

# ST0270 Lenguajes Formales y Compiladores

## Lenguajes libres de contexto y autómatas a pila

Andrés Sicard Ramírez

Universidad EAFIT

Semestre 2024-1

## Referencias

Las referencias principales para estas diapositivas son (Kozen 2012, Lecturas 19–21, 23, 24).

## Convenciones

- (i) Los números asignados a los definiciones, ejemplos, ejercicios, páginas y teoremas en estas diapositivas corresponden a los números asignados en (Kozen 2012).
- (ii) Los números naturales incluyen el cero, es decir,  $\mathbb{N} = \{0, 1, 2, \dots\}$ .
- (iii) El conjunto potencia de un conjunto  $A$  es denotado  $2^A$ .
- (iv) Los términos «definición inductiva» y «definición recursiva» se usan como sinónimos.

# Introducción

---

## Correspondencia entre las gramáticas y los modelos de computación

Un tipo de gramática genera una clase de lenguajes y un modelo de computación reconoce una clase de lenguajes.

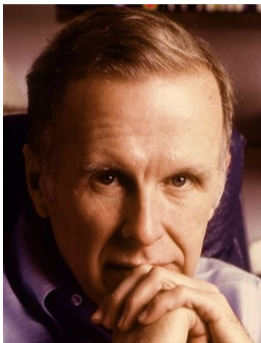
Tipo de gramática	Clase de lenguaje	Modelo de computación
3	Regulares	Autómatas finitos
2	Libres (independientes) de contexto	Autómatas a pila
1	Dependientes del contexto	Autómatas linealmente acotados
0	Recursivamente enumerables	Máquinas de Turing

# Formas de Backus-Naur

---

## Descripción

La **forma de Backus-Naur** (*Backus-Naur form*) (BNF) es un meta-lenguaje formal para especificar la sintaxis de un lenguaje libre de contexto. Una BNF es también una notación para denotar gramáticas libres de contexto.\*



John Backus (1924–2007)  
Recibió el premio Turing,  
1977



Peter Naur (1928–2016)  
Recibió el premio Turing,  
2005

---

\*Fotos tomadas de Wikipedia.

# Formas de Backus-Naur

---

## Ejemplo (BNF para programas simples estilo Pascal, pág. 129)

```

    <instr> ::= <instr-if> | <instr-while> | <instr-begin> | <instr-assign>
    <instr-if> ::= if <expr-bool> then <instr> else <instr>
    <instr-while> ::= while <expr-bool> do <instr>
    <instr-begin> ::= begin <lista-instr> end
    <lista-instr> ::= <instr> | <instr> ; <lista-instr>
    <instr-assig> ::= <var> := <expr-aritm>
    <expr-bool> ::= <expr-aritm> <op-comparación> <expr-aritm>
    <op-comparación> ::= < | > | ≤ | ≥ | = | ≠
    <expr-aritm> ::= <var> | <const> | ( <expr-aritm> <op-aritm> <expr-aritm> )
    <op-aritm> ::= + | - | * | /
    <const> ::= 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9
    <var> ::= a | b | c | ... | x | y | z
```

# Formas de Backus-Naur

---

## Ejemplo (continuación)

Las siguientes cadenas/programas son generados por la BNF anterior:

(i) `while x ≤ y do begin x := (x + 1) ; y := (y - 1) end`

(ii) `begin if z = (x + 3) then y := z else y := x end`

# Gramáticas libres de contexto

---

## Definición

Una **gramática libre de contexto** (o **gramática independiente del contexto**) (*context-free grammar*) es una estructura

$$G = (N, \Sigma, P, S),$$

donde

- (i)  $N$  es un conjunto **finito** (los **símbolos no terminales**),
- (ii)  $\Sigma$  es un conjunto **finito** (los **símbolos terminales**), los conjuntos  $\Sigma$  y  $N$  son disjuntos,
- (iii)  $P$  es un subconjunto **finito** de  $N \times (N \cup \Sigma)^*$  (las **producciones**),
- (iv)  $S \in N$  (el **símbolo inicial**).

