

Tema 5 – Lenguajes independientes del contexto - Objetivos

Objetivos:

- ❑ Conocer los fundamentos de las gramáticas independientes del contexto
- ❑ Saber aplicar las distintas transformaciones en las gramáticas independientes del contexto (gramáticas de tipo 2)
- ❑ Conocer los problemas de decisión en los lenguajes libres de contexto
- ❑ Conocer el lema de bombeo y las propiedades de cerradura para lenguajes libres de contexto
- ❑ Conocer los fundamentos de los autómatas de pila
- ❑ Aplicar la transformación de gramática de tipo 2 a un autómata de pila

Tema 5 - Lenguajes independientes del contexto - Conceptos

Índice:

- ❑ Gramáticas independientes del contexto
- ❑ Transformación de gramáticas independientes del contexto
- ❑ Forma Normal de Chomsky
- ❑ Lema de bombeo para lenguajes independientes del contexto
- ❑ Propiedades de cerradura para los lenguajes independientes del contexto
- ❑ Problemas de decisión para lenguajes independientes del contexto
- ❑ Algoritmo CYK
- ❑ Autómatas de pila
- ❑ Obtención de un autómata de pila a partir de la gramática

Tema 5 Gramáticas independientes del contexto - Definición

Gramáticas independientes del contexto

Las gramáticas independientes del contexto (de tipo 2 o de contexto libre) son aquellas cuyas producciones se ajustan a la siguiente forma:

$$A \rightarrow \alpha \quad \text{en donde } A \in N ; \alpha \in (\Sigma \mid N)^*$$

El término independiente del contexto viene de que se puede cambiar A por α en cualquier contexto donde aparezca A .

La siguiente gramática es independiente del contexto:

$$G = (\Sigma = \{a, b\}, N = \{S, A, B\}, S, P) \quad P: S \rightarrow aABAA, A \rightarrow \varepsilon \mid aA, B \rightarrow \varepsilon \mid bB$$

Es una gramática que genera las palabras del lenguaje::

$$L = \{ a^n b^m a^l \mid n, l \geq 1 \text{ y } m \geq 0 \}$$

Tema 5 Gramáticas independientes del contexto - Derivación

Derivación por la izquierda

Consiste en aplicar un proceso de derivación al no terminal más a la izquierda

$$S \rightarrow aABa \rightarrow aBAa \rightarrow aAa \rightarrow aa$$

Derivación por la derecha

Consiste en aplicar un proceso de derivación al no terminal más a la

$$S \rightarrow aABa \rightarrow aABa \rightarrow aAa \rightarrow aa$$

Árbol de derivación

Es una representación gráfica (en forma de árbol invertido) de un proceso de derivación en una gramática

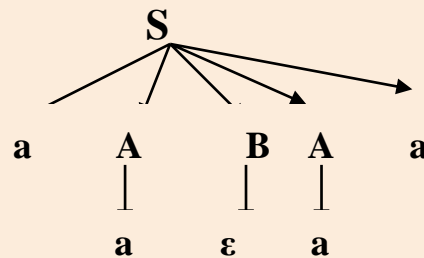
Se define el árbol de derivación como sigue:

- la raíz del árbol será el símbolo inicial de la gramática
- los nodo interiores del árbol están etiquetados por los símbolos no terminales
- las hojas están etiquetadas por símbolos terminales
- si un nodo interior etiquetado por A , posee como hijos los nodos etiquetados por X_1, X_2, \dots, X_n , entonces $A \rightarrow X_1 X_2 \dots X_n$ es una producción de la gramática, en donde X_i representa un símbolo terminal o no terminal

Tema 5 Gramáticas independientes del contexto - Gramáticas ambiguas

Sea la siguiente gramática:

$G = (\Sigma = \{a, b\}, N = \{S, A, B\}, S, P)$ $P: S \rightarrow aABAa; A \rightarrow \varepsilon | aA; B \rightarrow \varepsilon | bB$
árbol de derivación en el proceso de generación de la palabra $x = aaaa$ es el siguiente:



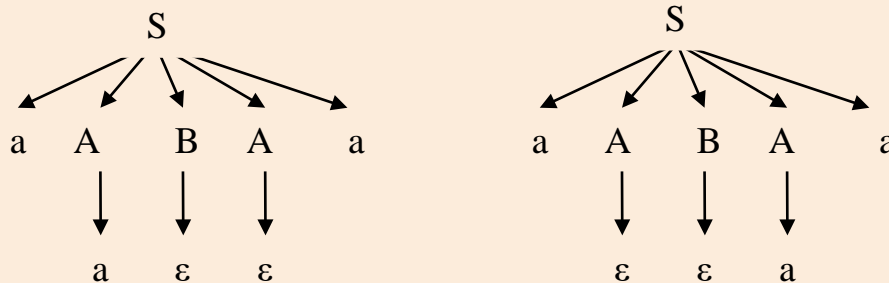
Gramáticas ambiguas

Son aquellas en las que se generan palabras por más de un árbol de derivación

La siguiente gramática es una gramática ambigua, la palabra $x = aaa$, se genera por más de un árbol de derivación:

$G = (\Sigma = \{a, b\}, N = \{S, A, B\}, S, P)$ $P: S \rightarrow aABAa; A \rightarrow \varepsilon | aA; B \rightarrow \varepsilon | bB$

La generación de la palabra $x = aaa$ se puede hacer por más de un árbol de derivación



Tema 5 Gramáticas independientes del contexto - Ambigüedad

La ambigüedad es un proceso indecidible, no existe un algoritmo para detectarlo y corregirlo

Lenguaje inherentemente ambiguo

Es aquel en que todas las gramáticas que lo generan son ambiguas

El siguiente lenguaje es inherentemente ambiguo:

$$L = \{a^n b^m c^l \mid n=m \text{ o } m=l; n, m, l \geq 0\}$$

Una de las posibles gramáticas que lo genera es la que sigue:

$$G = (\Sigma = \{a, b, c\}, N = \{S, A, B, C, D\}, S, P)$$

$$P: S \rightarrow AB \mid CD$$

$$A \rightarrow aAb \mid \varepsilon$$

$$B \rightarrow cB \mid \varepsilon$$

$$C \rightarrow aC \mid \varepsilon$$

$$D \rightarrow bDc \mid \varepsilon$$

Grado de ambigüedad - número de árboles que genera una misma palabra

Tema 5 Gramáticas independientes del contexto - Limpiar una gramática

Limpiar una gramática

Limpiar una gramática consiste en quitar los símbolos inútiles e inaccesibles

Símbolos inútiles

Símbolos no terminales a través de los cuales no se genera ninguna palabra perteneciente al lenguaje.

Son símbolos útiles los siguientes: $A \rightarrow^* w$; $w \in \Sigma^*$; A es útil

Símbolos inútiles = N - Símbolos útiles

Símbolos inaccesibles

Símbolos a los que no se llega a través del símbolo inicial en un proceso de derivación

Son símbolos accesibles: $S \rightarrow^* \alpha a A \beta$; $\alpha, \beta \in (\Sigma|N)^*$; a, A y S son accesibles

Símbolos inaccesibles = $(\Sigma|N)$ - Símbolos accesibles

Tema 5 Gramáticas independientes del contexto - Limpiar una gramática

Proceso para el cálculo símbolos útiles:

- a) El símbolo de la parte izquierda de una derivación directa $A \rightarrow w$; $w \in \Sigma^*$; A es útil
- b) Sí todos los símbolos de la parte derecha de una producción son útiles , el símbolo no terminal de la parte izquierda también lo es $B \rightarrow CcA$; si A y C son útiles, B también lo es
- c) Repetir proceso anterior hasta que se estén añadiendo símbolos útiles

Quitar producciones con símbolos inútiles

Ejemplo: quitar los símbolos inútiles en la siguiente gramática:

$G=(\Sigma=\{a, b, e\}, N=\{S, A, B, C, E\}, S, P)$

P: $S \rightarrow ABaC \mid SB \mid \varepsilon$

$A \rightarrow aAB \mid bB$

$B \rightarrow Ea \mid bB \mid \varepsilon$

$E \rightarrow Ee$

Aplicando el algoritmo, son símbolos útiles= $\{S, B, A\}$,

quitando las producciones con los símbolos inútiles queda la siguiente gramática:

