Existe un AP equivalente a G .

- Una GLC genera; un AP computa
- ¿Cómo hacemos que un AP reconozca el lenguaje generado por una GLC?

De GLC a AP

Lem.

Sea $G = (V, \Sigma, R, S)$ una GLC.

Existe un AP equivalente a G .

- Una GLC genera; un AP computa
- ¿Cómo hacemos que un AP reconozca el lenguaje generado por una GLC?
 - Adivinar la regla de producción

De GLC a AP

Lem.

Sea $G = (V, \Sigma, R, S)$ una GLC.

Existe un AP equivalente a G .

- Una GLC genera; un AP computa
- ¿Cómo hacemos que un AP reconozca el lenguaje generado por una GLC?
 - Adivinar la regla de producción
 - Usar el stack para mantener la derivación

De GLC a AP

Lem.

Sea $G = (V, \Sigma, R, S)$ una GLC.

Existe un AP equivalente a G .

- Una GLC genera; un AP computa
- ¿Cómo hacemos que un AP reconozca el lenguaje generado por una GLC?
 - Adivinar la regla de producción
 - Usar el stack para mantener la derivación
- ¡Problema!

De GLC a AP

Lem.

Sea $G = (V, \Sigma, R, S)$ una GLC.

Existe un AP equivalente a G .

- Una GLC genera; un AP computa
- ¿Cómo hacemos que un AP reconozca el lenguaje generado por una GLC?
 - Adivinar la regla de producción
 - Usar el stack para mantener la derivación
- ¡Problema!

$$\begin{array}{l} E \rightarrow E + T \mid T \\ T \rightarrow T \times F \mid F \\ F \rightarrow (E) \mid a \end{array}$$

$$E \Rightarrow E + T \Rightarrow T + T \Rightarrow T + T \times F \xRightarrow{*} a + a \times a.$$

De GLC a AP

Lem.

Sea $G = (V, \Sigma, R, S)$ una GLC.

Existe un AP equivalente a G .

- Una GLC genera; un AP computa
- ¿Cómo hacemos que un AP reconozca el lenguaje generado por una GLC?
 - Adivinar la regla de producción
 - Usar el stack para mantener la derivación
- ¡Problema!

$$\begin{array}{l} E \rightarrow E + T \mid T \\ T \rightarrow T \times F \mid F \\ F \rightarrow (E) \mid a \end{array}$$

$$E \Rightarrow E + T \Rightarrow T + T \Rightarrow T + T \times F \xRightarrow{*} a + a \times a.$$

- Solución:

De GLC a AP

Lem.

Sea $G = (V, \Sigma, R, S)$ una GLC.

Existe un AP equivalente a G .

- Una GLC genera; un AP computa
- ¿Cómo hacemos que un AP reconozca el lenguaje generado por una GLC?
 - Adivinar la regla de producción
 - Usar el stack para mantener la derivación

- ¡Problema!

$$\begin{array}{l} E \rightarrow E + T \mid T \\ T \rightarrow T \times F \mid F \\ F \rightarrow (E) \mid a \end{array}$$

$$E \Rightarrow E + T \Rightarrow T + T \Rightarrow T + T \times F \xRightarrow{*} a + a \times a.$$

- Solución:
 - Reemplazar sólo por un lado $\rightarrow$ aparecen terminales

De GLC a AP

Lem.

Sea $G = (V, \Sigma, R, S)$ una GLC.

Existe un AP equivalente a G .

- Una GLC genera; un AP computa
- ¿Cómo hacemos que un AP reconozca el lenguaje generado por una GLC?
 - Adivinar la regla de producción
 - Usar el stack para mantener la derivación

- ¡Problema!

$$\begin{array}{l} E \rightarrow E + T \mid T \\ T \rightarrow T \times F \mid F \\ F \rightarrow (E) \mid a \end{array}$$

$$E \Rightarrow E + T \Rightarrow T + T \Rightarrow T + T \times F \xRightarrow{*} a + a \times a.$$

- Solución:
 - Reemplazar sólo por un lado $\rightarrow$ aparecen terminales
 - Consumir terminales

De GLC a AP

Lem.