# Gramáticas libres de contexto

---

## Convenciones

- Los símbolos no terminales serán denotados con letras mayúsculas  $A, B, C, \dots$
- Los símbolos terminales serán denotados con letras minúsculas  $a, b, c, \dots$
- Las cadenas en  $(N \cup \Sigma)^*$  serán denotadas por letras griegas minúsculas  $\alpha, \beta, \delta, \dots$
- Las producciones serán escritas  $A \rightarrow \alpha$  en lugar de  $(A, \alpha)$ .
- La barra vertical « | » significa lo mismo que en las BNFs.
- Las gramáticas independientes de contexto serán llamadas gramáticas.



# Gramáticas libres de contexto

---

## Definición

Sea  $G = (N, \Sigma, P, S)$  una gramática. La relación **derivable en un paso**, denotada  $\xrightarrow{1}_G$ , es una relación binaria sobre  $(N \cup \Sigma)^*$ , definida por:

Para todas  $\alpha, \beta \in (N \cup \Sigma)^*$ ,

$\alpha \xrightarrow{1}_G \beta$  sii existen cadenas  $\alpha_1, \alpha_2 \in (N \cup \Sigma)^*$  y una producción  $A \rightarrow \delta \in P$ , tales que  $\alpha = \alpha_1 A \alpha_2$  y  $\beta = \alpha_1 \delta \alpha_2$ .

# Gramáticas libres de contexto

---

## Definición

Sea  $G = (N, \Sigma, P, S)$  una gramática. La relación **derivable en un paso**, denotada  $\xrightarrow{1}_G$ , es una relación binaria sobre  $(N \cup \Sigma)^*$ , definida por:

Para todas  $\alpha, \beta \in (N \cup \Sigma)^*$ ,

$\alpha \xrightarrow{1}_G \beta$  sii existen cadenas  $\alpha_1, \alpha_2 \in (N \cup \Sigma)^*$  y una producción  $A \rightarrow \delta \in P$ , tales que  
 $\alpha = \alpha_1 A \alpha_2$  y  $\beta = \alpha_1 \delta \alpha_2$ .

Si  $\alpha \xrightarrow{1}_G \beta$  decimos que la cadena  $\beta$  es derivable desde la cadena  $\alpha$  en un paso.

# Gramáticas libres de contexto

---

## Definición

Sea  $G = (N, \Sigma, P, S)$  una gramática. La relación **derivable en  $n$  pasos**, denotada  $\xrightarrow[n]{G}$ , es una relación binaria sobre  $(N \cup \Sigma)^*$ , definida por:

Para todas  $\alpha, \beta \in (N \cup \Sigma)^*$ ,

$$\alpha \xrightarrow[0]{G} \beta \quad \text{sii} \quad \alpha = \beta,$$

$$\alpha \xrightarrow[n+1]{G} \beta \quad \text{sii} \quad \text{existe } \delta \text{ tal que } \alpha \xrightarrow[n]{G} \delta \text{ y } \delta \xrightarrow[1]{G} \beta.$$

# Gramáticas libres de contexto

---

## Definición

Sea  $G = (N, \Sigma, P, S)$  una gramática. La relación **derivable en  $n$  pasos**, denotada  $\xrightarrow[n]{G}$ , es una relación binaria sobre  $(N \cup \Sigma)^*$ , definida por:

Para todas  $\alpha, \beta \in (N \cup \Sigma)^*$ ,

$$\alpha \xrightarrow[0]{G} \beta \quad \text{sii} \quad \alpha = \beta,$$

$$\alpha \xrightarrow[n+1]{G} \beta \quad \text{sii} \quad \text{existe } \delta \text{ tal que } \alpha \xrightarrow[n]{G} \delta \text{ y } \delta \xrightarrow[1]{G} \beta.$$

Si  $\alpha \xrightarrow[n]{G} \beta$  decimos que la cadena  $\beta$  es derivable desde la cadena  $\alpha$  en  $n$  pasos.