$S \rightarrow SB \mid \varepsilon$ $A \rightarrow aAB \mid bB$ $B \rightarrow bB \mid \varepsilon$

Tema 5 Gramáticas independientes del contexto - Limpiar una gramática

Proceso para el cálculo de símbolos accesibles:

- a) El símbolo inicial es accesible (definido en la gramática) $S \rightarrow w$; S es accesible
- b) Los símbolos que tengan como parte izquierda de la producción el símbolo inicial son accesibles $S \rightarrow AcB$; A, c y B son accesibles
- c) Para los símbolos no terminales que son accesibles y son parte izquierda de una producción, los símbolos de su parte derecha son accesibles
 $A \rightarrow DeF$; si A es accesible D, e y F son accesibles
- d) Repetir proceso anterior hasta que se estén añadiendo símbolos accesibles
- e) Quitar producciones con símbolos inaccesibles

Quitar los símbolos inaccesibles en la siguiente gramática:

$G = (\Sigma = \{a, b\}, N = \{S, A, B\}, S, P)$ P: $S \rightarrow SB | \varepsilon$; $A \rightarrow aAB | bB$; $B \rightarrow bB | \varepsilon$

Aplicando el algoritmo, son símbolos accesibles $\{S, B, b\}$, y quitando los símbolos inaccesibles queda la siguiente gramática:

$G = (\Sigma = \{b\}, N = \{S, B\}, S, P)$

P: $S \rightarrow SB | \varepsilon$,

$B \rightarrow bB | \varepsilon$

se trata de una gramática limpia (blanca)

Tema 5 Gramáticas independientes del contexto – Transformar una gramática

Transformación de gramáticas de tipo 2

Quitar los prefijos comunes y la recursión inmediata por la izquierda

Prefijos comunes

Dos o más producciones tienen prefijos comunes cuando teniendo igual parte izquierda su parte derecha comienza por la misma subpalabra

$$A \rightarrow \delta \alpha_1 \mid \delta \alpha_2 \mid \dots \mid \delta \alpha_n \mid \beta_1 \mid \beta_2 \mid \dots \mid \beta_m; \quad n \geq 2, m \geq 0, \delta \in (N \mid \Sigma)^+, \alpha_i, \beta_j \in (N \mid \Sigma)^*$$

Eliminación de prefijos comunes

$$A \rightarrow \delta A' \mid \beta_1 \mid \beta_2 \mid \dots \mid \beta_m, \quad A' \rightarrow \alpha_1 \mid \alpha_2 \mid \dots \mid \alpha_n$$

Ejemplo de gramáticas que generan el lenguaje $L = \{ b a^n, n \geq 0 \}$

Con prefijos comunes:

$$G_1 = (\Sigma = \{a, b\}, N = \{S, A\}, S, P) P:$$

$$S \rightarrow bA \mid b,$$

$$A \rightarrow aA \mid \varepsilon$$

Sin prefijos comunes

$$G_2 = (\Sigma = \{a, b\}, N = \{S, R, A\}, S, P) P:$$

$$S \rightarrow bR,$$

$$R \rightarrow A \mid \varepsilon,$$

$$A \rightarrow aA \mid \varepsilon$$

Tema 5 Gramáticas independientes del contexto – Transformar una gramática

Recursión por la izquierda

Una o más producciones de una gramática tienen recursión por la izquierda, cuando existen uno o más símbolos de la parte derecha de la producción que coinciden con el símbolo no terminal de la parte izquierda

$$A \rightarrow \alpha_1 A \alpha_2 | \dots | \beta_1 | \beta_2 | \dots | \beta_m \quad n > 0, m > 0 \quad \alpha_i \in (N|\Sigma)^+, \beta_j \in (N|\Sigma)^*$$

Recursión inmediata por la izquierda

Una o más producciones de una gramática tienen recursión inmediata por la izquierda, cuando el primer símbolo de la parte derecha de la producción coincide con el símbolo no terminal de la parte izquierda

$$A \rightarrow A\alpha_1 | A\alpha_2 | \dots | A\alpha_n | \beta_1 | \beta_2 | \dots | \beta_m \quad n > 0, m > 0 \quad \alpha_i \in (N|\Sigma)^+, \beta_j \in (N|\Sigma)^*$$

Eliminación de la recursión inmediata por la izquierda

$$A \rightarrow \beta_1 A' | \beta_2 A' | \dots | \beta_m A', \quad A' \rightarrow \alpha_1 A' | \alpha_2 A' | \dots | \alpha_n A' | \varepsilon$$

Ejemplo de gramáticas que generan el lenguaje $L = \{ b a^n, n \geq 0 \}$

Con recursión inmediata por la izquierda:

$$G_1 = (\Sigma = \{a, b\}, N = \{A\}, A, P) \quad P:$$

$$A \rightarrow Aa | b$$

Sin recursión inmediata por la izquierda:

$$G_2 = (\Sigma = \{a, b\}, N = \{A, A'\}, A, P)$$

$$A \rightarrow bA',$$

$$A' \rightarrow aA' | \varepsilon$$

Tema 5 Gramáticas independientes del contexto – Transformar una gramática

Eliminación de producciones vacías

Los siguientes pasos eliminan las producciones vacías de gramáticas cuya parte izquierda es un símbolo no terminal.

- Determinar los símbolos anulables: un símbolo A es anulable si $A \rightarrow^* \varepsilon$
- Incluir en P' todas las producciones de P menos las producciones vacías
- Si $A \rightarrow \alpha B \beta$ está en P' y B es anulable, entonces añadir a P' la producción $A \rightarrow \alpha \beta$ (excepto si es una producción vacía)
- Repetir el apartado anterior, hasta que no se pueden añadir más producciones.
- Si el símbolo inicial es anulable se añade una producción de la forma $S' \rightarrow \varepsilon | S$

Ejemplo: Sean las producciones correspondiente a una gramática P :

$S \rightarrow aSAb | \varepsilon$

$A \rightarrow aAB | \varepsilon$

$B \rightarrow bB | \varepsilon$

- Símbolos anulables = $\{S, A, B\}$

- P' : $\{ S \rightarrow aSAb, A \rightarrow aAB, B \rightarrow bB \}$

- por ser S, A, B anulables, a P' se la añaden

$S \rightarrow aSb, S \rightarrow aAb, S \rightarrow ab, A \rightarrow aA, A \rightarrow aB, A \rightarrow a,$
 $B \rightarrow Bb, B \rightarrow b$

- $S' \rightarrow \varepsilon | S$

Tema 5 Gramáticas independientes del contexto – Transformar una gramática

Eliminación de producciones unitarias (método de expansión)

Son producciones de la siguiente forma $A \rightarrow B$

Para su eliminación, habrá que realizar los siguientes pasos:

- Eliminar producciones vacías
- Para toda producción unitaria $A \rightarrow B$, con $B \rightarrow \alpha_1 | \alpha_2 | \dots | \alpha_n$; $n > 0$, $\alpha_i \in (N \cup \Sigma)^+$
 - Sustituir $A \rightarrow B$ por $A \rightarrow \alpha_1 | \alpha_2 | \dots | \alpha_n$; $n > 0$, $\alpha_i \in (N \cup \Sigma)^+$
- Repetir el paso anterior, mientras queden producciones unitarias

Ejemplo: Sea la siguiente gramática:

$$G = (\Sigma = \{a, b, c\}, N = \{S, A, B\}, S, P) \Leftrightarrow G = (\Sigma = \{a, b, c\}, N = \{S, A, B\}, S, P'')$$

P: $S \rightarrow SaA | A$
 $A \rightarrow AbB | B$
 $B \rightarrow cB | c$

P': $S \rightarrow SaA | AbB | B$
 $A \rightarrow AbB | B$
 $B \rightarrow cB | c$

P'': $S \rightarrow SaA | AbB | cB | c$
 $A \rightarrow AbB | cB | c$
 $B \rightarrow cB | c$

Tema 5 Gramáticas independientes del contexto – Transformar una gramática

Ejemplo 2: $G = (\Sigma=\{a, b, c\}, N=\{S, A, B\}, S, P)$
 $S \rightarrow A \mid cS \mid \varepsilon$
 $A \rightarrow B \mid aA$
 $B \rightarrow S \mid bB$

El método de expansión trabaja siempre y cuando no haya ciclos en las reglas, por ejemplo en: $S \rightarrow A, A \rightarrow B, B \rightarrow S$.