Sea $G = (V, \Sigma, R, S)$ una GLC.

Existe un AP equivalente a G .

- **Idea:**

- Adivinar la regla de producción
- Usar el stack para mantener la derivación
- Reemplazar sólo por un lado $\rightarrow$ aparecen terminales
- Consumir terminales

De GLC a AP

Lem.

Sea $G = (V, \Sigma, R, S)$ una GLC.

Existe un AP equivalente a G .

- **Idea:**

- Adivinar la regla de producción
- Usar el stack para mantener la derivación
- Reemplazar sólo por un lado $\rightarrow$ aparecen terminales
- Consumir terminales

- **Dem:**

- Para simplificar, suponemos que $\Sigma = \{a, b\}$.

De GLC a AP

Lem.

Sea $G = (V, \Sigma, R, S)$ una GLC.

Existe un AP equivalente a G .

- **Idea:**

- Adivinar la regla de producción
- Usar el stack para mantener la derivación
- Reemplazar sólo por un lado $\rightarrow$ aparecen terminales
- Consumir terminales

- **Dem:**

- Para simplificar, suponemos que $\Sigma = \{a, b\}$.

- **Ej:**

$$\begin{array}{l} S \rightarrow aTb \mid b \\ T \rightarrow Ta \mid \varepsilon \end{array}$$

De GLC a AP

De AP a GLC

Lem.

Sea $A = (Q, \Sigma, \Gamma, \delta, q_0, F)$ un AP.

De AP a GLC

Lem.

Sea $A = (Q, \Sigma, \Gamma, \delta, q_0, F)$ un AP.

Existe una GLC equivalente a A .

De AP a GLC

Lem.

Sea $A = (Q, \Sigma, \Gamma, \delta, q_0, F)$ un AP.

Existe una GLC equivalente a A .

- **Simplificaciones:**

De AP a GLC

Lem.

Sea $A = (Q, \Sigma, \Gamma, \delta, q_0, F)$ un AP.

Existe una GLC equivalente a A .

- **Simplificaciones:**
 - A tiene un único estado final

De AP a GLC

Lem.

Sea $A = (Q, \Sigma, \Gamma, \delta, q_0, F)$ un AP.

Existe una GLC equivalente a A .

- **Simplificaciones:**
 - A tiene un único estado final
 - A tiene la pila vacía cuando acepta

De AP a GLC

Lem.

Sea $A = (Q, \Sigma, \Gamma, \delta, q_0, F)$ un AP.

Existe una GLC equivalente a A .

- **Simplificaciones:**

- A tiene un único estado final
- A tiene la pila vacía cuando acepta
- A sólo hace push y pop (y no replace)

De AP a GLC

Lem.

Sea $A = (Q, \Sigma, \Gamma, \delta, q_0, F)$ un AP.

Existe una GLC equivalente a A .

- **Simplificaciones:**

- A tiene un único estado final
- A tiene la pila vacía cuando acepta
- A sólo hace push y pop (y no replace)

- Queremos que G genere los strings que llevan al autómata de q_0 a q_f con pila vacía.

De AP a GLC

Lem.

Sea $A = (Q, \Sigma, \Gamma, \delta, q_0, F)$ un AP.

Existe una GLC equivalente a A .

- **Simplificaciones:**

- A tiene un único estado final
- A tiene la pila vacía cuando acepta
- A sólo hace push y pop (y no replace)

- Queremos que G genere los strings que llevan al autómata de q_0 a q_f con pila vacía.
- **PD:** $A_{pq} \leftarrow$ strings que llevan al autómata de p a q con pila vacía.

De AP a GLC

Lem.

Sea $A = (Q, \Sigma, \Gamma, \delta, q_0, F)$ un AP.

