Teoría de Lenguajes

Soluciones

1er. Parcial – Curso 2025

Ejercicio 1.-

a) Dado un AFD M:(Q,Σ,δ,q_0,F) y dos tiras x,y ϵ Σ^* se dice que

x R_M y <=> $\hat{\delta}(q_0,x) = \hat{\delta}(q_0,y)$

δ	a	b	ε
q_0	$\{q_2\}$	{}	$\{q_1, q_3\}$
q_1	$\{q_3\}$	$\{q_2\}$	-
q_2	{}	$\{q_4\}$	-
q_3	$\{q_1\}$	$\{q_4\}$	-
q_4	{}	$\{q_4\}$	$\{q_1\}$

b) Para obtener un AFD, vamos a aplicar en cascada los algoritmos de equivalencia entre AFD's y AFND's.

Primero hacemos el pasaje AFND-eps a AFND con el algoritmo visto en el curso, por lo tanto calculamos las ε-clausuras de cada estado:

$\varepsilon_{cl}(q_0)=\{q_0,q_1,q_3\}$
$\varepsilon_{cl}(q_1)=\{q_1\}$
$\varepsilon_{cl}(q_2)=\{q_2\}$
$\varepsilon_{cl}(q_3)=\{q_3\}$
$\varepsilon_{cl}(q_4)=\{q_4,q_1\}$

Luego calculamos $\varepsilon_{cl}(\delta\ (\varepsilon_{cl}(q_i)))$ para todos los estados

$\varepsilon_{cl}(\delta\ (\varepsilon_{cl}(q_0)))=\varepsilon_{cl}(\delta\ (\{q_0,q_1,q_3\},a))=\varepsilon_{cl}(\{q_1,q_2,q_3\})=\{q_1,q_2,q_3\}$
$\varepsilon_{cl}(\delta\ (\varepsilon_{cl}(q_0)))=\varepsilon_{cl}(\delta\ (\{q_0,q_1,q_3\},b))=\varepsilon_{cl}(\{q_2,q_4\})=\{q_1,q_2,q_4\}$

$\varepsilon_{cl}(\delta\ (\varepsilon_{cl}(q_1)))=\varepsilon_{cl}(\delta\ (\{q_1\},a))=\varepsilon_{cl}(\{q_3\})=\{q_3\}$
$\varepsilon_{cl}(\delta\ (\varepsilon_{cl}(q_1)))=\varepsilon_{cl}(\delta\ (\{q_1\},b))=\varepsilon_{cl}(\{q_2\})=\{q_2\}$

$\varepsilon_{cl}(\delta\ (\varepsilon_{cl}(q_2)))=\varepsilon_{cl}(\delta\ (\{q_2\},a))=\varepsilon_{cl}(\{\})=\{\}$
$\varepsilon_{cl}(\delta\ (\varepsilon_{cl}(q_2)))=\varepsilon_{cl}(\delta\ (\{q_2\},b))=\varepsilon_{cl}(\{q_4\})=\{q_1,q_4\}$

$\varepsilon_{cl}(\delta\ (\varepsilon_{cl}(q_3)))=\varepsilon_{cl}(\delta\ (\{q_3\},a))=\varepsilon_{cl}(\{q_1\})=\{q_1\}$
$\varepsilon_{cl}(\delta\ (\varepsilon_{cl}(q_3)))=\varepsilon_{cl}(\delta\ (\{q_3\},b))=\varepsilon_{cl}(\{q_4\})=\{q_1,q_4\}$

$\varepsilon_{cl}(\delta\ (\varepsilon_{cl}(q_4)))=\varepsilon_{cl}(\delta\ (\{q_4,q_1\},a))=\varepsilon_{cl}(\{q_3\})=\{q_3\}$
$\varepsilon_{cl}(\delta\ (\varepsilon_{cl}(q_4)))=\varepsilon_{cl}(\delta\ (\{q_4,q_1\},b))=\varepsilon_{cl}(\{q_2,q_4\})=\{q_1,q_2,q_4\}$

Con eso construimos la tabla de transiciones del AFND

δ'	a	b
q_0	$\{q_1,q_2,q_3\}$	$\{q_1,q_2,q_4\}$
q_1	$\{q_3\}$	$\{q_2\}$
q_2	{}	$\{q_1,q_4\}$
q_3	$\{q_1\}$	$\{q_1,q_4\}$
q_4	$\{q_3\}$	$\{q_1,q_2,q_4\}$

Pero en este caso al conjunto de estados finales se le agrega q_0 porque $q_0 \notin F$ y $\varepsilon_{cl}(q_0)\cap F\neq\emptyset$. Por lo tanto $F=\{q_0,q_3\}$.

