Lenguajes Formales, Autómatas y Computabilidad

Clase Teórica

Autómatas de Pila y Lenguajes libres de contexto

Segundo Cuatrimestre 2025

Bibliografía: Capítulos 6 y 7, *Introduction to Automata Theory, Languages and Computation*, J. Hopcroft, R. Motwani, J. Ullman, Second Edition, Addison Wesley, 2001.

Definición (Oettinger 1961, Schutzenberger 1963)

Un autómata de pila está definido por $M = \langle Q, \Sigma, \Gamma, \delta, q_0, Z_0, F \rangle$ *donde*

- Q *es el conjunto de estados*
- Σ *es el alfabeto de entrada*
- Γ *es el alfabeto de la pila*
- $q_0 \in Q$ *es el estado inicial*
- $Z_0 \in \Gamma$ *es la configuración incial de la pila*
- $F \in Q$ *es el conjunto de estados finales*
- $\delta : Q \times (\Sigma \cup \{\lambda\}) \times \Gamma \to \mathcal{P}\left(Q \times \Gamma^*\right)$ *función de transición:*
  $\delta\left(q, a, Z\right) = \{(p_1, \gamma_1), (p_2, \gamma_2), \ldots, (p_m, \gamma_m)\}$

La interpretación de $\delta(q, a, Z) = \{(p_1, \gamma_1), (p_2, \gamma_2), \ldots, (p_n, \gamma_n)\}$, con $q, p_1, \ldots, p_n \in Q$, $a \in (\Sigma \cup \{\lambda\})$, $Z \in \Gamma$, y $\gamma_i \in \Gamma^*$ es la siguiente.

Cuando el estado del autómata es q, el símbolo que la cabeza lectora está inspeccionando en ese momento es a, y en el tope de la pila nos encontramos el símbolo Z, se realizan las siguientes acciones:

Si $a \in \Sigma$, es decir no es la cadena vacía, la cabeza lectora avanza una posición para inspeccionar el siguiente símbolo.

Se elimina el símbolo Z de la pila del autómata.

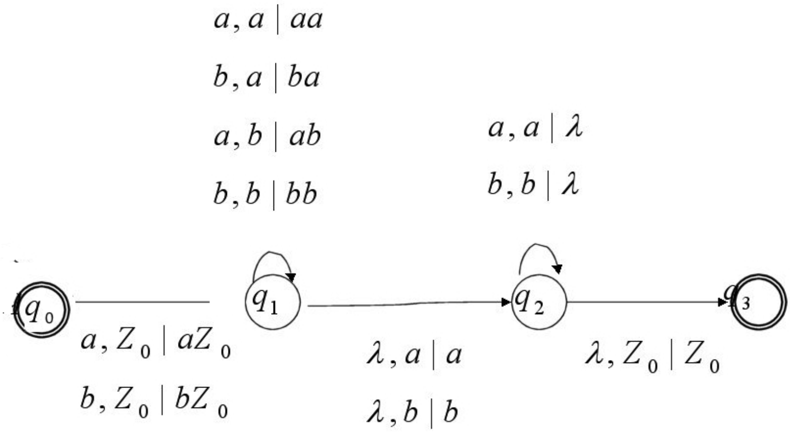
Se selecciona un (p_i, γ_i) entre los existentes en la definición de $\delta(q, a, Z)$.

Se apila la cadena $\gamma_i = c_1 c_2 \ldots c_k$, con $c_i \in \Gamma$ en la pila del autómata, quedando el símbolo c_1 en el tope de la pila.

El estado actual pasa a ser p_i.

