ST0270 Lenguajes Formales y Compiladores
Lenguajes libres de contexto y autómatas a pila

Andrés Sicard Ramírez

Universidad EAFIT

Semestre 2024-1

(Última actualización: 25 de julio de 2025)

Preliminares

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

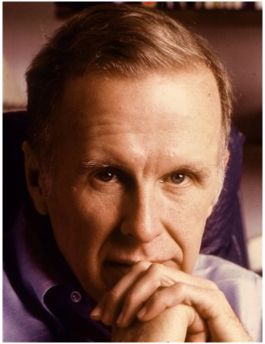
Un tipo de gramática genera una clase de lenguajes y un modelo de computación reconoce una clase de lenguajes.

Tipo de gramática	Clase de lenguaje	Modelo de computación
3	Regulares	Autómatas finitos
2	Libres 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)

$$
\begin{aligned}
\langle \text{instr} \rangle &::= \langle \text{instr-if} \rangle \mid \langle \text{instr-while} \rangle \mid \langle \text{instr-begin} \rangle \mid \langle \text{instr-assign} \rangle \\
\langle \text{instr-if} \rangle &::= \texttt{if}\ \langle \text{expr-bool} \rangle\ \texttt{then}\ \langle \text{instr} \rangle\ \texttt{else}\ \langle \text{instr} \rangle \\
\langle \text{instr-while} \rangle &::= \texttt{while}\ \langle \text{expr-bool} \rangle\ \texttt{do}\ \langle \text{instr} \rangle \\
\langle \text{instr-begin} \rangle &::= \texttt{begin}\ \langle \text{lista-instr} \rangle\ \texttt{end} \\
\langle \text{lista-instr} \rangle &::= \langle \text{instr} \rangle \mid \langle \text{instr} \rangle\ \texttt{;}\ \langle \text{lista-instr} \rangle \\
\langle \text{instr-assig} \rangle &::= \langle \text{var} \rangle\ \texttt{:=}\ \langle \text{expr-aritm} \rangle \\
\langle \text{expr-bool} \rangle &::= \langle \text{expr-aritm} \rangle \langle \text{op-comparación} \rangle \langle \text{expr-aritm} \rangle \\
\langle \text{op-comparación} \rangle &::= \texttt{<} \mid \texttt{>} \mid \leq \mid \geq \mid \texttt{=} \mid \neq \\
\langle \text{expr-aritm} \rangle &::= \langle \text{var} \rangle \mid \langle \text{const} \rangle \mid \texttt{(}\ \langle \text{expr-aritm} \rangle \langle \text{op-aritm} \rangle \langle \text{expr-aritm} \rangle\ \texttt{)} \\
\langle \text{op-aritm} \rangle &::= \texttt{+} \mid \texttt{-} \mid \texttt{*} \mid \texttt{/} \\
\langle \text{const} \rangle &::= \texttt{0} \mid \texttt{1} \mid \texttt{2} \mid \texttt{3} \mid \texttt{4} \mid \texttt{5} \mid \texttt{6} \mid \texttt{7} \mid \texttt{8} \mid \texttt{9} \\
\langle \text{var} \rangle &::= \texttt{a} \mid \texttt{b} \mid \texttt{c} \mid \ldots \mid \texttt{x} \mid \texttt{y} \mid \texttt{z}
\end{aligned}
$$

Formas de Backus-Naur

Ejemplo (continuación)

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

(i) `while x` $\leq$ `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) Σ es un conjunto finito (los **símbolos terminales**), los conjuntos Σ 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 $A, B, C, \ldots$.
- Los símbolos terminales serán denotados con letras minúsculas $a, b, c, \ldots$.
- Las cadenas en $(N \cup \Sigma)^*$ serán denotadas por letras griegas minúsculas $\alpha, \beta, \delta, \ldots$.
- Las producciones serán escritas $A \to \alpha$ en lugar de (A, α).
- 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 $\xrightarrow[G]{1}$, es una relación binaria sobre $(N \cup \Sigma)^*$, definida por:

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

$\alpha \xrightarrow[G]{1} \beta$ sii existen cadenas $\alpha_1, \alpha_2 \in (N \cup \Sigma)^*$ y una producción $A \to \delta \in P,$ 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 $\xrightarrow[G]{1}$, es una relación binaria sobre $(N \cup \Sigma)^*$, definida por:

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

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

Si $\alpha \xrightarrow[G]{1} \beta$ decimos que la cadena β es derivable desde la cadena α en un paso.

