Universidad Internacional de Valencia (VIU)

56GIIN – Teoría de la Computación

Actividad #2

Alumno: Gagliardo Miguel Angel

Ejercicio #1

Construya gramáticas libres de contexto que generen los siguientes lenguajes:

(a) $\{w \in \{0, 1\}^* \mid w$ contiene al menos tres 1's$\}$

Solucion

G = (V, T, P, S)
V = {S, B}
T = {0, 1}
S = S
P = {
$S \rightarrow B1B1B1B$
$B \rightarrow 0B \mid 1B \mid \varepsilon$
}

(b) $\{w \in \{0, 1\}^* \mid w = w^R$ y $|w|$ es par$\}$

Solucion

G = (V, T, P, S)
V = {S}
T = {0, 1}
S = S
P = $\{ S \rightarrow 0S0 \mid 1S1 \mid \varepsilon \}$

(c) $\{w \in \{0, 1\}^* \mid |w|$ es impar y el símbolo del medio es 0$\}$

Solucion

G = (V, T, P, S)
V = {S}
T = {0, 1}
S = S
P = $\{ S \rightarrow 0S0 \mid 0S1 \mid 1S0 \mid 1S1 \mid 0 \}$

(d) $\{a^i\, b^j\, c^k \mid i, j, k \geq 0 \text{ y } i+j = k\}$

Solucion

Si **k = i + j**, entonces podemos reescribir este lenguaje de la forma:

$\{a^i\, b^j\, c^i\, c^j \mid i, j \geq 0\}$

Con esta representacion podemos verificar que la siguiente gramatica genera el lenguaje:

G = (V, T, P, S)
V = {S, B}
T = {a, b, c}
S = S
P = {
  S → aSc **|** B
  B → bBc **|** ε
}

(e) El lenguaje L de cadenas de corchetes izquierdo y derecho estén correctamente equilibrados: cada corchete izquierdo se puede emparejar con un corchete derecho subsiguiente único, y cada corchete derecho se debe emparejar con un corchete izquierdo anterior único.
Además, la subcadena entre cualquiera de estos pares tiene la misma propiedad. Por ejemplo: [ ] [[[ ] [ ]] [ ]] ∈ L.

Solucion

L (G) = { *w* ∈ { [,] }* **|** *w* es una lista de corchetes equilibrados}

G = (V, T, P, S)
V = {S}
T = {[,]}
S = S
P = { S → ε **|** SS **|** [S] }

Ejercicio #2

(a) Construya un autómata de pila con ∑ = {(,)} que acepte cadenas con paréntesis balanceados por estado final. Por ejemplo, la cadena ((( ))) es aceptada pues al terminar su procesamiento se ha alcanzado un estado final del autómata.

Solucion

Estados del automata: $Q = \{q_0, q_f\}$
Alfabeto del automata: ∑ = {(,)}
Alfabeto de pila: $\Gamma = \{a, \$\}$
Estado inicial: q_0
Estado final: $F = q_f$

Tabla de transiciones:

Entrada	**(**		**)**		**ε**	
Cima	\$	a	\$	a	\$	a
$\mathbf{q_0}$	$(q_0, a\$)$	(q_0, aa)		(q_0, ε)	$(q_f, \$)$	

Ejemplo para ((( ))):

	Simbolo	Operacion de Pila	Transicion
$\delta(q_0, \text{‘(‘}, \$) = \{(q_0, a\$)\}$	(	Colocar a	Para q_0
$\delta(q_0, \text{‘(‘}, a) = \{(q_0, aa)\}$	(	Colocar a	Para q_0
$\delta(q_0, \text{‘(‘}, a) = \{(q_0, aa)\}$	(	Colocar a	Para q_0
$\delta(q_0, \text{‘)‘}, a) = \{(q_0, \varepsilon)\}$	)	Remover a	Para q_0
$\delta(q_0, \text{‘)‘}, a) = \{(q_0, \varepsilon)\}$	)	Remover a	Para q_0
$\delta(q_0, \text{‘)‘}, a) = \{(q_0, \varepsilon)\}$	)	Remover a	Para q_0
$\delta(q_0, \varepsilon, \$) = \{(q_f, \$)\}$			Para q_f

Dado que estamos en el estado final q_f, la cadena es valida.