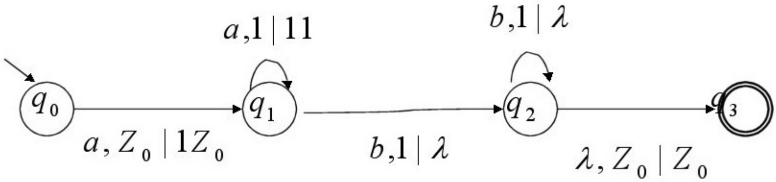
Ejemplo

Autómata de pila que acepta $L = \{\alpha\alpha^R : \alpha \in \Sigma^*\}$:
$M = \langle Q, \Sigma, \Gamma, \delta, q_0, Z_0, F\rangle$ donde
$Q = \{q_0, q_1, q_2, q_3\}$ $\Sigma = \{a, b\}$, $\Gamma = \{Z_0, a, b\}$ $F = \{q_0, q_3\}$



Ejemplo

Autómata de pila que acepta $L = \{a^n b^n : n \geq 1\}$:
$M = \langle Q, \Sigma, \Gamma, \delta, q_0, Z_0, F \rangle$ donde
$Q = \{q_0, q_1, q_2, q_3\}$ $\Sigma = \{a, b\}$, $\Gamma = \{Z_0, 1\}$ $F = \{q_3\}$



Autómatas de Pila Determinísticos

Definición

Un autómata de pila $M = \langle Q, \Sigma, \Gamma, \delta, q_0, Z_0, F \rangle$ *es determinístico si,* $\forall q \in Q, \forall a \in \Sigma, \forall A \in \Gamma$*,*

1. $\#\delta\,(q, a, A) \leq 1$
2. $\#\delta\,(q, \lambda, A) \leq 1$
3. *si* $\#\delta\,(q, \lambda, A) = 1$ *entonces* $\#\delta\,(q, a, A) = 0$

Teorema

No es cierto que para cada autómata de pila no determinístico existe otro determinístico que acepta el mismo lenguaje.
Es decir, hay lenguajes libres de contexto que no son aceptados por ningún autómata de pila determinístico.

Demostración. $L = \{ww^R\}$ es aceptado por AP no-determinístico pero no es aceptado por ningun AP determinístico ver Hopcroft, Motwani Ulman (2001) página 249:

On the other hand, there are CFL's like L_{wwr} that cannot be $L(P)$ for any DPDA P. A formal proof is complex, but the intuition is transparent. If P is a DPDA accepting L_{wwr}, then given a sequence of 0's, it must store them on the stack, or do something equivalent to count an arbitrary number of 0's. For instance, it could store one X for every two 0's it sees, and use the state to remember whether the number was even or odd.

Suppose P has seen n 0's and then sees 110^n. It must verify that there were n 0's after the 11, and to do so it must pop its stack.[5] Now, P has seen 0^n110^n. If it sees an identical string next, it must accept, because the complete input is of the form ww^R, with $w = 0^n110^n$. However, if it sees 0^m110^m for some $m \neq n$, P must *not* accept. Since its stack is empty, it cannot remember what arbitrary integer n was, and must fail to recognize L_{wwr} correctly. Our conclusion is that:

- The languages accepted by DPDA's by final state properly include the regular languages, but are properly included in the CFL's.

Sea un autómata de pila $M = \langle Q, \Sigma, \Gamma, \delta, q_0, Z_0, F \rangle$.

Definición (Configuración de un autómata de pila)

Es una tripla de $Q \times \Sigma^* \times \Gamma^*$ *donde* Q *es el estado actual,* Σ^* *es la parte de la cadena de entrada que falta consumir y* Γ^* *es el contenido de la pila. La configuración inicial del autómata es* (q_0, α, Z_0)*, para alguna cadena* $\alpha \in \Sigma^*$.

Definición (Cambio de configuración $\vdash$)

Para todo $a \in \Sigma$, $\alpha \in \Sigma^*$,$t \in \Gamma$, $\tau, \pi \in \Gamma^*$, $q, r \in Q$

- *Si* $(r, \tau) \in \delta(q, a, t)$ *entonces* $(q, a\alpha, t\pi) \vdash (r, \alpha, \tau\pi)$.
- *Si* $(r, \tau) \in \delta(q, \lambda, t)$ *entonces* $(q, \alpha, t\pi) \vdash (r, \alpha, \tau\pi)$.

Definición

El lenguaje aceptado por el autómata de pila $M = \langle Q, \Sigma, \Gamma, \delta, q_0, Z_0, F \rangle$ *es*

$$\mathcal{L}(M) = \left\{ \alpha \in \Sigma^* : \exists p \in F \quad (q_0, \alpha, Z_0) \overset{*}{\vdash} (p, \lambda, \gamma) \right\}$$

Definición

Los lenguajes libre sde contexto son exactamente lso lenguajes aceptados por autómatas de pila.