# Gramáticas libres de contexto

---

## Definición

Sea  $G = (N, \Sigma, P, S)$  una gramática. La relación **derivable en cero o más pasos**, denotada  $\xrightarrow{*}_G$ , es una relación binaria sobre  $(N \cup \Sigma)^*$ , definida por:

Para todas  $\alpha, \beta \in (N \cup \Sigma)^*$ ,

$$\alpha \xrightarrow{*}_G \beta \quad \text{sii} \quad \alpha \xrightarrow{n}_G \beta, \text{ para algún } n \in \mathbb{N}.$$

# Gramáticas libres de contexto

---

## Definición

Sea  $G = (N, \Sigma, P, S)$  una gramática. La relación **derivable en cero o más pasos**, denotada  $\xrightarrow{*}_G$ , es una relación binaria sobre  $(N \cup \Sigma)^*$ , definida por:

Para todas  $\alpha, \beta \in (N \cup \Sigma)^*$ ,

$$\alpha \xrightarrow{*}_G \beta \quad \text{sii} \quad \alpha \xrightarrow{n}_G \beta, \text{ para algún } n \in \mathbb{N}.$$

Si  $\alpha \xrightarrow{*}_G \beta$  decimos que la cadena  $\beta$  es derivable desde la cadena  $\alpha$  en cero o más pasos.

# Gramáticas libres de contexto

---

## Definición

Sea  $G = (N, \Sigma, P, S)$  una gramática. La relación **derivable en cero o más pasos**, denotada  $\xrightarrow{*}_G$ , es una relación binaria sobre  $(N \cup \Sigma)^*$ , definida por:

Para todas  $\alpha, \beta \in (N \cup \Sigma)^*$ ,

$$\alpha \xrightarrow{*}_G \beta \quad \text{sii} \quad \alpha \xrightarrow{n}_G \beta, \text{ para algún } n \in \mathbb{N}.$$

Si  $\alpha \xrightarrow{*}_G \beta$  decimos que la cadena  $\beta$  es derivable desde la cadena  $\alpha$  en cero o más pasos.

## Observación

La relación  $\xrightarrow{*}_G$  es la cerradura reflexiva y transitiva de la relación  $\xrightarrow{1}_G$ .

# Gramáticas libres de contexto

---

## Definiciones

Sea  $G = (N, \Sigma, P, S)$  una gramática.

- (i) Una cadena  $\alpha \in (N \cup \Sigma)^*$  es una **forma sentencial** (*sentential form*) sii  $\alpha$  es derivable desde el símbolo inicial  $S$  de la gramática, es decir,  $S \xrightarrow[G]{*} \alpha$ .
- (ii) Una forma sentencial  $x$  es una **sentencia** (*sentence*) sii  $x$  solo contiene símbolos terminales, es decir,  $x \in \Sigma^*$ .



# Gramáticas libres de contexto

---

## Definición

Sea  $G = (N, \Sigma, P, S)$  una gramática. El **lenguaje generado por  $G$** , denotado  $L(G)$ , es el conjunto de sentencias derivables en  $G$ , es decir,

$$L(G) \stackrel{\text{def}}{=} \left\{ x \in \Sigma^* \mid S \xrightarrow[G]{*} x \right\}.$$

# Gramáticas libres de contexto

---

## Definición

Un lenguaje  $L$  es un **lenguaje libre de contexto** (o **lenguaje independiente del contexto**) (*context-free language*) sii existe una gramática libre de contexto  $G$  tal que  $L = L(G)$ .

# Gramáticas libres de contexto

---

## Ejemplo 19.1

El lenguaje  $L = \{a^n b^n \mid n \geq 0\}$  es no regular. Este lenguaje es libre de contexto. El lenguaje es generado por la gramática

$$S \rightarrow aSb \mid \epsilon$$

# Gramáticas libres de contexto

---

## Ejemplo 19.1

El lenguaje  $L = \{a^n b^n \mid n \geq 0\}$  es no regular. Este lenguaje es libre de contexto. El lenguaje es generado por la gramática

$$S \rightarrow aSb \mid \epsilon$$

Es decir, el lenguaje  $L$  es generado por la gramática  $G = (N, \Sigma, P, S)$ , donde

$$N = \{S\},$$

$$\Sigma = \{a, b\},$$

$$P = \{S \rightarrow aSb, S \rightarrow \epsilon\}.$$

(continua en la próxima diapositiva)

# Gramáticas libres de contexto

---

## Ejemplo 19.1 (continuación)

La derivación de  $a^3b^3 \in L(G)$  está dada por:

$$S \xrightarrow[G]{1} aSa \xrightarrow[G]{1} aaSbb \xrightarrow[G]{1} aaaSbbb \xrightarrow[G]{1} aaabbb,$$

es decir,

$$S \xrightarrow[G]{4} aaabbb.$$

# Gramáticas libres de contexto

---

## Definición

Un gramática  $G$  es **ambigua** sii existe una sentencia en  $L(G)$  que tiene más de una derivación (por el mismo lado).