Gramáticas libres de contexto

Definición

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

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

$$\begin{aligned} \alpha \xrightarrow[G]{0} \beta \quad &\text{sii} \quad \alpha = \beta, \\ \alpha \xrightarrow[G]{n+1} \beta \quad &\text{sii} \quad \text{existe } \delta \text{ tal que } \alpha \xrightarrow[G]{n} \delta \text{ y } \delta \xrightarrow[G]{1} \beta. \end{aligned}$$

Gramáticas libres de contexto

Definición

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

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

$$\begin{aligned} \alpha \xrightarrow[G]{0} \beta \quad & \text{sii} \quad \alpha = \beta, \\ \alpha \xrightarrow[G]{n+1} \beta \quad & \text{sii} \quad \text{existe } \delta \text{ tal que } \alpha \xrightarrow[G]{n} \delta \text{ y } \delta \xrightarrow[G]{1} \beta. \end{aligned}$$

Si $\alpha \xrightarrow[G]{n} \beta$ decimos que la cadena β es derivable desde la cadena α 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 $\alpha, \beta \in (N \cup \Sigma)^*$,

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

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

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

Si $\alpha \xrightarrow[G]{*} \beta$ decimos que la cadena β es derivable desde la cadena α 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 $\alpha, \beta \in (N \cup \Sigma)^*$,

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

Si $\alpha \xrightarrow[G]{*} \beta$ decimos que la cadena β es derivable desde la cadena α en cero o más pasos.

Observación

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

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 α 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 $\boldsymbol{G}$**, denotado $\mathrm{L}(G)$, es el conjunto de sentencias derivables en G, es decir,

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

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 $L = \mathrm{L}(G)$.

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 es generado por la gramática

$$S \to aSb \mid \epsilon$$

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 es generado por la gramática

$$S \to aSb \mid \epsilon$$

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

$$\begin{aligned} N &= \{S\}, \\ \Sigma &= \{a, b\}, \\ P &= \{S \to aSb, S \to \epsilon\}. \end{aligned}$$

(continua en la próxima diapositiva)

Ejemplo 19.1 (continuación)

La derivación de $a^3b^3 \in \mathrm{L}(G)$ 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.

$$S \to S + S \mid S - S \mid S * S \mid S/S \mid \mathrm{id}$$

Gramáticas libres de contexto

Ejemplo

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

$$S \to S \mid \epsilon$$

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:

Grammar type	Form of productions
right-linear	$A \to xB$ or $A \to x$
strongly right-linear	$A \to aB$ or $A \to \epsilon$
left-linear	$A \to Bx$ or $A \to x$
strongly left-linear	$A \to Ba$ or $A \to \epsilon$

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: $\epsilon \in \mathrm{PAREN}$.

(ii) Paso inductivo:

  a) Si $s \in \mathrm{PAREN}$ entonces $[\,s\,] \in \mathrm{PAREN}$.

  b) Si $s, t \in \mathrm{PAREN}$ entonces $st \in \mathrm{PAREN}$.

(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:

$$\frac{}{\epsilon \in \mathrm{PAREN}}, \qquad \frac{s \in \mathrm{PAREN}}{[\, s\,] \in \mathrm{PAREN}}, \qquad \frac{s, t \in \mathrm{PAREN}}{st \in \mathrm{PAREN}}.$$

Lenguaje de paréntesis balanceados

Gramática

Una gramática G para el lenguaje PAREN es

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

Lenguaje de paréntesis balanceados

Gramática

Una gramática G para el lenguaje PAREN es

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

Teorema 20.1

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

Lenguaje de paréntesis balanceados

Gramática

Una gramática G para el lenguaje PAREN es

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

Teorema 20.1

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

Demostración

En el tablero.