Los lenguajes libre sd econtexto determinístico son la subclase de lenguajes libres de contexto aceptados por autómtas de pila determinísticos

Ya vimos que los autómats de pila reconocen lenguajes que no son regulares, como por ejemplo $\{a^n b^n : n \geq 0\}$.

Teorema

Los lenguajes regulares son libres de contexto.

Es decir, para todo autómata finito M exite un autómata de pila P tal que $\mathcal{L}(M) = \mathcal{L}(P)$.

Lema (“Pumping” para lenguajes libres de contexto.)

Para todo lenguaje L *libre de contexto, existe* $n > 0$ *tal que para toda cadena* α *en* L *con* $|\alpha| \geq n$,

- *Existe una descomposición de* α *en cadenas* r, x, y, z, s, *es decir* $\alpha = rxyzs$.
- $|xyz| \leq n$.
- $|xz| \geq 1$.
- *Para todo* $i \geq 0$, *la cadena* rx^iyz^is *pertenece a* L.

Ejemplo

$L = \{a^k b^k c^k : \ k \geq 0\}$ *no es libre de contexto*

Demostración.

Supongamos L es libre de contexto. Sea n la constante del Lema de Pumping.
Sea $w = a^n b^n c^n$. Por el Lema de Pumping, hay una descomposicion $w = rxyzs$ con $|xyz| \leq n$ y $|xz| \geq 1$ tal que $rx^i yz^i sw$ en L, para todo $i \geq 0$.

Dado que $|xyz| \leq n$, xyz no puede tener as, bs y cs.
Como $|xz| \geq 1$, xz tiene alguna de las tres letras, pero no las tres.
Por Lema de Pumping, $rx^0 yz^0 s = rys$ está en L. Pero esto es imposible porque xy no tiene la misma cantidad de las tres letras por lo tanto rys no tiene la misma cantidad de as, bs y cs.
Llegamos a la imposibilidad porque supusimos que L era libre de contexto. Entonces concluimos que no lo es. $\square$

Propiedades de Clausura

Teorema

Los lenguajes libres de contexto están cerrados por unión, concatenación, reversa y clausura de Kleene.

No están cerrados por intersección, diferencia, complemento.

Sin embargo la intersección de libre de contexto con regular es libre de contexto.

Union:

Dados AP $A = (Q_A, \Sigma, \Gamma_A, \delta_A, q_{0A}, Z_{0A}, F_A)$ y AP
$B = (Q_B, \Sigma, \Gamma_B, \delta_B, q_{0B}, Z_{0B}, F_B)$

Tenemos

$\mathcal{L}(A) = \{w \in \Sigma^* : (q_{0A}, w, Z_{0A}) \overset{*}{\vdash} (q, \lambda, v), v \in \Gamma_A^*, q \in F_A\}$

$\mathcal{L}(B) = \{w \in \Sigma^* : (q_{0B}, w, Z_{0B}) \overset{*}{\vdash} (q, \lambda, v), v \in \Gamma_B^*, q \in F_B\}$

Construimos AP $C = (Q, \Sigma, \Gamma, \delta, q_0, Z_0, F)$
donde

$Q = Q_A \cup Q_B \cup \{q_0\}$
$\Gamma = \Gamma_A \cup \Gamma_B \cup \{Z_0\}$
$F = F_A \cup F_B$
$\delta : Q \times \Sigma \cup \{\lambda\} \times \Gamma \to \mathcal{P}(Q) \times \Gamma^*)$

$$\delta(q_0, \lambda, Z_0) = \{(q_{0A}, Z_{0A}), (q_{0B}, Z_{0B})\}$$

para todo $q \in Q_A$, $c \in \Sigma \cup \{\lambda\}$, $Z \in \Gamma_A$, para todo $a \in \Sigma$

$$\delta_A(q, a, Z) = \delta(q, a, Z)$$

para todo $q \in Q_B$, $c \in \Sigma \cup \{\lambda\}$, $Z \in \Gamma_B$, para todo $a \in \Sigma$

$$\delta_B(q, a, Z) = \delta(q, a, Z)$$

Por definición

$$\mathcal{L}(C) = \{w \in \Sigma^* : (q_0, w, Z_0) \overset{*}{\underset{C}{\vdash}} (q, \lambda, v), v \in \Gamma^*, q \in F\}$$