Para resolverlo, primero calculamos el conjunto de todos los **pares unitarios** de G , $u(G)$ utilizando el siguiente algoritmo:

- *Base*: (A,A) es un par unitario para cualquier variable A . Esto es, $A \Rightarrow^* A$ en cero pasos. $u(G)=\{(A,A): A \in N\}$.
- *Inducción*: Suponemos que $(A,B) \in u(G)$ y que $B \rightarrow C \in P$ donde C es un no terminal. Entonces añadimos (A,C) a $u(G)$.

Calcular $u(G)$ en la gramática del ejemplo 2:

$u(G) = \{(S,S), (S,A), (S,B), (A,A), (A,B), (A,S), (B,B), (B,S), (B,A)\}$

Tema 5 Gramáticas independientes del contexto – Transformar una gramática

Eliminación de producciones unitarias (con recursividad)

Dada $G = (\Sigma, N, S, P)$, podemos construir $G1 = (\Sigma, N, S, P')$:

1. Encontrando el conjunto $u(G)$ de todos los pares unitarios de G .
2. Para cada par unitario (A, B) , añadimos a $P1$ todas las producciones $A \rightarrow \alpha$, donde $B \rightarrow \alpha$ es una producción no unitaria en P .
3. Note que es posible tener $A = B$; de esta manera, $P1$ contiene todas las producciones unitarias en P .
4. $P1 = \{A \rightarrow \alpha : \alpha \in N, B \rightarrow \alpha \in P, (A, B) \in u(G)\}$

$G = (\Sigma = \{a, b, c\}, N = \{S, A, B\}, S, P)$
 $S \rightarrow A \mid cS \mid \varepsilon$
 $A \rightarrow B \mid aA$
 $B \rightarrow S \mid bB$



$G1 = (\Sigma = \{a, b, c\}, N = \{S, A, B\}, S, P1)$
 $S \rightarrow cS \mid aA \mid bB \mid \varepsilon$
 $A \rightarrow aA \mid bB \mid cS \mid \varepsilon$
 $B \rightarrow bB \mid cS \mid \varepsilon \mid aA$

Tema 5 Gramáticas independientes del contexto – Forma Normal de Chomsky

Forma Normal de Chomsky (FNC)

Todo LCL (sin ϵ) se genera por una gramática donde todas las producciones son de la forma: $A \rightarrow BC$ o $A \rightarrow a$, donde A , B y C son no terminales, y a es un terminal.

A esto se le conoce como Forma Normal de Chomsky (FNC), y para llegar a ella debemos:

1. Limpiar la gramática, siguiendo los siguientes pasos (en este orden)
 - Eliminar las producciones- ϵ , es decir, producciones de la forma $A \rightarrow \epsilon$.
 - Eliminar producciones unitarias, es decir, producciones de la forma $A \rightarrow B$
 - Eliminar símbolos inútiles, aquellos que no aparecen en ninguna derivación $S \rightarrow^* w$, para el símbolo de inicio S y la cadena de símbolos terminales w .
2. Hacemos que todos los cuerpos de longitud 2 o más consistan solo de no terminales
3. Dividimos los cuerpos de longitud 3 o más en una cascada de producciones con cuerpos de dos variables

Tema 5 Gramáticas independientes del contexto – Forma Normal de Chomsky

Para el paso 2, por cada terminal a que aparece en una parte derecha de longitud ≥ 2 , creamos un nuevo no terminal, A , y reemplazamos la a por A en todas las partes derechas. Después añadimos una nueva regla $A \rightarrow a$.

Para el paso 3, por cada regla de la forma

$$A \rightarrow B^1 B^2 \dots B^k, k \geq 3,$$

introducimos variables nuevas C^1, C^2, \dots, C^{k-2} , y reemplazamos la regla con:

$$A \rightarrow B^1 C^1$$

$$C^1 \rightarrow B^2 C^2$$

...

$$C^{k-3} \rightarrow B^{k-2} C^{k-2}$$

$$C^{k-2} \rightarrow B^{k-1} B^k$$

Tema 5 Gramáticas independientes del contexto – Forma Normal de Chomsky

Ejemplo

Con la gramática:

$S \rightarrow ASB|\epsilon$

$A \rightarrow aAS|a$

$B \rightarrow SbS|A|bb$

(a) Eliminar producciones ϵ

(b) Eliminar producciones unitarias

(c) Eliminar símbolos inútiles

(d) ponerla en CNF

a) Solo S es anulable, por lo que tenemos que ponerla y quitarla en cada lugar donde ocurre:

$S \rightarrow ASB|AB$

$A \rightarrow aAS|aA|a$

$B \rightarrow SbS|Sb|bS|b|A|bb$

b) La única producción unitaria es: $B \rightarrow A$ por lo que la reemplazamos directamente:

$S \rightarrow ASB|AB$

$A \rightarrow aAS|aA|a$

$B \rightarrow SbS|Sb|bS|b|aAS|aA|a|bb$

Tema 5 Gramáticas independientes del contexto – Forma Normal de Chomsky

c) A y B generan símbolos terminales, y por lo tanto también S, por lo que no hay símbolos inútiles

d) Introducir variables y producciones $C \rightarrow a$ y $D \rightarrow b$ y ponerla en todas las partes derechas que no tienen un solo símbolo terminal

$$S \rightarrow ASB \mid AB$$
$$A \rightarrow CAS \mid CA \mid a$$
$$B \rightarrow SDS \mid SD \mid DS \mid b \mid CAS \mid CA \mid a \mid DD$$
$$C \rightarrow a, D \rightarrow b$$

Para las producciones con más de 3 símbolos se introducen nuevas variables:

$$S \rightarrow AE \mid AB$$
$$A \rightarrow CF \mid CA \mid a$$
$$B \rightarrow SG \mid SD \mid DS \mid b \mid CF \mid CA \mid a \mid DD$$
$$C \rightarrow a, D \rightarrow b$$
$$E \rightarrow SB, F \rightarrow AS$$
$$G \rightarrow DS$$

5.4 Lenguajes independientes del contexto – Lema de bombeo

Lema de bombeo para lenguajes independientes del contexto

Sea L un LIC. Existe una constante n tal que si z es cualquier cadena de L tal que $|z|$ es al menos $|n|$, podemos escribir $z = uvwxy$ tal que:

1. $|vwx| \leq n$
2. $vx \neq \varepsilon$
3. $\forall i \geq 0, uv^iwx^iy \in L$

5.4 Lenguajes independientes del contexto – Lema de bombeo

- Es muy parecido al de lenguajes regulares solo que ahora tenemos que encontrar dos subcadenas que podamos repetir (bombear) indefinidamente.
- La idea de demostración viene de que todo LLC lo podemos representar como FNC y generar árboles de parseo binarios
- Mientras que los LLC pueden servir para dos grupos de caracteres, no sirven para tres (e.g., $\{0^n 1^n 2^n \mid n \geq 1\}$)
- Los LLC tampoco sirven si tenemos pares de símbolos intercalados (e.g., $\{0^i 1^i 2^i 3^j \mid i \geq 1, j \geq 1\}$)
- Tampoco pueden verificar la igualdad de dos cadenas arbitrariamente largas (e.g., $ww \mid w \in \{0, 1\}^*$)

5.4 Lenguajes independientes del contexto – Lema de bombeo

Ejemplos

- $L = \{0^n 1^n 2^n \mid n \geq 1\}$

Idea: $z = uvwxy$, $|vwx| \leq n$ (lema), vwx no puede tener al mismo tiempo 0's y 2's porque el último 0 y el primer 2 están separados por $n + 1$ posiciones. Si vwx no tiene 2's (o 0's) con el bombeo, por ejemplo, de $i = 0$ lo desbalanceamos.