Formas normales

Definición 21.1

Una gramática libre de contexto está en **forma normal de Chomsky** sii todas sus producciones son de la forma

$$A \to BC \quad \text{o} \quad A \to a,$$

donde $A, B, C \in N$ y $a \in \Sigma$.

Formas normales

Ejemplo

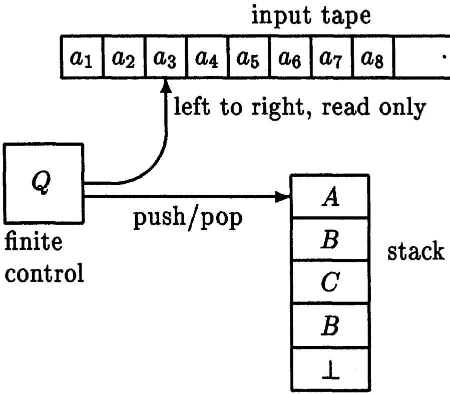
Una gramática en forma normal de Chomsky para el lenguaje de paréntesis balanceados no nulos.

$$
\begin{aligned}
S &\to AB \mid AC \mid SS \\
C &\to SB \\
A &\to [\\
B &\to]
\end{aligned}
$$

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) Σ es un conjunto finito, el **alfabeto de entrada**,

(iii) Γ es un conjunto finito, el **alfabeto de la pila**,

(iv) δ 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

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

(ii) Procesando la palabra vacía

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

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 autómata, 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 autómata, 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 autómata, 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 $\xrightarrow[P]{1}$, 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 $\beta \in \Gamma^*$,

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

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

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

Definición

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

Para todas las configuraciones C y D,

$$
\begin{aligned}
C \xrightarrow[P]{0} D \quad & \text{sii} \quad C = D, \\
C \xrightarrow[P]{n+1} D \quad & \text{sii} \quad \text{existe una configuración } E \text{ tal que } C \xrightarrow[P]{n} E \text{ y } E \xrightarrow[P]{1} D.
\end{aligned}
$$

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,

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

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,

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

Observación

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

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:

$$\mathrm{L}(P) \stackrel{\text{def}}{=} \left\{ x \in \Sigma^* \;\middle|\; (s, x, \perp) \xrightarrow[P]{*} (q, \epsilon, \epsilon), \text{ para algún } q \in Q \right\}.$$

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

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

Autómatas a pila

Ejemplo 23.1

El APND P acepta el lenguaje PAREN por pila vacía:

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

donde δ consiste de las siguientes transiciones:

(i) $((q, [, \bot), (q, [\,\bot))$,

(ii) $((q, [, [\,), (q, [\,[\,))$,

(iii) $((q,], [\,), (q, \epsilon))$,

(iv) $((q, \epsilon, \bot), (q, \epsilon))$.

Autómatas a pila

Ejemplo 23.1

El APND P acepta el lenguaje PAREN por pila vacía:

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

donde δ consiste de las siguientes transicione

(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)	$S \to [\,BS$	$((q, [, S), (q, BS))$
(ii)	$S \to [\,B$	$((q, [, S), (q, B))$
(iii)	$S \to [\,SB$	$((q, [, S), (q, SB))$
(iv)	$S \to [\,SBS$	$((q, [, S), (q, SBS))$
(v)	$B \to]$	$((q,], S), (q, \epsilon))$

Rule applied	*Sentential forms in a leftmost derivation of x in G*	*Configurations of M in an accepting computation of M on input x*
	S	$(q,$ [[[]][]], $S)$
(iii)	[SB	$(q,$ [[]][]], $SB)$
(iv)	[[$SBSB$	$(q,$ []][]], $SBSB)$
(ii)	[[[$BBSB$	$(q,$]][]], $BBSB)$
(v)	[[[]BSB	$(q,$][]], $BSB)$
(v)	[[[]]SB	$(q,$ []], $SB)$
(ii)	[[[]][BB	$(q,$]], $BB)$
(v)	[[[]][]B	$(q,$], $B)$
(v)	[[[]][]]	(q, ϵ, ϵ)

(Tabla pág. 169)

Gramáticas libres de contexto y autómatas a pila

Lema 24.1

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

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

Lema 24.1

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

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

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

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