Ahora aplicaremos el algoritmo de pasaje de AFND a AFD, llamaremos $[q_{123}]$ a lo que normalmente llamaríamos $[q_1 q_2 q_3]$ para que la notación no quede muy engorrosa.

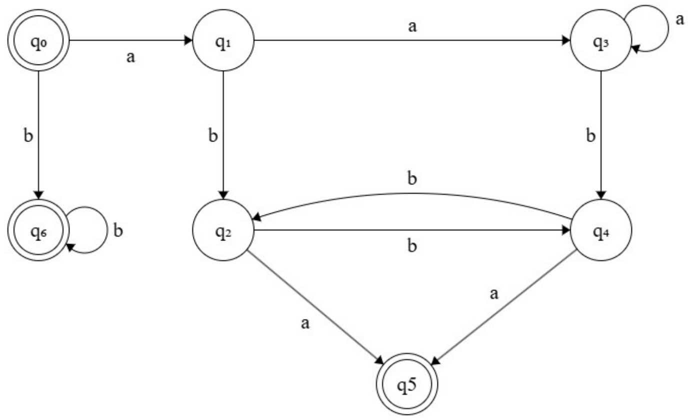
δ''	a	b
$[q_0]$	$[q_{123}]$	$[q_{124}]$
$[q_{123}]$	$[q_{13}]$	$[q_{124}]$
$[q_{124}]$	$[q_3]$	$[q_{124}]$
$[q_{13}]$	$[q_{13}]$	$[q_{124}]$
$[q_3]$	$[q_1]$	$[q_{14}]$
$[q_1]$	$[q_3]$	$[q_2]$
$[q_{14}]$	$[q_3]$	$[q_{124}]$
$[q_2]$	$[q_p]$	$[q_{14}]$
$[q_p]$	$[q_p]$	$[q_p]$

Siendo el estado inicial $[q_0]$ y $F=\{[q_0],[q_{123}],[q_{13}],[q_3]\}$

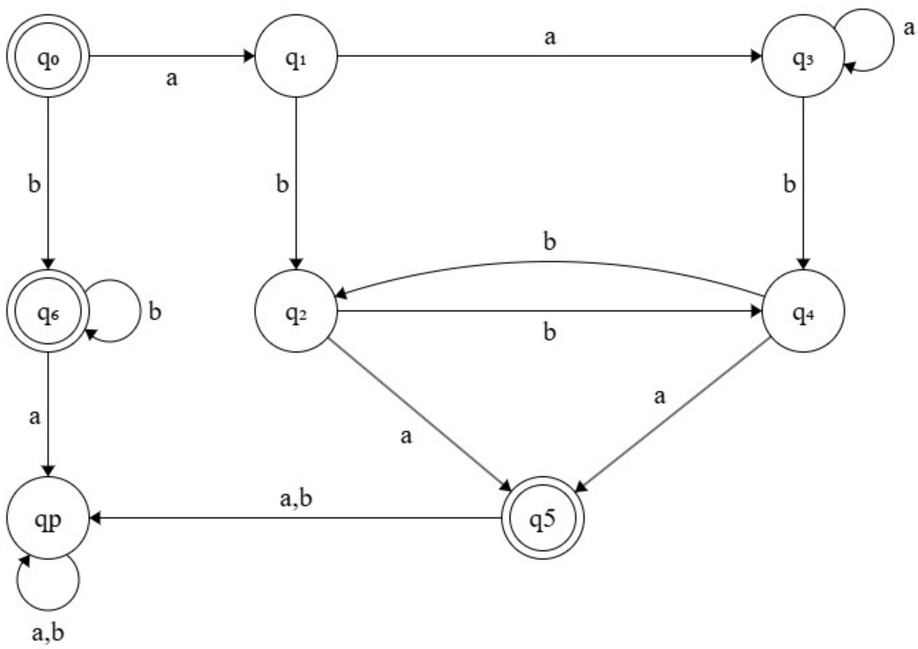
c) Como este autómata tiene todas sus transiciones definidas (es completo) y la cantidad de estados es 9 entonces la cantidad de clases de equivalencia de R_M son 9.