- $L = \{0^i 1^i 2^i 3^j \mid i \geq 1, j \geq 1\}$

Con un argumento similar podemos tener que la cadena no tiene todos los símbolos y desbalancear la cadena.

- $L = \{ww \mid w \in \{0, 1\}^*\}$

En este caso, podemos poner como ejemplo: $0^n 1^n 0^n 1^n$, la cual es una cadena ww y mostrar, tomando las ideas de los ejemplos anteriores, que podemos generar una cadena que no está en L .

5.5 Lenguajes independientes del contexto – Propiedades de cerradura

Propiedades de cerradura de los LLCs

- Existen varias propiedades de los LLC, una de las más importantes es la de sustitución.
- **Sustitución:** Sea Σ un alfabeto y supongamos que para cada símbolo a en Σ definimos un lenguaje arbitrario L_a
- Estos lenguajes definen una función s o una sustitución.
- Si $w = a_1 a_2 \dots a_n$ es una cadena en Σ^* , $s(w)$ es la concatenación de los lenguajes $s(a_1)s(a_2) \dots s(a_n)$.

Ejemplo

$\Sigma = \{0, 1\}$ y $s(0) = \{a^n b^n : n \geq 1\}$, $s(1) = \{aa, bb\}$

Sea $w = 01$. Calcula $s(w)$

- $s(w) = s(0)s(1) = \{a^n b^n aa : n \geq 1\} \cup \{a^n b^{n+2} : n \geq 1\}$.

Si $L = L(0^*)$, calcula $s(L)$

- $s(L) = (s(0))^* = \{a^{n_1} b^{n_1} a^{n_2} b^{n_2} \dots a^{n_k} b^{n_k} : k \geq 0, n_i \geq 1\}$

5.5 Lenguajes independientes del contexto – Propiedades de cerradura

Teoremas

Sea L un LLC sobre Σ , y s una sustitución, tal que $s(a)$ sea un LLC, $\forall a \in \Sigma$. Entonces $s(L)$ es un LLC.

Si tenemos uno o más LLC's, también son LLC el resultado de hacer:

- (i) **Unión:** La unión $L = L_1 \cup L_2$ es un lenguaje de contexto libre, ya que la gramática $G = (\Sigma_i, N_i, S_i, P_i)$ con $\Sigma = \Sigma_1 \cup \Sigma_2$, $N = N_1 \cup N_2$, $P = P_1 \cup P_2 \cup \{S \rightarrow S_1 | S_2\}$, siendo S una nueva variable, que satisface que $L = L(G)$
- (ii) **Concatenación:** La concatenación $L = L_1 L_2$ es un LLC. Se prueba igual que antes, siendo ahora $P = P_1 \cup P_2 \cup \{S \rightarrow S_1 S_2\}$.
- (iii) Cerradura de Kleene: $L = L_1^*$ es un LLC. Basta considerar el conjunto de producciones $P = P_1 \cup \{S \rightarrow SS_1 \mid \epsilon\}$, donde $S \notin N_1$.
- (iv) cerradura positiva +
- (v) Inversión
- (vi) homomorfismo, y homomorfismo inverso.

5.5 Lenguajes independientes del contexto – Propiedades de cerradura

Intersección:

La intersección $L = L_1 \cap L_2$ no es, en general, un LLC. Por ejemplo, los lenguajes $L_1 = \{a^n b^n c^m : n, m \geq 0\}$ y $L_2 = \{a^n b^m c^m : n, m \geq 0\}$ son ambos de contexto libre, pero $L = L_1 \cap L_2 = \{a^n b^n c^n : n \geq 0\}$ no lo es

Complementario:

El complemento $L = L^c$ no es, en general, un LLC. Esto se sigue teniendo en cuenta que $L_1 \cap L_2 = (L_1^c \cup L_2^c)^c$, ya que si el complementario fuera siempre LLC, también lo sería la intersección de cualesquiera dos lenguajes de contexto libre.

Diferencia:

La diferencia $L = L_1 - L_2$ no es en general un LCL. Esto es debido a que $L = L_1 \cap L_2^c$.

Intersección con lenguaje regular

Si R es un lenguaje regular, entonces $L = L_1 \cap R$ es siempre un LLC.

5.5 Lenguajes independientes del contexto – Problemas de decisión

Problemas de decisión para LLC

Problema de infinitud. Dada una gramática G , resolver la pregunta ¿es $L(G)$ infinito?

Ejemplo 1

$$\begin{aligned} S &\rightarrow AB \\ A &\rightarrow BC \mid a \\ B &\rightarrow CC \mid b \\ C &\rightarrow a \end{aligned}$$

El lenguaje generado por la gramática es $\{ab, aaa, bab, baaa, aaab, aaaaa\}$ que es finito

Ejemplo 2

$$\begin{aligned} S &\rightarrow BA \mid BC \\ A &\rightarrow SC \\ B &\rightarrow a \\ C &\rightarrow b \end{aligned}$$

El lenguaje generado por esta gramática es infinito

Decisión: Si el árbol de parseo de la gramática, en forma FNC, tiene un ciclo, el lenguaje es infinito

5.5 Lenguajes independientes del contexto – Problemas de decisión

Lenguaje vacío

Dado un LCL L , ¿es L vacío? Si $L = L(G)$ basta con simplificar la gramática eliminando los símbolos inútiles,. Si el símbolo inicial es inútil, entonces $L = \emptyset$, en otro caso $L \neq \emptyset$

Problemas no-decidibles

Hay problemas que no los podemos resolver con una computadora (no-decidibles)

Por ejemplo:

- ¿Es una GLC dada ambigua?
- Es un LLC dado inheremente ambiguo?
- ¿Es la intersección de dos GLCs vacía?
- ¿Son dos GLCs iguales?
- ¿Es un LLC dado igual a Σ^* donde Σ es el alfabeto de ese lenguaje?

5.5 Lenguajes independientes del contexto – Algoritmo CYK

Probar membresía en un LLC: ¿ $w \in L$?

- ❑ Para probar si una cadena w es parte del lenguaje (L) de una LLC (i.e., $w \in L$), usar el algoritmo CYK (Cocke, Younger, Kasami), que soluciona este problema en $O(|w|^3)$
- ❑ Se construye una matriz triangular inferior de $n \times n$
- ❑ La gramática debe estar en FNC
- ❑ El eje x corresponde a la cadena a probar $w = a_1, a_2, a_3, \dots, a_n$
- ❑ En el primer renglón se ponen producciones tipo $A \rightarrow a$
- ❑ En el resto (e.g., X_{ij}) se ponen los no terminales (e.g., A) tales que tengan una producción (e.g., $A \rightarrow BC$) donde B está en X_{ik} (parte del prefijo) y C está en X_{kj} (resto de la cadena), $i < k < j$.