Existe una GLC equivalente a A .

- **Simplificaciones:**

- A tiene un único estado final
- A tiene la pila vacía cuando acepta
- A sólo hace push y pop (y no replace)

- Queremos que G genere los strings que llevan al autómata de q_0 a q_f con

- **PD:** A_{pq} ← strings que llevan al autómata de p a q con pila vacía.

- **C1:** Pasamos por estado intermedio con pila vacía:

De AP a GLC

Lem.

Sea $A = (Q, \Sigma, \Gamma, \delta, q_0, F)$ un AP.

Existe una GLC equivalente a A .

- **Simplificaciones:**

- A tiene un único estado final
- A tiene la pila vacía cuando acepta
- A sólo hace push y pop (y no replace)

- Queremos que G genere los strings que llevan al autómata de q_0 a q_f con

- **PD:** $A_{pq} \leftarrow$ strings que llevan al autómata de p a q con pila vacía.

- **C1:** Pasamos por estado intermedio con pila vacía:

$$A_{pq} \rightarrow A_{pr} A_{rq}$$

De AP a GLC

Lem.

Sea $A = (Q, \Sigma, \Gamma, \delta, q_0, F)$ un AP.

Existe una GLC equivalente a A .

- **Simplificaciones:**

- A tiene un único estado final
- A tiene la pila vacía cuando acepta
- A sólo hace push y pop (y no replace)

- Queremos que G genere los strings que llevan al autómata de q_0 a q_f con pila vacía.
- **PD:** $A_{pq} \leftarrow$ strings que llevan al autómata de p a q con pila vacía.

- **C1:** Pasamos por estado intermedio con pila vacía:

$$A_{pq} \rightarrow A_{pr}A_{rq}$$

- **C2:** No pasamos por estado intermedio con pila vacía:

De AP a GLC

Lem.

Sea $A = (Q, \Sigma, \Gamma, \delta, q_0, F)$ un AP.

Existe una GLC equivalente a A .

- **Simplificaciones:**

- A tiene un único estado final
- A tiene la pila vacía cuando acepta
- A sólo hace push y pop (y no replace)

- Queremos que G genere los strings que llevan al autómata de q_0 a q_f con

- **PD:** $A_{pq} \leftarrow$ strings que llevan al autómata de p a q con pila vacía.

- **C1:** Pasamos por estado intermedio con pila vacía:

$$A_{pq} \rightarrow A_{pr}A_{rq}$$

- **C2:** No pasamos por estado intermedio con pila vacía:

$$\text{Apilamos y desapilamos el mismo símbolo: } A_{pq} \rightarrow aA_{rs}b$$

De AP a GLC

Lem.

Sea $A = (Q, \Sigma, \Gamma, \delta, q_0, F)$ un AP.

Existe una GLC equivalente a A .

- **Simplificaciones:**

- A tiene un único estado final
- A tiene la pila vacía cuando acepta
- A sólo hace push y pop (y no replace)

- Queremos que G genere los strings que llevan al autómata de q_0 a q_f con

- **PD:** A_{pq} ← strings que llevan al autómata de p a q con pila vacía.

- **C1:** Pasamos por estado intermedio con pila vacía:

$$A_{pq} \rightarrow A_{pr}A_{rq}$$

- **C2:** No pasamos por estado intermedio con pila vacía:

Apilamos y desapilamos el mismo símbolo: $A_{pq} \rightarrow aA_{rs}b$

- Caso base de la PD:

Lem.

Sea $A = (Q, \Sigma, \Gamma, \delta, q_0, F)$ un AP.

Existe una GLC equivalente a A .

- **Simplificaciones:**

- A tiene un único estado final
- A tiene la pila vacía cuando acepta
- A sólo hace push y pop (y no replace)

- Queremos que G genere los strings que llevan al autómata de q_0 a q_f con pila vacía.
- **PD:** $A_{pq} \leftarrow$ strings que llevan al autómata de p a q con pila vacía.