Ejemplo para ((( ):

	Simbolo	Operacion de Pila	Transicion
$\delta(q_0, \text{‘(‘}, \$) = \{(q_0, a\$)\}$	(	Colocar a	Para q_0
$\delta(q_0, \text{‘(‘}, a) = \{(q_0, aa)\}$	(	Colocar a	Para q_0
$\delta(q_0, \text{‘(‘}, a) = \{(q_0, aa)\}$	(	Colocar a	Para q_0
$\delta(q_0, \text{‘)‘}, a) = \{(q_0, \varepsilon)\}$	)	Remover a	Para q_0

Dado que el ultimo estado donde estamos es q_0, la cadena es rechazada.

(b) Construya un autómata de pila con $\sum = \{(,)\}$ que acepte cadenas con paréntesis balanceados por pila vacía. Por ejemplo, la cadena ((( ))) es aceptada pues al terminar su procesamiento la pila del autómata está vacía.

<u>Solucion</u>

Estados del automata: $Q = \{q_0, q_f\}$
Alfabeto del automata: $\sum = \{(,)\}$
Alfabeto de pila: $\Gamma = \{a, \$\}$
Estado inicial: q_0
Estado final: $F = q_f$

<u>Tabla de transiciones</u>

Entrada	(		)		**ε**	
Cima	\$	a	\$	a	\$	a
q_0	$(q_0, a\$)$	(q_0, aa)		(q_0, ε)	(q_f, ε)	

Ejemplo para ((( ))):

	Simbolo	Operacion de Pila	Transicion
$\delta(q_0, \text{‘(‘}, \$) = \{(q_0, a\$)\}$	‘(‘	Colocar a	Para q_0
$\delta(q_0, \text{‘(‘}, a) = \{(q_0, aa)\}$	‘(‘	Colocar a	Para q_0
$\delta(q_0, \text{‘(‘}, a) = \{(q_0, aa)\}$	‘(‘	Colocar a	Para q_0
$\delta(q_0, \text{‘)‘}, a) = \{(q_0, \varepsilon)\}$	‘)‘	Remover a	Para q_0
$\delta(q_0, \text{‘)‘}, a) = \{(q_0, \varepsilon)\}$	‘)‘	Remover a	Para q_0
$\delta(q_0, \text{‘)‘}, a) = \{(q_0, \varepsilon)\}$	‘)‘	Remover a	Para q_0
$\delta(q_0, \varepsilon, \$) = \{(q_0, \varepsilon)\}$		Remover \$	Para q_f

Dado que el estado de la pila es **vacio**, la cadena es valida.

Ejemplo para ((( ):

	Simbolo	Operacion de Pila	Transicion
$\delta(q_0, \text{‘(‘}, \$) = \{(q_0, a\$)\}$	‘(‘	Colocar a	Para q_0
$\delta(q_0, \text{‘(‘}, a) = \{(q_0, aa)\}$	‘(‘	Colocar a	Para q_0
$\delta(q_0, \text{‘(‘}, a) = \{(q_0, aa)\}$	‘(‘	Colocar a	Para q_0
$\delta(q_0, \text{‘)‘}, a) = \{(q_0, \varepsilon)\}$	‘)‘	Remover a	Para q_0

Dado que el estado de la pila es **aa\$**, esto es, la pila **no** esta vacia, la cadena es invalida.

(c) Construya un autómata de pila que acepte las cadenas/palabras por estado final del siguiente lenguaje: $\{a^i b^i c^j d^j \mid i, j \geq 1\}$

Solucion

Nos estan solicitando un automata de pila que acepte cualquier cadena donde la cantidad de **a's** y **b's** es igual o mayor a 1 e identica, y donde la cantidad de **c's** y **d's** es igual o mayor a 1 e identica, y a la vez que se mantenga el orden (abcd) de las mismas.