X_{15}				
X_{14}	X_{25}			
X_{13}	X_{24}	X_{35}		
X_{12}	X_{23}	X_{34}	X_{45}	
X_{11}	X_{22}	X_{33}	X_{44}	X_{55}
a_1	a_2	a_3	a_4	a_5

5.5 Lenguajes independientes del contexto – Algoritmo CYK

Ejemplo

Probar que la siguiente gramática genera: *baaba*

$$S \rightarrow AB|BC$$

$$A \rightarrow BA|a$$

$$B \rightarrow CC|b$$

$$C \rightarrow AB|a$$

$$b : \{B\}$$

$$a : \{A, C\}$$

$$ba : \{B\}\{A, C\} = \{BA, BC\} : \{S, A\}$$

$$aa : \{A, C\}\{A, C\} = \{AA, AC, CA, CC\} : \{B\}$$

$$ab : \{A, C\}\{B\} = \{AB, CB\} : \{S, C\}$$

$$ba : \{B\}\{A, C\} = \{BA, BC\} : \{S, A\}$$

5.5 Lenguajes independientes del contexto – Algoritmo CYK

$$\begin{aligned}bab &: b/ab \text{ o } ba/b : \{B\}\{B\} \cup \{S,A\}\{A,C\} = \{BB,SA,SC,AA,AC\} : \emptyset \\aab &: a/ab \text{ o } aa/b : \{A,C\}\{S,C\} \cup \{B\}\{B\} = \{AS,AC,CS,CC,BB\} : \{B\} \\aba &: a/ba \text{ o } ab/a : \{A,C\}\{S,A\} \cup \{S,C\}\{A,C\} = \{AS,AA,CS,CA,SA,SC,CA,CC\} : \{B\} \\baab &: b/aab \text{ o } ba/ab \text{ o } baa/b : \{B\}\{B\} \cup \{S,A\}\{S,C\} \cup \emptyset\{B\} = \{BB,SS,SC,AS,AC\} : \emptyset \\aaba &: a/aba \text{ o } aa/ba \text{ o } aab/a : \{A,C\}\{B\} \cup \{B\}\{S,A\} \cup \{B\}\{A,C\} \\&= \{AB,AC,BS,BA,BA,BC\} : \{S,A,C\}\end{aligned}$$

5.5 Lenguajes independientes del contexto – Algoritmo CYK

baaba : *b/aaba o ba/aba o baa/ba o baab/a* :

$$\{B\}\{S,A,C\} \cup \{S,A\}\{B\} \cup \emptyset \cup \emptyset = \\ \{BS,BA,BC,SB,AB\} : \{S,A,C\}$$

	{S,A,C}				
-	{S,A,C}				
-	{B}		{B}		
	{S,A}	{B}	{S,C}	{S,A}	
	{B}	{A,C}	{A,C}	{B}	{A,C}
	<i>b</i>	<i>a</i>	<i>a</i>	<i>b</i>	<i>a</i>

$S \rightarrow AB|BC$

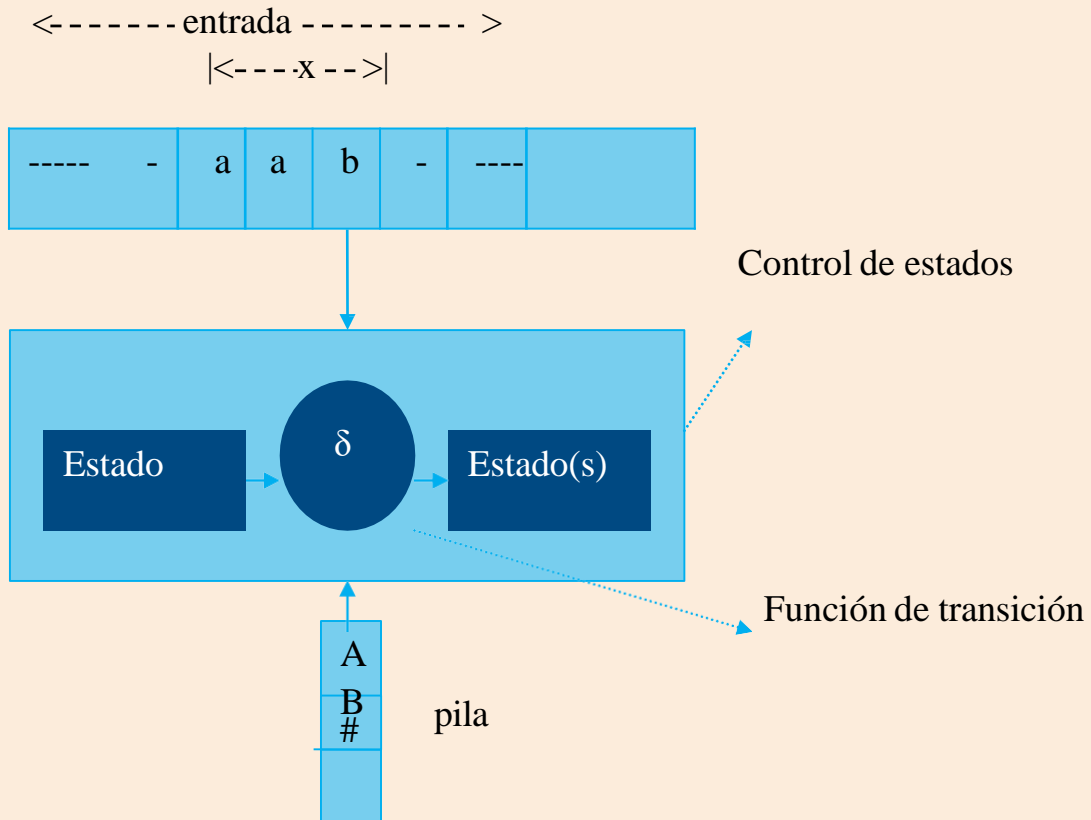
$A \rightarrow BA|a$

$B \rightarrow CC|b$

$C \rightarrow AB|a$

Tema 5 - Autómata de pila

Esquema conceptual



Lenguajes independientes del contexto

Tema 5 Definición formal de un AP

■ Definición formal AP

Un AP se define mediante una septupla $AP=(\Sigma, Q, \Gamma, \delta, q_0, \#, F)$, donde:

- Σ : alfabeto de las palabras a reconocer por el autómata
- Q : conjunto de estados, es un conjunto finito no vacío
- Γ : alfabeto de los símbolos de la pila
- $\delta : Q \times \Sigma \cup \{\epsilon, \$\} \times \Gamma \rightarrow P \times \Gamma^*$, en donde $P \subseteq Q$
función de transición, que transforma la terna ordenada $(q_i, a, A) \rightarrow (q_j, \alpha)$
- $q_0 \in Q$, estado inicial (estado de partida)
- $\#$: símbolo de la base de la pila
- $F \subseteq Q$: conjunto de estados finales o de aceptación

■ Significación de una función de transición $(q_i, a, A) \rightarrow (q_j, BB)$

q_i - estado actual del autómata

a - símbolo leído de la entrada, puede que no se lea de la entrada ϵ

A - elemento que se quita de la pila, puede que no se quite nada de la pila ϵ

q_j - estado destino del autómata

BB - elementos a poner en la pila. Puede que no se ponga nada en la pila ϵ

Tema 5 Construcción de un AP

□ Ejemplo de construcción de un AP

Dado el lenguaje $L = \{a^n b^n \mid n > 0\}$, especificar un AP que lo reconozca:

- Hay que buscar una lógica que simule la entrada a través de la pila:

Apilar las a's mientras que vengan, desapilar las a's cuando lleguen las b's, cuando llegue el final de palabra no habrá elementos en la pila

$AP = (\Sigma = \{a, b\}, Q = \{q_0, q_1, q_2\}, \Gamma = \{A, \#\}, \delta, q_0, \#, F = \{q_2\})$

$\delta(q_0, a, \#) \rightarrow (q_0, A\#)$ - lee a, desapila # y apila A#, no se cambia de estado

$\delta(q_0, a, A) \rightarrow (q_0, AA)$ - lee a, desapila A y apila AA, no se cambia de estado

$\delta(q_0, b, A) \rightarrow (q_1, \varepsilon)$ - lee b, desapila A y no apila nada, se cambia de estado

$\delta(q_1, b, A) \rightarrow (q_1, \varepsilon)$ - lee b, desapila A y no apila nada, no se cambia de estado

$\delta(q_1, \$, \#) \rightarrow (q_2, \varepsilon)$ - lee \$, desapila # y no apila nada, se cambia de estado

La palabra ha finalizado - la pila está vacía y estamos en un estado final

□ Representación gráfica de los AP

Para mejorar la legibilidad y visibilidad de los autómatas se pueden utilizar dos tipos de representaciones gráficas: diagramas o las tablas de transición:

Tema 5 Diagrama de transición de un AP

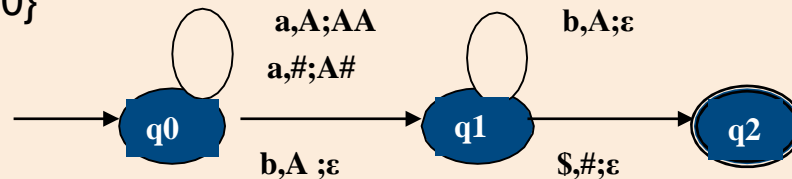
■ Diagramas de transición

La representación de diagrama de transiciones, es un grafo dirigido, en el que:

- Los nodos se representa por un círculo, que se etiquetan con los estados $q \in Q$
- Los arcos son etiquetados con el símbolo de lectura, una coma que lo separa del símbolo que se desapila, un punto y coma que lo separa de los símbolos que se quieren apilar
- así desde el nodo q_i al q_j , si existe la función $\delta(q_i, a, A) \rightarrow (q_j, BB)$. El arco será $a, A; BB$
- El estado inicial tiene un arco entrante no etiquetado \rightarrow
- Los estados finales $q_f \in F$ se señalan por doble círculo, en los AP puede que no existan estados finales

■ Ejemplo de AP representado por un diagrama de transición, que reconoce el lenguaje :

$$L = \{a^n b^n \mid n > 0\}$$



Tema 5 Tabla de transición de un AP

□ Tablas de transición

Representación tabular que relaciona los componentes: entradas, estados y pila

- Cada fila corresponde a un estado $q \in Q$
- Cada columna corresponde símbolo de la entrada $a \in \Sigma$ y símbolo de la cima de pila $A \in \Gamma$
- La fila asociada al estado inicial tiene una \rightarrow
- Las filas asociadas a los estados finales de existir tienen el símbolo *

En la posición $(q, (a, A))$, $q \in Q$, $a \in \Sigma$ y $A \in \Gamma$ está el estado y los elementos a apilar que determine la función $\delta(q, a, A) \rightarrow (p, \alpha)$ en este caso (p, α) $p \in Q$ y $\alpha \in \Gamma^*$

AFD representado por la tabla de transición que reconoce el lenguaje: $L = \{a^n b^m \mid n, m > 0\}$

estados	a, A	a, #	b, A	b, #	\$, #	ϵ , B	A, ϵ	B, ϵ
$\rightarrow q_0$	q_0, AA	$q_0, A\#$	q_1, ϵ					
q_1			q_1, ϵ		q_2, ϵ			
$*q_2$								

Tema 5 Lenguaje reconocido por un AP

Lenguaje reconocido por un autómata de pila

El lenguaje que reconoce un AP será: el conjunto de palabras que partiendo de una situación inicial, se llega a una situación final en cero o más movimientos.

En los AP existen dos formas de reconocimiento de los lenguajes:

- Reconocimiento por llegada a estado final

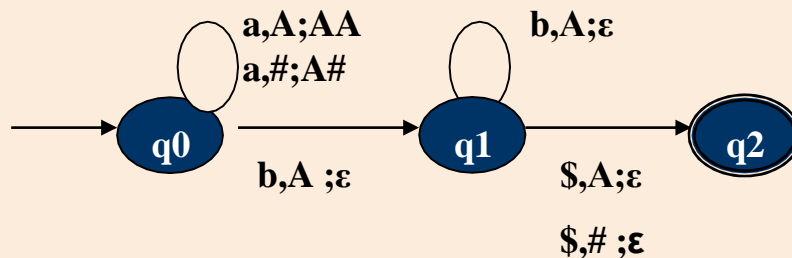
Lenguaje = $\{ x \mid x \in \Sigma^*, (q_i, x, \#) \rightarrow^* (q_f, \varepsilon, \alpha), q_f \in F, \alpha \in \Gamma^* \}$ AP

Ejemplo: $L = \{ a^n b^m \mid n \geq m; n, m > 0 \}$

$AF = (\Sigma = \{a, b\}, Q = \{q_0, q_1, q_2\}, \Gamma = \{A, \#\}, \delta, q_0, \#, F = \{q_2\})$,

$\delta(q_0, a, \#) \rightarrow (q_0, A\#), \delta(q_0, a, A) \rightarrow (q_0, AA), \delta(q_0, b, A) \rightarrow (q_1, \varepsilon)$

$\delta(q_1, b, A) \rightarrow (q_1, \varepsilon), \delta(q_1, \$, a) \rightarrow (q_2, \varepsilon)$



Tema 5 Lenguaje reconocido por AP por vaciado de pila

□ Reconocimiento por llegada vaciado de pila

Lenguaje = $\{ x \mid x \in \Sigma^*, (q_i, x, \#) \rightarrow^* (q_j, \varepsilon, \varepsilon) \}$ AP

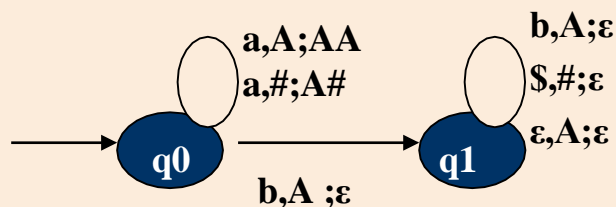
$L = \{ a^n b^m \mid n \geq m; n, m > 0 \}$

$AF = (\Sigma = \{a, b\}, Q = \{q_0, q_1\}, \Gamma = \{A, \#\}, \delta, q_0, \#, F = \{q_1\})$,

$\delta(q_0, a, \#) \rightarrow (q_0, A\#), \delta(q_0, a, A) \rightarrow (q_0, AA), \delta(q_0, b, A) \rightarrow (q_1, \varepsilon)$

$\delta(q_1, b, A) \rightarrow (q_1, \varepsilon), \delta(q_1, \$, \#) \rightarrow (q_1, \varepsilon)$

$(q_0, x = aab\$, \#) \rightarrow (q_0, ab\$, A\#) \rightarrow (q_0, b\$, AA\#) \rightarrow (q_1, \$, A\#)$
 $\rightarrow (q_1, \varepsilon, \#) \rightarrow (q_1, \varepsilon, \varepsilon)$



Tema 5 Autómatas de pila deterministas - APD

Autómatas de pila deterministas APD

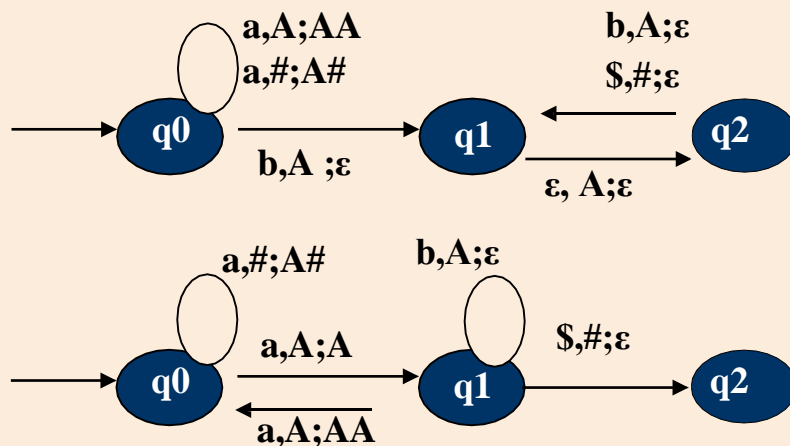
Un autómata de pila es determinista APD cuando para cualquier situación sólo puede darse como mucho un movimiento, para ello debe cumplir las siguientes condiciones:

$$\forall q \in Q, \forall A \in \Gamma \text{ y } a \in \Sigma \quad | (q, a, A) | < 2$$

$$\forall q \in Q \text{ y } \forall A \in \Gamma \quad | (q, \varepsilon, A) | < 0$$

$$\forall q \in Q \text{ y } \forall a \in \Sigma \quad | (q, a, \varepsilon) | < 0$$

Ejemplo: $L = \{a^{2n} b^n \mid n > 0\}$, APD por vaciado de pila que reconozca el anterior