Hay dos posibilidades (no determinísticas)
$(q_0, w, Z_0) \underset{C}{\vdash} (q_{0A}, w, Z_{0A})$ y $(q_0, w, Z_0) \underset{C}{\vdash} (q_{0B}, w, Z_{0B})$.
Sabemos que para todo $q \in Q$, $c \in \Sigma \cup \{\lambda\}$, $Z \in \Gamma$,
$\delta(q_{0_A}, c, Z) = \delta_A(q_{0_A}, c, Z)$ y $\delta(q_{0_B}, c, Z) = \delta_A(q_{0_A}, c, Z)$.
Por lo tanto, $w \in \mathcal{L}(C)$ implica

$(q_0, w, Z_0) \underset{C}{\vdash} (q_{0A}, w, Z_{0A}) \overset{*}{\underset{C}{\vdash}} (q, \lambda, v), v \in \Gamma^*, q \in F_A$
o
$(q_0, w, Z_0) \vdash (q_{0A}, w, Z_{0A}) \overset{*}{\underset{C}{\vdash}} (q, \lambda, v), v \in \Gamma^*, q \in F_A$
es decir, $w \in \mathcal{L}(A)$ o $w \in \mathcal{L}(B)$.

Recíprocamente, si $w \in \mathcal{L}(A)$ tenemos
$((q_{0A}, w, Z_{0A}) \overset{*}{\underset{A}{\vdash}} (q, \lambda, v), v \in \Gamma^*, q \in F_A$
y por definición de AP C,
$(q_0, w, Z_0) \underset{C}{\vdash} (q_{0A}, w, Z_{0A})$
y
$(q_{0A}, w, Z_{0A}) \overset{*}{\underset{C}{\vdash}} (q, w, v)$, con $q \in F$ m $v \in \Gamma$
Luego
$(q_0, w, Z_0) \overset{*}{\underset{C}{\vdash}} (q, w, v)$, $q \in F$ m $v \in \Gamma$
Por lo tanto, $w \in \mathcal{L}(C)$.

Demostración del teorema

No están cerrados por :

intersección

Sean los lenguajes libres de contexto
$L_1 = \{a^i b^j c^j\}$ *y* $L_2 = \{a^i b^i c^j\}$ *.*
Notemos que $L_1 \cap L_2 = \{a^i b^i c^i\}$ *no es libre de contexto.*

complemento

Supongamos que el complemento fuera libre de contexto.
Entonces
$L_1 \cap L_2 = \overline{\overline{L_1} \cup \overline{L_2}}$ *sería libre de contexto.*

diferencia

Si lo fuera entonces $\overline{L} = \Sigma^* - L$ *deberia ser libre de contexto*

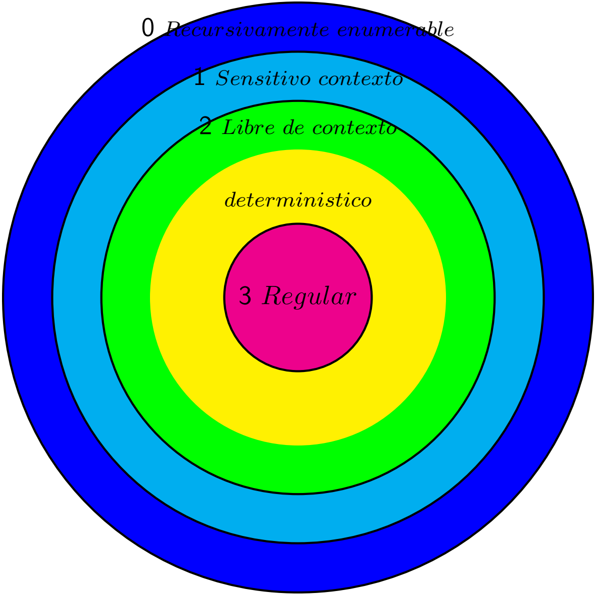
□

La jerarquía de Chomsky (Noam Chomsky en 1956).