Ejemplos validos serian:

- abcd
- aabbcd
- abccdd
- aabbccdd

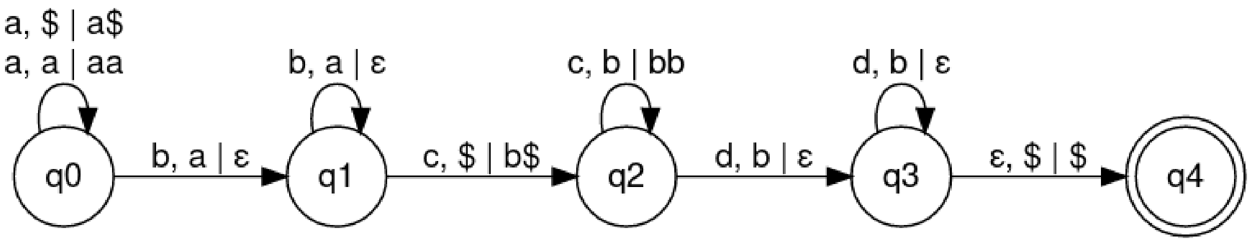
Estados del automata: $Q = \{q_0, q_1, q_2, q_3, q_4\}$
Alfabeto del automata: $\sum = \{a, b, c, d\}$
Alfabeto de pila: $\Gamma = \{a, b, \$\}$
Estado inicial: q_0
Estado final: $F = q_4$

Tabla de transiciones

	a			b			c		
	a	b	$	a	b	$	a	b	$
q_0	(q_0, aa)		$(q_0, a\$)$	(q_1, ε)					
q_1				(q_1, ε)					$(q_2, b\$)$
q_2								(q_2, bb)	
q_3									

	d			ε		
	a	b	$	a	b	$
q_0						
q_1						
q_2		(q_3, ε)				
q_3		(q_3, ε)				$(q_4, \$)$

El automata que corresponde a esta tabla es:



Podemos observar:

- Cada vez que enviamos “a”, apilamos “a” en nuestro stack.

- Cada vez que enviamos “b”, desapilamos “a” de nuestro stack.
    - Primero moviendonos de q_0 a q_1, esto solo es posible si en el stack tenemos “a” dado que la condicion necesaria es a^ib^i con $i \geq 1$.
    - Luego podemos seguir el loop en q_1 para terminar de desapilar todas las a’s de nuestro stack.

- Cada vez que enviamos “c”, apilamos “b” en nuestro stack
    - Condicion necesaria para poder pasar de q_1 a q_2 es que en el stack se encuentre solamente el simbolo “$” dado que la condicion necesaria es que a^ib^i con $i \geq 1$.
    - Tanto en la transicion $q_1 \rightarrow q_2$ como en $q_2 \rightarrow q_2$ cada vez que enviamos “c”, apilamos “b” en nuestro stack.

- Cada vez que enviamos “d”, desapilamos “b” de nuestro stack.
    - Primero moviendonos de q_2 a q_3, esto solo es posible si en el stack tenemos “b” dado que la condicion necesaria es c^jd^j con $j \geq 1$.
    - Luego podemos seguir el loop en q_3 para terminar de desapilar todas las b’s de nuestro stack.