Ejercicio 2.-

a) Como lo que tenemos es un AFD, lo primero que se va a hacer es a partir de AFD dado, obtener su AFDMinimo, así luego, a partir del obtenido, responder la pregunta.



Lo completamos agregándole el pozo:

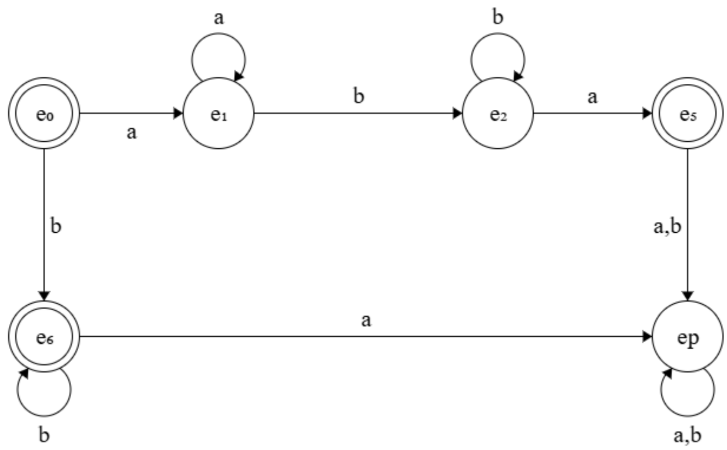


Ahora lo minimizamos

Π_0 [q0 q5 q6] [q1 q2 q3 q4 qp]
Π_1 [q0 q6] [q5] [q2 q4] [q1 q3 qp]
Π_2 [q0 q6] [q5] [q2 q4] [q1 q3] [qp]
Π_3 [q0] [q6] [q5] [q2 q4] [q1 q3] [qp]
Π_4 [q0] [q6] [q5] [q2 q4] [q1 q3] [qp]

Renombramos los estados y dibujamos el autómata:

e0 = [q0]
e1 = [q1q3]
e2 = [q2q4]
e5 = [q5]
e6 = [q6]
ep = [qp]



Como el autómata es mínimo, por el corolario del teorema de Myhill-Nerode sabemos que las clases de R_M van a coincidir con las clases de R_L. Por lo tanto, planteamos ahora el sistema de ecuaciones de R_M, para hallar las expresiones regulares que describen cada una de sus clases y que, a su vez, describen cada una de las clases de R_L.

$X_0 = \varepsilon$
$X_1 = X_1a \mid X_0a$
$X_2 = X_2b \mid X_1b$
$X_5 = X_2a$
$X_6 = X_0b \mid X_6b$
$X_p = X_6a \mid X_5a \mid X_5b \mid X_pa \mid X_5b = X_6a \mid X_5(a|b) \mid X_p(a|b)$

Sustituimos X_0 en X_1:
$X_1 = X_1a \mid \varepsilon.a = X_1a \mid a$
Usando el Lema de Arden: $X_1 = a.a^*$

Sustituimos X_1 en X_2:
$X_2 = X_2b \mid (a.a^*)b = X_2b \mid aa^*b$
Usando el Lema de Arden: $X_2 = aa^*b.b^*$

Sustituimos X_2 en X_5:
$X_5 = (aa^*bb^*)a = aa^*bb^*a$

Sustituimos X_0 en X_6:
$X_6 = \varepsilon.b \mid X_6b = b \mid X_6b$
Usando Lema de Arden: $X_6 = bb^*$

Sustituimos X_6 y X_5 en X_p:
X_p = (bb*)a | (aa*bb*a). (a|b) | X_p (a|b) = bb*a | aa*bb*aa | aa*bb*ab | X_p (a|b)