Es una clasificación jerárquica de tipos de gramáticas formales que generan lenguajes formales.

Noam Chomsky

La jerarquía de Chomsky de lenguajes



Gramáticas

Definición Una gramática es una 4-upla $G = \langle V_N, V_T, P, S \rangle$ donde

- V_N es un conjunto de símbolos llamados no-terminales (también, variables o categorías sintácticas)
- V_T es un conjunto de símbolos terminales (tal como lo era Σ en los ejemplos anteriores)
- P es el conjunto de "producciones", que es un conjunto finito de

$$\left(V_N \cup V_T\right)^* V_N \left(V_N \cup V_T\right)^* \times \left(V_N \cup V_T\right)^*,$$

  estas producciones son entonces pares ordenados (α, β), que usualmente son notados como $\alpha \rightarrow \beta$.
- $S \in V_N$ es el símbolo distinguido de V_N.

La jerarquía de Chomsky

Gramáticas de tipo 0 (gramáticas sin restricciones)
$\alpha \to \beta$,

Gramáticas de tipo 1 (gramáticas sensibles al contexto)
$\alpha \to \beta$, con $|\alpha| \leq |\beta|$

Gramáticas de tipo 2 (gramáticas libres de contexto)
$A \to \gamma$ con $A \in V_N$.

Gramáticas de tipo 3 (gramáticas regulares).
$A \to a$, $A \to aB$, $A \to \lambda$ con $A, B \in V_N, a \in V_T$.

Si $\alpha\beta\gamma \in (V_N \cup V_T)^*$ y $(\beta \to \delta) \in P$, se dice que $\alpha\delta\gamma$ se deriva directamente en G de $\alpha\beta\gamma$ y se denota como

$$\alpha\beta\gamma \underset{G}{\Rightarrow} \alpha\delta\gamma.$$

Entonces, $\underset{G}{\Rightarrow}$ es una relación sobre $(V_N \cup V_T)^*$, es decir,

$$\underset{G}{\Rightarrow} \subseteq (V_N \cup V_T)^* \times (V_N \cup V_T)^* .$$

Dada la relación $\underset{G}{\Rightarrow}$,

$\overset{k}{\underset{G}{\Rightarrow}}$ es su potencia k

$\overset{+}{\underset{G}{\Rightarrow}}$ es su clausura transitiva

$\overset{*}{\underset{G}{\Rightarrow}}$ es su clausura transitiva y reflexiva

Definición

Lenguaje generado por una gramática $G = \langle V_N, V_T, P, S \rangle$, $\mathcal{L}(G)$,

$$\mathcal{L}(G) = \left\{ \alpha \in V_T^* : S \overset{+}{\underset{G}{\Rightarrow}} \alpha \right\}$$

Gramáticas Libres de Contexto

Una gramática $G = \langle V_N, V_T, P, S \rangle$ es libre de contexto si las producciones en P son de la forma

$$A \to \alpha, \quad \text{con} \quad A \in V_N \quad \text{y} \quad \alpha \in (V_N \cup V_T)^* \,.$$

¿Por qué son tan importantes?
Para cada gramática libre de contexto G hay un autómata de pila M que acepta el lenguaje generado por dicha gramática y viceversa.

Dada una gramática libre de contexto G, se puede decidir si una palabra pertenece a $\mathcal{L}(G)$ en tiempo del orden cúbico de la longitud de la palabra. En casos especiales (determinismo), se puede reconocer en tiempo lineal.

Los lenguajes de programación son lenguajes libres de contexto.

Ejemplo

Sea $G = \langle V_N, V_T, P, S \rangle$ la gramática libre de contexto tal que $V_N = \{E\}$, $V_T = \{+, *, \mathbf{id}, (,)\}$, $S = E$ y P tiene

$$E \to E + E \mid E * E \mid (E) \mid \mathbf{id}$$

En cada paso de la de la derivación debemos elegir qué símbolo no-terminal reescribiremos y luego debemos elegir una producción que tenga ese símbolo del lado izquierdo.