# Gramáticas libres de contexto

---

## Ejemplo

Una gramática ambigua para expresiones aritméticas.

$$S \rightarrow S + S \mid S - S \mid S * S \mid S / S \mid \text{id}$$

# Gramáticas libres de contexto

---

## Ejemplo

La siguiente gramática ambigua genera el lenguaje  $L = \{\epsilon\}$ :

$$S \rightarrow S \mid \epsilon$$



# Gramáticas libres de contexto

---

## Observaciones

- (i) Los lenguajes regulares son lenguajes libres de contexto.

# Gramáticas libres de contexto

---

## Observaciones

- (i) Los lenguajes regulares son lenguajes libres de contexto.
- (ii) Existen lenguajes que no son libres de contexto, *p.* *ej.*  $\{ a^n b^n c^n \mid n \geq 0 \}$ .

# Gramáticas libres de contexto

---

## Observaciones

- (i) Los lenguajes regulares son lenguajes libres de contexto.
- (ii) Existen lenguajes que no son libres de contexto, *p.* *ej.*  $\{ a^n b^n c^n \mid n \geq 0 \}$ .
- (iii) El problema de determinar si una gramática es ambigua es un problema indecidible.

# Gramáticas libres de contexto

---

## Ejercicio (Homework 5, exercise 1)

The following table defines four special types of CFGs obtained by restricting productions to the form shown, where  $A$ ,  $B$  represent nonterminals,  $a$  a single terminal symbol, and  $x$  a string of terminals:

<u>Grammar type</u>	<u>Form of productions</u>
right-linear	$A \rightarrow xB$ or $A \rightarrow x$
strongly right-linear	$A \rightarrow aB$ or $A \rightarrow \epsilon$
left-linear	$A \rightarrow Bx$ or $A \rightarrow x$
strongly left-linear	$A \rightarrow Ba$ or $A \rightarrow \epsilon$

Prove that each of these four types of grammars generates exactly the regular sets. Conclude that every regular set is a CFL.

# Lenguaje de paréntesis balanceados

---

## Definición

En las expresiones matemáticas los paréntesis balanceados (*p. ej.*  $[[[]]$ ,  $[[[]]]$  y  $[[[]]][[]]$ ) son válidos y los paréntesis no balanceados (*p. ej.*  $] [ [ ] ]$  y  $[ [ ] ] [ [ [ ] ]$ ) son inválidos.

# Lenguaje de paréntesis balanceados

---

## Definición

En las expresiones matemáticas los paréntesis balanceados (p. ej.  $[[[]]$ ,  $[[[]]]$  y  $[[[]]][[]]$ ) son válidos y los paréntesis no balanceados (p. ej.  $] [[]]$  y  $[[[]]][[[]]$ ) son inválidos.

El lenguaje de los paréntesis balanceados **PAREN** es definido inductivamente por:

- (i) Paso base:  $\epsilon \in \text{PAREN}$ .
- (ii) Paso inductivo:
  - a) Si  $s \in \text{PAREN}$  entonces  $[ s ] \in \text{PAREN}$ .
  - b) Si  $s, t \in \text{PAREN}$  entonces  $st \in \text{PAREN}$ .
- (iii) Restricción: No hay elementos en **PAREN** distintos a los obtenidos de (i) y (ii).

(continua en la próxima diapositiva)

# Lenguaje de paréntesis balanceados

---

## Definición (continuación)

Es decir, el lenguaje **PAREN** es definido por las siguientes reglas de inferencia:

$$\frac{}{\epsilon \in \text{PAREN},}$$

$$\frac{s \in \text{PAREN}}{[ s ] \in \text{PAREN},}$$

$$\frac{s, t \in \text{PAREN}}{st \in \text{PAREN}.}$$

# Lenguaje de paréntesis balanceados

---

## Gramática

Una gramática  $G$  para el lenguaje PAREN es

$$S \rightarrow [ S ] \mid SS \mid \epsilon$$



# Lenguaje de paréntesis balanceados

---

## Gramática

Una gramática  $G$  para el lenguaje PAREN es

$$S \rightarrow [ S ] \mid SS \mid \epsilon$$

## Teorema 20.1

$$L(G) = \{ x \in \{ [, ] \}^* \mid x \in \text{PAREN} \}.$$

# Lenguaje de paréntesis balanceados

---

## Gramática

Una gramática  $G$  para el lenguaje PAREN es

$$S \rightarrow [ S ] \mid SS \mid \epsilon$$

## Teorema 20.1

$$L(G) = \{ x \in \{[, ]\}^* \mid x \in \text{PAREN} \}.$$

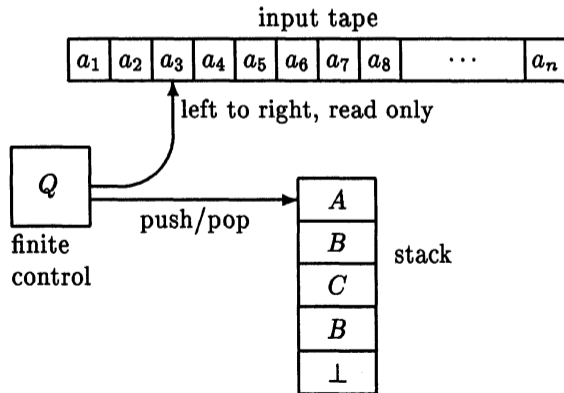
## Demostración

En el tablero.

# Autómatas a pila

## Descripción

Un autómata a pila no determinista es similar a un autómata finito no determinista, pero tiene además una pila que puede ser usada para almacenar una cantidad no acotada de información. Fig. pág. 157.



# Autómatas a pila

---

## Definición

Un **autómata a pila no determinista** (APND) (*nondeterministic pushdown automaton*) es una estructura

$$P = (Q, \Sigma, \Gamma, \delta, s, \perp, F),$$

donde

- (i)  $Q$  es un conjunto finito, los **estados**,
- (ii)  $\Sigma$  es un conjunto finito, el **alfabeto de entrada**,
- (iii)  $\Gamma$  es un conjunto finito, el **alfabeto de la pila**,
- (iv)  $\delta$  es una relación finita de  $Q \times (\Sigma \cup \{\epsilon\}) \times \Gamma$  en  $Q \times \Gamma^*$ , la **relación de transición**,
- (v)  $s \in Q$ , el **estado inicial**,
- (vi)  $\perp \in \Gamma$ , el **símbolo inicial de la pila**,
- (vii)  $F \subseteq Q$ , el conjunto de **estados finales** o **estados de aceptación**.

# Autómatas a pila

---

## Transiciones

Sea  $P = (Q, \Sigma, \Gamma, \delta, s, \perp, F)$  un APND. Las posibles transiciones de  $P$  son de dos tipos:

(i) Procesando un símbolo del alfabeto de entrada

$$((p, a, A), (q, B_1 B_2 \dots B_k)) \in \delta.$$

(ii) Procesando la palabra vacía

$$((p, \epsilon, A), (q, B_1 B_2 \dots B_k)) \in \delta.$$

# Autómatas a pila

---

## Definición

Una **configuración** de un APND es un elemento de  $Q \times \Sigma^* \times \Gamma^*$ . Una configuración describe el estado actual de autómatata, la parte de la entrada que no ha sido procesada y el contenido de la pila.

# Autómatas a pila

---

## Definición

Una **configuración** de un APND es un elemento de  $Q \times \Sigma^* \times \Gamma^*$ . Una configuración describe el estado actual de autómatata, la parte de la entrada que no ha sido procesada y el contenido de la pila.

## Ejemplo

En el tablero.

# Autómatas a pila

---

## Definición

Una **configuración** de un APND es un elemento de  $Q \times \Sigma^* \times \Gamma^*$ . Una configuración describe el estado actual de autómatas, la parte de la entrada que no ha sido procesada y el contenido de la pila.

## Ejemplo

En el tablero.

## Definición

La **configuración inicial** para una palabra de entrada  $x \in \Sigma^*$  es  $(s, x, \perp)$ .