- **C1:** Pasamos por estado intermedio con pila vacía:

$$A_{pq} \rightarrow A_{pr}A_{rq}$$

- **C2:** No pasamos por estado intermedio con pila vacía:

Apilamos y desapilamos el mismo símbolo: $A_{pq} \rightarrow aA_{rs}b$

- Caso base de la PD:

$$A_{pp} \rightarrow \varepsilon$$

De AP a GLC

Lem.

Sea $A = (Q, \Sigma, \Gamma, \delta, q_0, F)$ un AP.

De AP a GLC

Lem.

Sea $A = (Q, \Sigma, \Gamma, \delta, q_0, F)$ un AP.

Existe una GLC equivalente a A .

De AP a GLC

Lem.

Sea $A = (Q, \Sigma, \Gamma, \delta, q_0, F)$ un AP.

Existe una GLC equivalente a A .

- **Simplificaciones:**

De AP a GLC

Lem.

Sea $A = (Q, \Sigma, \Gamma, \delta, q_0, F)$ un AP.

Existe una GLC equivalente a A .

- **Simplificaciones:**
 - A tiene un único estado final

De AP a GLC

Lem.

Sea $A = (Q, \Sigma, \Gamma, \delta, q_0, F)$ un AP.

Existe una GLC equivalente a A .

- **Simplificaciones:**
 - A tiene un único estado final
 - A tiene la pila vacía cuando acepta

De AP a GLC

Lem.

Sea $A = (Q, \Sigma, \Gamma, \delta, q_0, F)$ un AP.

Existe una GLC equivalente a A .

- **Simplificaciones:**

- A tiene un único estado final
- A tiene la pila vacía cuando acepta
- A sólo hace push y pop (y no replace)

De AP a GLC

Lem.

Sea $A = (Q, \Sigma, \Gamma, \delta, q_0, F)$ un AP.

Existe una GLC equivalente a A .

- **Simplificaciones:**

- A tiene un único estado final
- A tiene la pila vacía cuando acepta
- A sólo hace push y pop (y no replace)

- **Idea:**

De AP a GLC

Lem.

Sea $A = (Q, \Sigma, \Gamma, \delta, q_0, F)$ un AP.

Existe una GLC equivalente a A .

- **Simplificaciones:**

- A tiene un único estado final
- A tiene la pila vacía cuando acepta
- A sólo hace push y pop (y no replace)

- **Idea:**

- $A_{pq} \rightarrow A_{pr}A_{rq}$

De AP a GLC

Lem.

Sea $A = (Q, \Sigma, \Gamma, \delta, q_0, F)$ un AP.

Existe una GLC equivalente a A .

- **Simplificaciones:**

- A tiene un único estado final
- A tiene la pila vacía cuando acepta
- A sólo hace push y pop (y no replace)

- **Idea:**

- $A_{pq} \rightarrow A_{pr}A_{rq}$
- Apilamos y desapilamos el mismo símbolo: $A_{pq} \rightarrow aA_{rs}b$

De AP a GLC

Lem.

Sea $A = (Q, \Sigma, \Gamma, \delta, q_0, F)$ un AP.

Existe una GLC equivalente a A .

- **Simplificaciones:**

- A tiene un único estado final
- A tiene la pila vacía cuando acepta
- A sólo hace push y pop (y no replace)

- **Idea:**

- $A_{pq} \rightarrow A_{pr}A_{rq}$
- Apilamos y desapilamos el mismo símbolo: $A_{pq} \rightarrow aA_{rs}b$
- $A_{pp} \rightarrow \varepsilon$

De AP a GLC

Lem.

Sea $A = (Q, \Sigma, \Gamma, \delta, q_0, F)$ un AP.

Existe una GLC equivalente a A .