$E \overset{*}{\Rightarrow} (\mathbf{id})$, porque $E \Rightarrow (E) \Rightarrow (\mathbf{id})$

$E \overset{*}{\Rightarrow} (\mathbf{id} + \mathbf{id})$ porque

$E \Rightarrow (E) \Rightarrow (E + E) \Rightarrow (\mathbf{id} + E) \Rightarrow (\mathbf{id} + \mathbf{id})$.

La relación de derivación $\Rightarrow$ determina un árbol cuyas hojas denotan la palabra derivada.

Definición (Árbol de derivación)

Sea $G = \langle V_N, V_T, P, S\rangle$ *una gramática libre de contexto y sea* $\alpha \in V_T^*$*.*
Un árbol de derivación para α *en* G *es tal que*

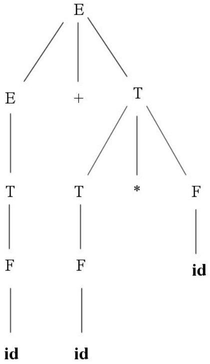
1. *la etiqueta de la raiz es el símbolo distinguido* S*.*
2. *cada vértice posee una etiqueta que pertenece al conjunto* $V_N \cup V_T \cup \{\lambda\}$*.*
3. *si un vértice es interior, su etiqueta debe pertenecer a* V_N*.*
4. *si un vértice* n *posee la etiqueta* A *y sus hijos* $n_1, \ldots, n_k$ *poseen etiquetas* $X_1, \ldots, X_k$ *respectivamente, entonces* $A \to X_1, \ldots, X_k$ *debe ser una producción de* P*.*
5. *si un vértice posee la etiqueta* λ*, entonces es una hoja y es el único hijo de su padre.*

Ejemplo de un árbol de derivación

Sea $G = \langle V_N, V_T, P, S \rangle$ gramática libre de contexto dada por:

$$E \to E + T \mid T$$
$$T \to T * F \mid F$$
$$F \to \mathbf{id} \mid \mathbf{const} \mid (E)$$

Este es un árbol de derivación para $\mathbf{id} + \mathbf{id} * \mathbf{id}$ en G (y es único):

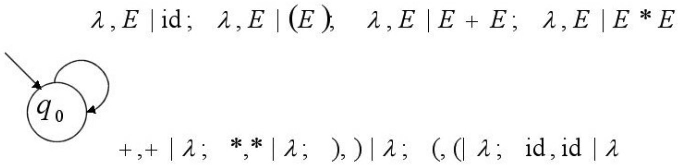


Un atómata de pila para esta gramática libre de contexto

Ejemplo

Sea $G = \langle V_N, V_T, P, S \rangle$ la gramática libre de contexto tal que $V_N = \{E\}$, $V_T = \{+, *, \mathbf{id}, (,)\}$, $S = E$ y P tiene $E \to E + E \mid E * E \mid (E) \mid \mathbf{id}$

Sea $M = \langle Q, \Sigma, \Gamma, \delta, q_0, Z_0, \emptyset \rangle$ con $Q = \{q_0\}$, $\Sigma = V_T$, $\Gamma = V_N \cup V_T$ y $Z_0 = S$.



Si en el tope de la pila hay un símbolo no-terminal, el autómata M lo reemplazará en la pila por el lado derecho de alguna producción.

Si en el tope de la pila hay un símbolo terminal el autómata M constatará que es igual al próximo símbolo en la cadena de entrada y lo desapilará.

Este autómata acepta $\mathcal{L}(G)$ por pila vacía.

Teorema (Chomsky 1962, Evey 1963)

Para cada gramática G libre de contexto existe un autómata de pila M que acepta por pila vacía tal que $\mathcal{L}(G) = \mathcal{L}(M)$.

Teorema (Chomsky 1962, Evey 1963)

Si M es un autómata de pila entonces el lenguaje aceptado por pila vacía, $\mathcal{L}(M)$, es libre de contexto.

No daremos las demostraciones.

¿Cuál es el leguaje generado por esta gramática libre de contexto ?

$G = (V_N, V_T, P_S)$,con $V_N = \{S\}$, $V_T = \{a, b\}$, y P tiene estas 4 producciones:

$S \to aSa \mid bSb \mid a \mid b$