Tema 5 Autómatas de pila no deterministas - APND

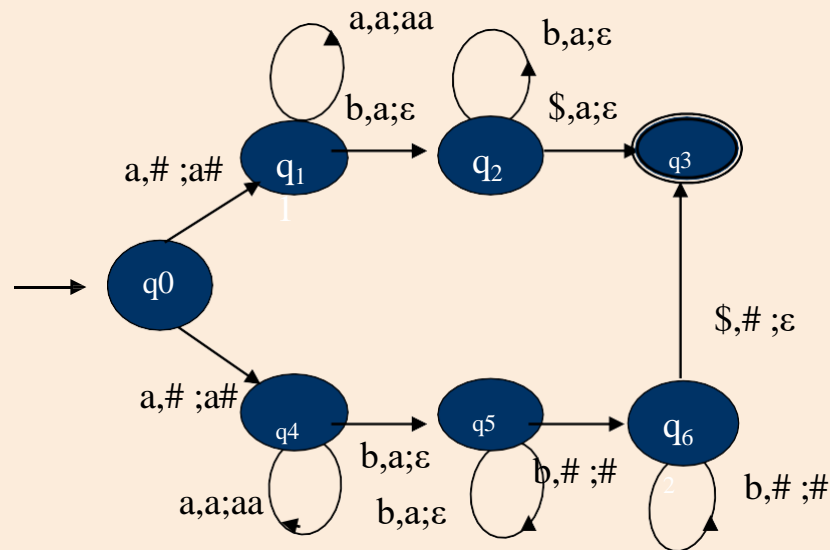
□ Autómatas de pila no deterministas APND

Un autómata de pila es no determinista APD cuando existe alguna situación que puede darse mas de un movimiento.

Se pueden dar tres situaciones de no determinismo que veremos a continuación:

1) $(q_i, a, a) \rightarrow \{(q_j, \alpha_1) (q_k, \alpha_2), \dots\}$

Ejemplo: $L = \{a^n b^m \mid n < m; n, m > 0\}$



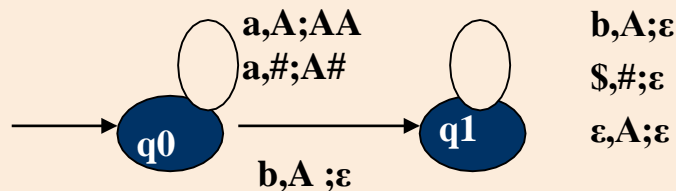
Tema 5 Autómatas de pila no determinista - APND

APND

2) $(q_i, a, a) \rightarrow (\dots)$
 $(q_i, \epsilon, a) \rightarrow (\dots)$

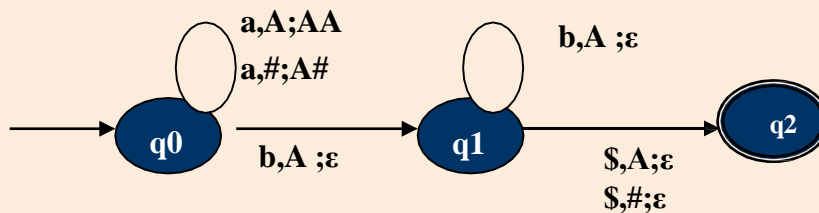
Ejemplo: $L = \{a^n b^m \mid n \geq m; n, m \geq 1\}$

a) reconocimiento por vaciado de pila



$(q_0, aab$, #) \rightarrow (q0, ab$, a#) \rightarrow (q0, b$, aa#) \rightarrow (q1, $, a#) \rightarrow (q1, ε, #) \rightarrow (q1, ε, ε)$
 $(q_0, aaabb$, #) \rightarrow (q0, aabb$, a#) \rightarrow (q0, bb$, aaa#) \rightarrow (q1, b$, aa#) \rightarrow (q1, b$, #)$

b) reconocimiento por llegada a estado final



Lenguajes independientes del contexto

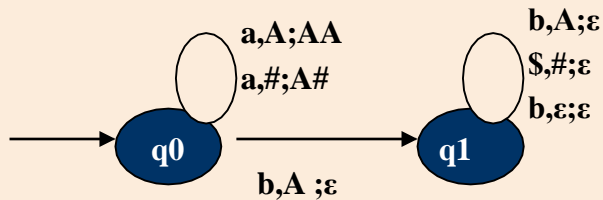
Tema 5 Autómatas de pila no deterministas - APND

3) $(q_i, a, a) \rightarrow (\dots)$

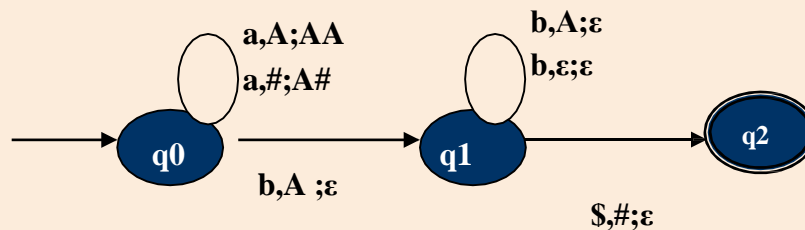
$(q_i, a, \epsilon) \rightarrow (\dots)$

Ejemplo: $L = \{a^n b^m \mid n \leq m, n, m \geq 1\}$

a) reconocimiento por vaciado de pila



b) reconocimiento por llegada a estado final



Tema 5 Gramáticas independientes del contexto transformación GIC->AP

Transformación de gramática independiente del contexto a AP

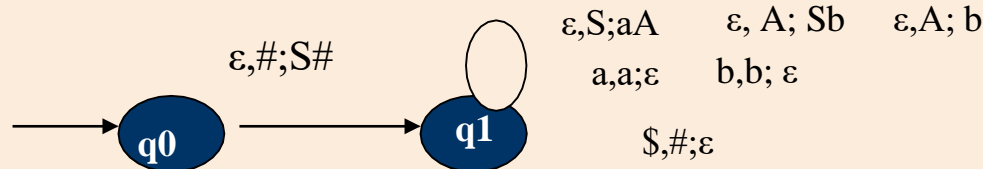
Dada la gramática $G = (\Sigma = \{a, b\}, N = \{S, A\}, S, P)$

$P: S \rightarrow aA \quad A \rightarrow Sb|b$

construir un AP que reconozca el lenguaje que genera la gramática G .

Proceso de transformación:

- 1) Poner el símbolo inicial en la pila $(\epsilon, \#, S\#)$
- 2) Para cada producción $A \rightarrow \alpha$ sustituir parte izquierda por parte derecha $(\epsilon, A; \alpha)$
- 3) Para cada símbolo del alfabeto de la gramática $\forall a \in \Sigma$ poner $(a, a; \epsilon)$
- 4) Poner la transición para que el reconocimiento sea por vaciado de pila $(\$, \#, \epsilon)$



$q_0, aabb\$, \# \rightarrow (q_1, aabb\$, S\#) \rightarrow (q_1, aabb\$, aA\#) \rightarrow (q_1, abb\$, A\#) \rightarrow (q_1, abb\$, Sb\#)$
 $(q_1, abb\$, aAb\#) \rightarrow (q_1, bb\$, Ab\#) \rightarrow (q_1, bb\$, bb\#) \rightarrow (q_1, b\$, b\#) \rightarrow (q_1, \$, \#) \rightarrow (q_1, \epsilon, \epsilon)$

Tema 5 – Ejercicios de gramáticas independientes del contexto

Obtener una gramática y autómatas de pila para cada uno de los siguientes lenguajes:

1. $\Sigma = \{a, b, c\}, L = \{a^n b^m c^n \mid n, m \geq 1\}$
2. $\Sigma = \{a, b, c\}, L = \{a^n b^m c^{n+2m} \mid n \geq 0, m \geq 1\}$
3. $\Sigma = \{a, b, c\}, L = \{a^n b^{n+m} c^m \mid n \geq 1, m \geq 0\}$
4. $\Sigma = \{a, b\}, L = \{a^n b^m \mid n \geq m \geq 1\}$
5. $\Sigma = \{a, b\}, L = \{a^n b^m \mid n > m \geq 0\}$
6. $\Sigma = \{a, b, c\}, L = \{a^m b^n c^k \mid m > n + k ; n, k \geq 0\}$
7. $\Sigma = \{a, b\}, L = \{a^n b^m \mid m \leq n \leq 2m ; n, m \geq 0\}$
8. $\Sigma = \{a, b\}, L = \{a^n b^m \mid n \neq m ; n, m > 0\}$
9. $L = \{v \cdot w \mid v \in (ab)^+, \text{ con } w \in (a \mid b)^+ \text{ y con } |v| \leq |w| \}$
10. $\Sigma = \{a, b, c\}, L = \{a^n b^m c^k \mid n \text{ es un número par, posiblemente el } 0, k \text{ es un número impar y } m > k+n \}$
11. $\Sigma = \{a, b\}, L = \{a^{2n} b^n \mid n > 0\} \cup \{a^n b^{2n} \mid n > 0\}$.
12. $\Sigma = \{a, b, c\}, L = \{x c y \mid x, y \in \{a, b\}^* \text{ en donde } |x| = 2 |y| \}$

Tema 5 - Gramáticas independientes del contexto - Ejercicios

13. Sea el alfabeto $\Sigma = \{a, b\}$ se considera el lenguaje de las palabras que tienen al menos dos símbolos "b" y que cumplen la propiedad de que el número de símbolos "a" que hay antes de la primera "b" es estrictamente mayor que el número de símbolos "a" que hay después de la última "b".

Por ejemplo, pertenecen a este lenguaje: aaabbaa, aabb, abbb, aabababaaba.

No pertenecen: abaabaa, baaba, bab.

14. Dado el alfabeto $\Sigma = \{a, b\}$ sea el lenguaje: $L = \{vbbw \mid v, w \in \{a,b\}^* \text{ y } |v| \geq |w|\}$ (la longitud de v es mayor o igual que la de w)
15. Dado el alfabeto $\Sigma = \{a, b\}$ y, sea el lenguaje $L = \{vw \mid \text{en donde } v, w \in \{a,b\}^* \text{ y } |v| = |w|\}$.
16. Con el alfabeto $\Sigma = \{a,b,c\}$ se define el lenguaje $L = \{w \cdot c^n \mid w \text{ pertenece a } \{a,b\}^+, n > 0\}$ en el que además se cumple que:
- si w termina en a, entonces la longitud de w es mayor o igual que n
 - si w termina en b, entonces la longitud de w es menor o igual que n
- Ejemplos de palabras del lenguaje: abbacc, aaaccc, abccc, bbbabccccc
 - Ejemplos de palabras que no son del lenguaje: abbacccc, aaacccccccc, abc, bbbabccc

Tema 5 - Lenguajes independientes del contexto - Ejercicios

17. Dado el alfabeto $\Sigma = \{a, b, 0, 1\}$, sea el lenguaje $L = \{w \cdot (0|1) \cdot v \mid v, w \text{ pertenecen a } \{a, b\}^+ \text{ y } |w| = |v|\}$, en el que además se cumplen las siguientes restricciones:
- Si el dígito central es 0, w deberá empezar por a
 - Pero si el dígito central es 1, entonces es v quien empezará por a.
 - Ejemplos de palabras válidas: aba0bba, a0a, a0b, a1a, b1a, abb1abb
 - Ejemplos de algunas palabras **no** válidas: b0a, ba0aa, a1b, ba1ba
18. Dado el alfabeto $\Sigma = \{a, b, c\}$, sea el lenguaje $L = \{wc^n \mid w \in \{a, b\}^* ; n = \text{número de subcadenas ab de } w ; n \geq 0\}$
- Es decir, el número de símbolos c del final coincide con el número de subcadenas ab que tiene la palabra al principio, es decir, en w (teniendo en cuenta que en w puede haber cualquier combinación de símbolos a y b)

Tema 5 - Lenguajes independientes del contexto - Ejercicios

17. Convierte la siguiente gramática en Forma Normal de Chomsky (FNC)
 $G = (\Sigma = \{a, b, 0, 1, *, +, (,)\}, N = \{E, T, F, I\}, E, P)$, donde P :
- $$\begin{aligned} E &\rightarrow T \mid E+T \\ T &\rightarrow F \mid T*F \\ F &\rightarrow I \mid (E) \\ I &\rightarrow a \mid b \mid Ia \mid Ib \mid I0 \mid I1 \end{aligned}$$
18. Mostrar que $\{a^i b^j c^k \mid i < j < k\}$ no es LCL
19. Probar que: $\{a^n b^n c^i \mid i \leq n\}$ no es LCL.
20. Probar que: $\{a^{n^2} \mid n \geq 1\}$ no es LCL.

Tema 5 - Lenguajes independientes del contexto - Ejercicios

21. Dada la siguiente gramática libre de contexto en FNC:

$$S \rightarrow AB \mid BC$$

$$A \rightarrow BA \mid a$$

$$B \rightarrow CC \mid b$$

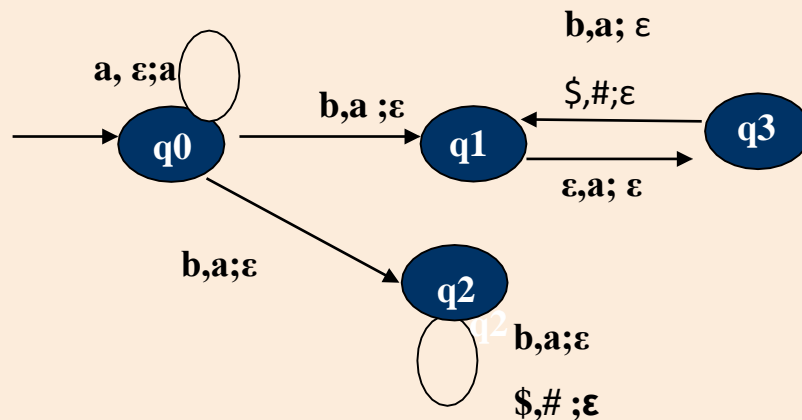
$$C \rightarrow AB \mid a$$

Comprueba si las siguientes palabras pertenecen al lenguaje utilizando el algoritmo CYK

- $w = abb$
- $w = aaba$
- $w = baba$
- $w = abaa$

Tema 5 - Ejercicios de autómatas de pila

1. Justificar si las siguientes transiciones de un AP suponen no determinismo:
 $(q_0, a, b) = (q_2, ab), (q_0, a, \varepsilon) = (q_1, a)$
2. Sea el siguiente autómata a pila con reconocimiento por vaciado de pila:
 $\delta(q_0, a, \#) = (q_0, a\#)$, $\delta(q_0, a, a) = (q_0, aa)$, $\delta(q_0, b, a) = \{ (q_1, \varepsilon), (q_2, \varepsilon) \}$
 $\delta(q_1, b, a) = (q_1, \varepsilon)$, $\delta(q_1, \$, \#) = (q_1, \varepsilon)$
 $\delta(q_2, \varepsilon, a) = (q_3, \varepsilon)$
 $\delta(q_3, b, a) = (q_2, \varepsilon)$, $\delta(q_3, \$, \#) = (q_3, \varepsilon)$
 Determinar el lenguaje reconocido por el mismo.
3. Dado el siguiente AP_1 que realiza el reconocimiento por vaciado de pila:



Se pide: Obtener el lenguaje reconocido por cada uno de ellos

Tema 5 – Ejercicios de autómatas de pila

4. Dado el alfabeto $\Sigma = \{a, b\}$, sea el lenguaje $L = \{a^n b^n \mid n \geq 0\}$.

Encontrar un autómata de pila que reconozca el lenguaje complementario de L.
Considerar que todas las palabras vienen finalizadas por el símbolo \$.

5. Dado el alfabeto $\Sigma = \{a, b\}$, sea el lenguaje $L = \{a^n b^m \mid n > m; n, m > 0\}$. Encontrar un autómata a pila (AP) que reconozca L^* (L estrella).

Se pide:

Sin aplicar ningún algoritmo, es decir, partiendo de la definición del lenguaje, obtener el diagrama de transiciones de un autómata a pila (AP) que lo reconozca.