- **Simplificaciones:**

- A tiene un único estado final
- A tiene la pila vacía cuando acepta
- A sólo hace push y pop (y no replace)

- **Idea:**

- $A_{pq} \rightarrow A_{pr}A_{rq}$
- Apilamos y desapilamos el mismo símbolo: $A_{pq} \rightarrow aA_{rs}b$
- $A_{pp} \rightarrow \varepsilon$

- **Dem:**

De AP a GLC

Lem.

Sea $A = (Q, \Sigma, \Gamma, \delta, q_0, F)$ un AP.

Existe una GLC equivalente a A .

- **Simplificaciones:**

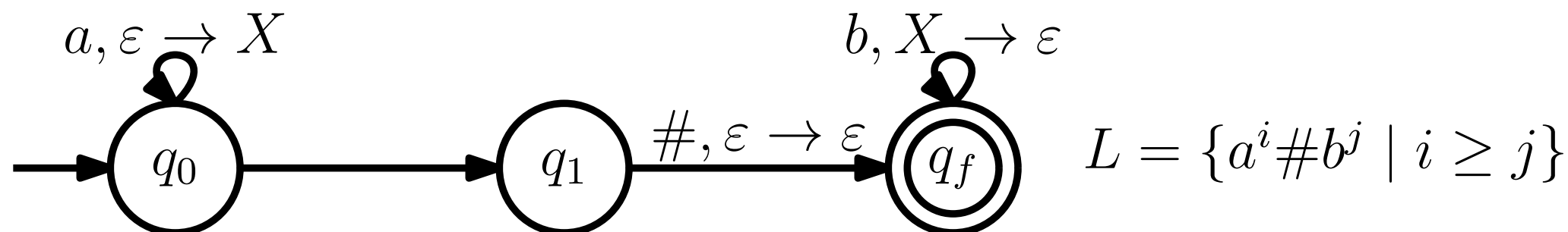
- A tiene un único estado final
- A tiene la pila vacía cuando acepta
- A sólo hace push y pop (y no replace)

- **Idea:**

- $A_{pq} \rightarrow A_{pr}A_{rq}$
- Apilamos y desapilamos el mismo símbolo: $A_{pq} \rightarrow aA_{rs}b$
- $A_{pp} \rightarrow \varepsilon$

- **Dem:**

- **Ej:**



Conclusión

- ¿Cuál es la relación con las GLC?

Conclusión

- ¿Cuál es la relación con las GLC?

Teo.

Sea $L \subseteq \Sigma^*$.

Conclusión

- ¿Cuál es la relación con las GLC?

Teo.

Sea $L \subseteq \Sigma^*$.

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

Conclusión

- ¿Cuál es la 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:** combinar ambos lemas.

Conclusión

- ¿Cuál es la 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:** combinar ambos lemas.
- AP y GLC son *equivalentes*.

Conclusión

- ¿Cuál es la 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:** combinar ambos lemas.
- AP y GLC son *equivalentes*.
- **Alt.:** todo lenguaje regular es libre de contexto.

Dudas/Preguntas

Wrap-Up

- **Hoy:**

Wrap-Up

- **Hoy:**
 - Equivalencia AP-GLC

Wrap-Up

- **Hoy:**

- Equivalencia AP-GLC
- $\text{GLC} \rightarrow \text{AP}$: simular las producciones por un lado.

Wrap-Up

- **Hoy:**

- Equivalencia AP-GLC
- $GLC \rightarrow AP$: simular las producciones por un lado.
- $AP \rightarrow GLC$: programación dinámica.

Wrap-Up

- **Hoy:**

- Equivalencia AP-GLC
- $GLC \rightarrow AP$: simular las producciones por un lado.
- $AP \rightarrow GLC$: programación dinámica.

- **Próx. clase:**

Wrap-Up

- **Hoy:**

- Equivalencia AP-GLC
- $GLC \rightarrow AP$: simular las producciones por un lado.
- $AP \rightarrow GLC$: programación dinámica.

- **Próx. clase:**

- Tópicos en lenguajes libres de contexto