Usando el Lema de Arden:
X_p = (bb*a | aa*bb*aa | aa*bb*ab).(a|b)*

Entonces, las expresiones regulares para cada clase de R_M y, por lo tanto, de R_L son:

X_0 = ε
X_1 = a.a*
X_2 = aa*b.b*
X_5 = aa*bb*a
X_6 = bb*
X_p = (bb*a | aa*bb*aa | aa*bb*ab).(a|b)*

b) Una expresión regular r que genere el lenguaje aceptado por el autómata M, puede ser la obtenida por la unión de las expresiones regulares asociadas a estados finales. En este caso:

r = X_0 | X_5 | X_6 = ε | aa*bb*a | bb*

Ejercicio 3.-

$L_3 = \{ a^k b^r a^p b^t$; con k, p, t ≥ 0, r > 0 , p > 2*t }

a) El lenguaje **no es regular**. Para demostrarlo, utilizamos el contrarrecíproco del Pumping Lema.
Sea N la constante del PL, y se considera **z= $ba^{2N+1}b^N$**; donde para elegirla se tomo: k=0, r=1, t=N y p= 2N+1. Notar que cumple |z|>N (de hecho |z| = 3N+2)

Consideramos todas las descomposiciones posibles para z = uvw que cumplan:
|uv| ≤ N y |v|≥1

i) u=ε $0 \le j < N$
v=ba^j
w= $a^{2N+1-j} b^N$

$z_i = (ba^j)^i a^{2N+1-j} b^N$
En este caso, para i=0 quedaría:
$z_0 = a^{2N+1-j} b^N$ donde $z_0 \notin L_3$ ya que por un lado no tiene al menos una **b** antes de la segunda secuencia de **a**'s, por otro lado, tampoco cumple la relación de que la cantidad precisamente de esas **a**'s sea mayor que el doble de la última secuencia de **b**'s

ii) u=ba^j $1+j+p \le N$
v= a^p $p \ge 1$
w= $a^{2N+1-j-p} b^N$

$z_i = ba^j(a^p)^i a^{2N+1-j+(i-1)p} b^N = ba^{2N+1+(i-1)p} b^N$ Nuevamente en este caso tomamos i=0

$z_0 = ba^{2N+1-p}\ b^N$ donde z_0 en este caso no se cumple la relación de que la cantidad de **a**'s (del segundo bloque) sea mayor que el doble de la última secuencia de **b**'s ya que $p \geq 1$. Por lo tanto $z_0 \notin L_3$

Estas son las únicas familias a analizar bajo los supuestos $|uv| \leq N$ y $|v| \geq 1$
Con lo cual, como esas son todas las descomposiciones posibles, por el CR del PL L_3 no es un lenguaje regular.

b) La afirmación es **falsa**, ya que que como L_3 **no es regular**, por el Teorema de Myhill-Nerode tiene infinitas clases de equivalencia; con lo cual NUNCA puede tener 2 clases.

c) Sea el homomorfismo $h: \{a,b\} \rightarrow \{0\}^*$ / $h(a) = 00$ y $h(b) = \varepsilon$
Lo aplicamos al L_3 de donde $h(a^k b^r a^p b^t)$ quedaría teniendo en cuenta que $h(b) = \varepsilon$

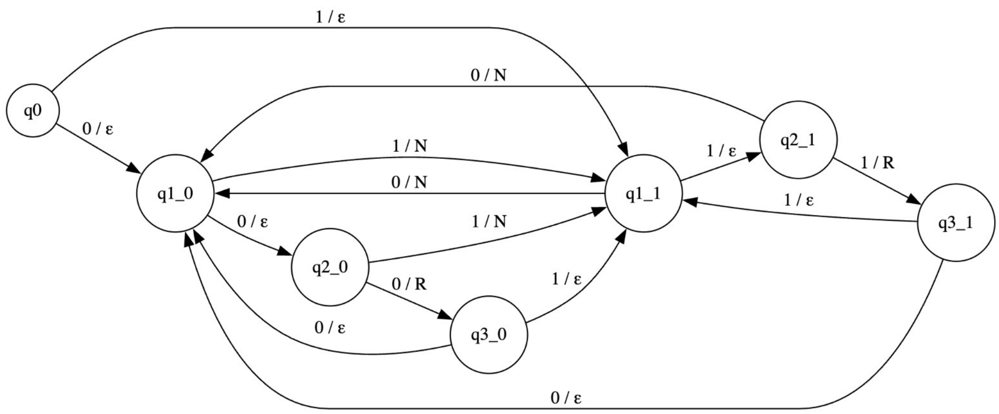
Observar que la tira "más corta" del lenguaje original es a^0bab^0 = ba porque
$p > 2*t \geq 0$, con lo cual en realidad $p>0$
Con esto, si se aplica el homomorfismo, esa tira sería $h(ba) = h(b).h(a) = 00$