# Autómatas a pila

---

## Definición

Sea  $P = (Q, \Sigma, \Gamma, \delta, s, \perp, F)$  un APND. La relación **siguiente configuración en un paso**, denotada  $\xrightarrow{1}_P$ , es una relación binaria entre configuraciones  $Q \times \Sigma^* \times \Gamma^*$  definida por:

(i) Si  $((p, a, A), (q, \gamma)) \in \delta$ , entonces para toda  $y \in \Sigma^*$  y para toda  $\beta \in \Gamma^*$ ,

$$(p, ay, A\beta) \xrightarrow{1}_P (q, y, \gamma\beta).$$

(ii) Si  $((p, \epsilon, A), (q, \gamma)) \in \delta$ , entonces para toda  $y \in \Sigma^*$  y para toda  $\beta \in \Gamma^*$ ,

$$(p, y, A\beta) \xrightarrow{1}_P (q, y, \gamma\beta).$$

# Autómatas a pila

---

## Definición

Sea  $P = (Q, \Sigma, \Gamma, \delta, s, \perp, F)$  un APND y sea  $n \in \mathbb{N}$ . La relación **siguiente configuración en  $n$  pasos**, denotada  $\xrightarrow[n]{P}$ , es una relación binaria entre configuraciones  $Q \times \Sigma^* \times \Gamma^*$  definida inductivamente por:

Para todas las configuraciones  $C$  y  $D$ ,

$$C \xrightarrow[P]{0} D \quad \text{sii} \quad C = D,$$

$$C \xrightarrow[P]{n+1} D \quad \text{sii} \quad \text{existe una configuración } E \text{ tal que } C \xrightarrow[P]{n} E \text{ y } E \xrightarrow[P]{1} D.$$

# Autómatas a pila

---

## Definición

Sea  $P = (Q, \Sigma, \Gamma, \delta, s, \perp, F)$  un APND. La relación **siguiente configuración en cero o más pasos**, denotada  $\xrightarrow{*}_P$ , es una relación binaria entre configuraciones  $Q \times \Sigma^* \times \Gamma^*$  definida por:

Para todas las configuraciones  $C$  y  $D$ ,

$$C \xrightarrow{*}_P D \quad \text{sii} \quad C \xrightarrow{n}_P D, \text{ para algún } n \in \mathbb{N}.$$

# Autómatas a pila

---

## Definición

Sea  $P = (Q, \Sigma, \Gamma, \delta, s, \perp, F)$  un APND. La relación **siguiente configuración en cero o más pasos**, denotada  $\xrightarrow{*}_P$ , es una relación binaria entre configuraciones  $Q \times \Sigma^* \times \Gamma^*$  definida por:

Para todas las configuraciones  $C$  y  $D$ ,

$$C \xrightarrow{*}_P D \quad \text{sii} \quad C \xrightarrow{n}_P D, \text{ para algún } n \in \mathbb{N}.$$

## Observación

La relación  $\xrightarrow{*}_P$  es la cerradura reflexiva y transitiva de la relación  $\xrightarrow{1}_P$ .

# Autómatas a pila

---

## Definición

Sea  $P = (Q, \Sigma, \Gamma, \delta, s, \perp, F)$  un APND y sea  $x \in \Sigma^*$ .

(i) El autómata  $P$  **acepta** la palabra  $x$  **por pila vacía** sii

$$(s, x, \perp) \xrightarrow{*}_P (q, \epsilon, \epsilon), \text{ para algún } q \in Q.$$

(ii) El autómata  $P$  **acepta** la palabra  $x$  **por estado de aceptación** sii

$$(s, x, \perp) \xrightarrow{*}_P (q, \epsilon, \gamma), \text{ para algún } q \in F \text{ y } \gamma \in \Gamma^*.$$

# Autómatas a pila

---

## Definición

Sea  $P = (Q, \Sigma, \Gamma, \delta, s, \perp, F)$  un APND.

(i) El **lenguaje aceptado** por el autómata  $P$  por **pila vacía** está definido por:

$$L(P) \stackrel{\text{def}}{=} \left\{ x \in \Sigma^* \mid (s, x, \perp) \xrightarrow{*}_P (q, \epsilon, \epsilon), \text{ para algún } q \in Q \right\}.$$

(ii) El **lenguaje aceptado** por el autómata  $P$  por **estado de aceptación** está definido por:

$$L(P) \stackrel{\text{def}}{=} \left\{ x \in \Sigma^* \mid (s, x, \perp) \xrightarrow{*}_P (q, \epsilon, \gamma), \text{ para algún } q \in F \text{ y } \gamma \in \Gamma^* \right\}.$$

## Ejemplo 23.1

El APND  $P$  acepta el lenguaje PAREN por pila vacía:

$$P = (\{q\}, \{[, ]\}, \{[, \perp\}, \delta, q, \perp, \emptyset),$$

donde  $\delta$  consiste de las siguientes transiciones:

- (i)  $((q, [, \perp), (q, [ \perp)),$
- (ii)  $((q, [, [), (q, [ [)),$
- (iii)  $((q, ], [), (q, \epsilon),$
- (iv)  $((q, \epsilon, \perp), (q, \epsilon)).$

# Autómatas a pila

## Ejemplo 23.1

El APND  $P$  acepta el lenguaje **PAREN** por pila vacía:

$$P = (\{q\}, \{[, ]\}, \{[, \perp\}, \delta, q, \perp, \emptyset),$$

donde  $\delta$  consiste de las siguientes transiciones:

- (i)  $((q, [, \perp), (q, [ \perp)),$
- (ii)  $((q, [, [), (q, [ [)),$
- (iii)  $((q, ], [), (q, \epsilon),$
- (iv)  $((q, \epsilon, \perp), (q, \epsilon)).$



(imagen generada por JFLAP)



# Gramáticas libres de contexto y autómatas a pila

---

## Introducción

Las gramáticas libres de contexto y los autómatas a pila no deterministas son equivalentes en potencia expresiva: Los lenguajes generados por las gramáticas libres de contexto, es decir, los lenguajes libres de contexto, son exactamente los lenguajes aceptados por los autómatas a pila no deterministas.

# Gramáticas libres de contexto y autómatas a pila

## Ejemplo (lenguaje de paréntesis balanceados no nulos)

		<i>Sentential forms in a leftmost derivation of <math>x</math> in <math>G</math></i>	<i>Configurations of <math>M</math> in an accepting computation of <math>M</math> on input <math>x</math></i>
	<i>Rule applied</i>		
(i) $S \rightarrow [BS$	$((q, [, S), (q, BS))$	$S$	$(q, [ [ [ ] ] [ ] ], S)$
(ii) $S \rightarrow [B$	$((q, [, S), (q, B))$	$[SB$	$(q, [ [ ] ] [ ] ], SB)$
(iii) $S \rightarrow [SB$	$((q, [, S), (q, SB))$	$[ [SBSB$	$(q, [ ] ] [ ] ], SBSB)$
(iv) $S \rightarrow [SBS$	$((q, [, S), (q, SBS))$	$[ [ [BBSB$	$(q, ] ] [ ] ], BBSB)$
(v) $B \rightarrow ]$	$((q, ], S), (q, \epsilon))$	$[ [ [ ] ]BSB$	$(q, ] [ ] ], BSB)$
		$[ [ [ ] ]SB$	$(q, [ ] ], SB)$
		$[ [ [ ] ] ]BB$	$(q, ] ], BB)$
		$[ [ [ ] ] [ ]B$	$(q, ] , B)$
		$[ [ [ ] ] [ ] ]$	$(q, \epsilon, \epsilon)$

(Tabla pág. 169)

# Gramáticas libres de contexto y autómatas a pila

---

## Lema 24.1

Sean  $z, y \in \Sigma^*$ ,  $\gamma \in N^*$  y  $A \in N$ .

$A \xrightarrow[G]{n} z\gamma$  vía una derivación más a la izquierda   sii    $(q, zy, A) \xrightarrow[P]{n} (q, y, \gamma)$ .

# Gramáticas libres de contexto y autómatas a pila

---

## Lema 24.1

Sean  $z, y \in \Sigma^*$ ,  $\gamma \in N^*$  y  $A \in N$ .

$A \xrightarrow[G]{n} z\gamma$  vía una derivación más a la izquierda    sii     $(q, zy, A) \xrightarrow[P]{n} (q, y, \gamma)$ .

Demostración (por inducción en  $n$ )

En el tablero.

# Gramáticas libres de contexto y autómatas a pila

---

Teorema 24.2

$$L(P) = L(G).$$

# Gramáticas libres de contexto y autómatas a pila

---

Teorema 24.2

$$L(P) = L(G).$$

Demostración

En el tablero.

# Referencias

---



Kozen, Dexter C. [1997] (2012). *Automata and Computability*. Third printing. Undergraduate Texts in Computer Science. Springer. DOI: [10.1007/978-1-4612-1844-9](https://doi.org/10.1007/978-1-4612-1844-9) (vid. pág. 2).