- Por ultimo, solo nos moveremos a la estado final q_4 si la pila esta vacia, en este caso todo el stack fue vaciado exceptuando el simbolo “$”, donde se cumple la condicion $\{a^ib^ic^jd^j \mid i, j \geq 1\}$

(d) Construya un autómata de pila que acepte las cadenas/palabras por pila vacía del siguiente lenguaje:

$\{ w \mid w \in \{a, b\}^*, |w|_a = |w|_b \}$

Solucion

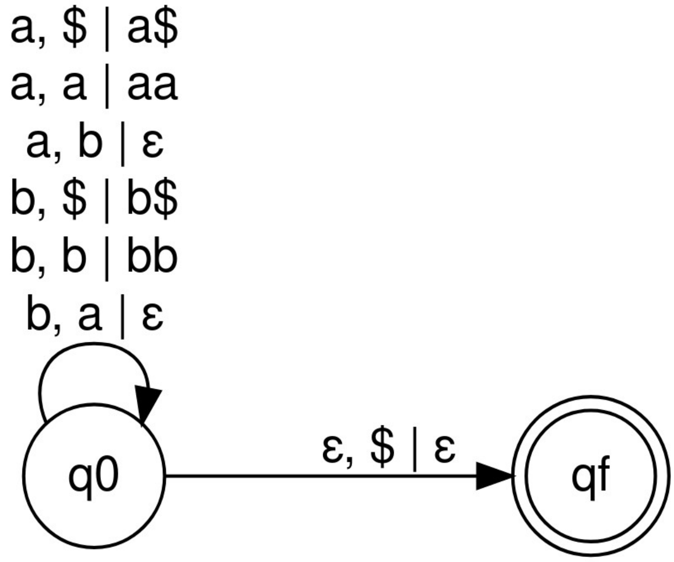
Este automata de pila debe aceptar un lenguaje que tiene igual numero de a's que de b's. Por tanto:

Estados del automata: $Q = \{q_0, q_f\}$
Alfabeto del automata: $\sum = \{a, b\}$
Alfabeto de pila: $\Gamma = \{a, b, \$\}$
Estado inicial: q_0
Estado final: $F = q_f$

Ejemplos:
- ab
- aabb
- abba
- aababb
- etc..

Tabla de transiciones

Entrada	a			b			**ε**		
Cima	\$	a	b	\$	a	b	\$	a	b
q_0	$(q_0, a\$)$	(q_0, aa)	(q_0, ε)	$(q_0, b\$)$	(q_0, ε)	(q_0, bb)	(q_f, ε)		



Ejemplo para **aabb**:

	Simbolo	Operacion de Pila	Transicion
$\delta(q_0, a, \$) = \{(q_0, a\$)\}$	a	Colocar a	Para q_0
$\delta(q_0, a, a) = \{(q_0, aa)\}$	a	Colocar a	Para q_0
$\delta(q_0, b, a) = \{(q_0, \varepsilon)\}$	b	Remover a	Para q_0
$\delta(q_0, b, a) = \{(q_0, \varepsilon)\}$	b	Remover a	Para q_0
$\delta(q_0, \varepsilon, \$) = \{(q_0, \varepsilon)\}$		Remover $	Para q_f

Dado que el estado de la pila es **vacio**, la cadena es valida.

Ejemplo para **abba**:

	Simbolo	Operacion de Pila	Transicion
$\delta(q_0, a, \$) = \{(q_0, a\$)\}$	a	Colocar a	Para q_0
$\delta(q_0, b, a) = \{(q_0, \varepsilon)\}$	b	Remover a	Para q_0
$\delta(q_0, b, \$) = \{(q_0, b\$)\}$	b	Colocar b	Para q_0
$\delta(q_0, a, b) = \{(q_0, \varepsilon)\}$	a	Remover b	Para q_0
$\delta(q_0, \varepsilon, \$) = \{(q_0, \varepsilon)\}$		Remover $	Para q_f

Dado que el estado de la pila es **vacio**, la cadena es valida.

Ejemplo para **aba**:

	Simbolo	Operacion de Pila	Transicion
$\delta(q_0, a, \$) = \{(q_0, a\$)\}$	a	Colocar a	Para q_0
$\delta(q_0, b, a) = \{(q_0, \varepsilon)\}$	b	Remover a	Para q_0
$\delta(q_0, a, \$) = \{(q_0, a\$)\}$	a	Colocar a	Para q_0

Dado que el estado de la pila es **‘a$’**, esto es, la pila **no** esta vacia, la cadena es invalida.