Por lo expuesto entonces:
$(00)^*$ - por las primeras a^k, ya que $k \geq 0$
$00(00)^*$ - por las segundas a's, ya que $p>2*t$ con lo cual en realidad $p>0$

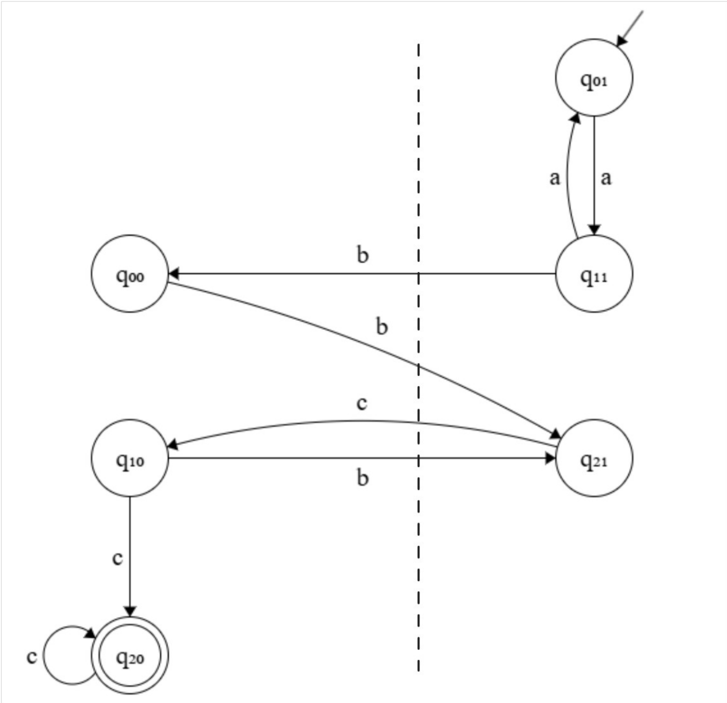
De donde $h(a^k b^r a^p b^t) = (00)^*00(00)^* =$ $\mathbf{00(00)^*}$ y que por existir una ER que lo define, el lenguaje $h(L_3)$ es regular

Ejercicio 4.-

a)



b) $L_{4b} = \{ \langle b^p c^k, a^t b c^p \rangle$ con $p, k > 0$; t MOD $2 = 1$ }



Ejercicio 5.- [Evaluación individual del obligatorio]

a)El objetivo principal de validar una expresión regular antes de construir el autómata es garantizar que la entrada sea sintácticamente correcta y significativa, evitando errores durante la construcción del autómata.
Si una expresión no es válida (por ejemplo, si tiene operadores mal ubicados, símbolos no permitidos o estructuras incompletas), el proceso de traducción a autómata podría fallar, producir resultados incorrectos o generar autómatas inconsistentes.

b)

```
afnd = AFND_e()
ini, fin = nuevo_estado(), nuevo_estado()
afnd.establecer_inicial(ini)
afnd.agregar_final(fin)
afnd.agregar_transicion(ini, 'e', af1.inicial)
afnd.agregar_transicion(ini, 'e', af2.inicial)
for estado in af1.finales:
   afnd.agregar_transicion(estado, 'e', fin)
for estado in af2.finales:
   afnd.agregar_transicion(estado, 'e', fin)
      copiar(afnd, af1)
      copiar(afnd, af2)
```

Nota: Se valora positivamente la simplicidad de las soluciones propuestas así como una breve explicación de éstas. Todas las respuestas deben estar debidamente justificadas.