Teoría de Lenguajes
Soluciones

Ejercicio 1

Indique si las siguientes afirmaciones son verdaderas o falsas. Justifique adecuadamente cada respuesta.

a) Si un lenguaje L_a tiene 4 clases según la relación R_L – con la definición de R_L vista en el curso – L_a no necesariamente es regular.

Falso. Por el Teorema de Myhill-Nerode, un lenguaje es regular si y solo si tiene una cantidad finita de clase de R_L. Por lo tanto La es regular al tener 4 clases.

b) En las propiedades de clausura de los lenguajes regulares está probado que son cerrados bajo la operación de unión. Por tanto se puede afirmar que la unión infinita de lenguajes regulares es regular.

Falso. Basta considerar los lenguajes unitarios de la forma $\{a^k b^k\}$ con k natural. Cada uno de ellos es regular (k fijo) al ser finitos. Sin embargo, su unión es el lenguaje {x/ x es de la forma $a^n b^n$ con k≥0}, que se vio en teórico que no es regular.

c) Sea L_c un lenguaje regular. Sea L_c' el lenguaje resultado de eliminar un subconjunto finito de tiras del lenguaje L_c. Entonces L_c' puede no ser regular.

Falso. Llamemos L' a ese lenguaje formado por el subconjunto finito de tiras de L_c. Al ser finito, es regular.
$L_c' = L_c - L' = L_c \cap (L')^c$ y se tiene que L_c es regular y como L' es regular, su complemento también es regular (propiedad de clausura). Además, la intersección de regulares es regular (propiedad de clausura). Por tanto, L_c' es regular.

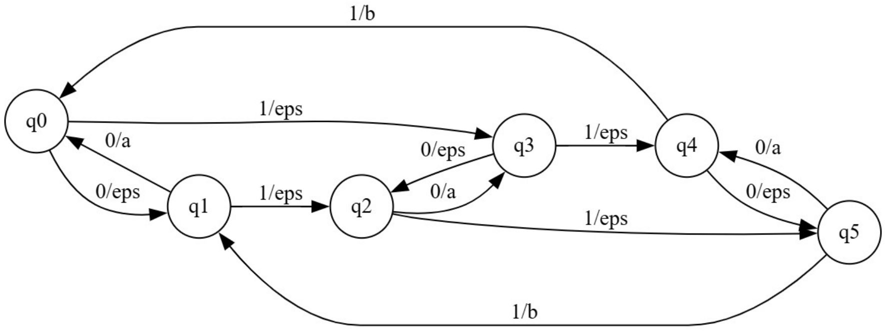
d) En las propiedades de clausura de los lenguajes regulares está probado que son cerrados bajo la operación de homomorfismo. Sea ***h*** un homomorfismo y L_d un lenguaje cualquiera. Entonces si ***h*** (L_d) es regular, L_d es regular.

Falso. Sea $L_d = \{0^k 1^k , k>0\}$ que se sabe no es regular. Consideremos el homomorfismo: ***h***(0) = a, ***h***(1) = a.
Se tiene que ***h***(L_d)= {w / w es de la forma a^{2k}}, que son las tiras con cantidad par de a's, que es regular (existe la ER (aa)* que la define).

Nota: Las gramáticas y los autómatas deben corresponderse con el tipo del lenguaje considerado en cada caso, según la Jerarquía de Chomsky. Se valora positivamente la simplicidad de las soluciones propuestas así como una breve explicación de éstas. Todas las respuestas deben estar debidamente justificadas.

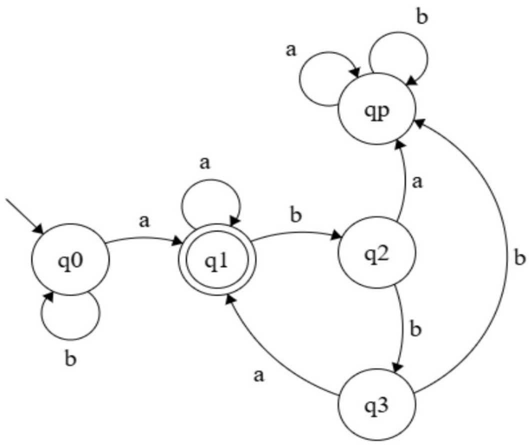
Ejercicio 2

a)



b) Para construir un AFD mínimo que reconozca el lenguaje definido por la expresión regular, primero debo construir un AFD que reconozca dicha ER y luego verificar que este sea mínimo aplicando el algoritmo de minimización para AFD visto en el curso.

El autómata propuesto:



El autómata es determinista ya que en cualquier estado, al leer un carácter de la tira queda determinado el estado al que se mueve. El estado inicial es q0, en el cual luego de haber leído cualquier cantidad de b's, si leo una a paso a q1, estado final que reconoce (a | bba)* con ayuda de q2 y q3.

Para aplicar el algoritmo de minimización de AFD hay que hacer que la función de transición sea completa, por lo tanto agregamos el estado pozo qp.

Se aplica ahora el algoritmo:

La tabla de transiciones es la siguiente:

	a	b
q0	q1	q0
q1	q1	q2
q2	qp	q3
q3	q1	qp
qp	qp	qp

En п0 separamos en estados finales y no finales.

п0 [q0,q2,q3,qp] [q1]
п1 [q0,q3] [q2,qp] [q1]
п2 [q0] [q3] [q2] [qp] [q1]

El algoritmo termina acá ya que todas las clases contienen un único estado, lo que también nos confirma que el autómata propuesto ya era mínimo.

Ejercicio 3

Sea el lenguaje $L_3 = \{ a^p b^m \# b^t \ / \ m = q+r+s \ ; \ p > r \geq 0 \ ; \ q \geq t \geq 0 \ ; \ s > 0 \}$

a) Clasifique L_3 según la Jerarquía de Chomsky. Justifique.

El lenguaje es libre de contexto y no regular. Se probará que no es regular utilizando el contrarrecíproco del *pumping lemma* para lenguajes regulares y en la siguiente parte se probará que es libre de contexto dando un autómata push down que lo reconozca.

Consideremos entonces la tira $z= ab^{N+1}\#b^N$ donde: p=1, r=0, q=t=N, s=1

Familia 1

$u = \varepsilon$ $1+p \leq N$
$v = ab^p$ $p \geq 0$
$w = b^{N-p+1}\#b^N$

$z_i = (ab^p)^i \ b^{N-p+1}\#b^N$

Tomando i=0 se obtiene $z_0 = (ab^p)^0 \ b^{N-p+1}\#b^N = b^{N-p+1}\#b^N$, que provoca la eliminación de la **a**, por lo que se rompe la condición de que al inicio de toda tira debe haber al menos una **a** por ser p>r ≥ 0. Entonces $z_0 \notin L_3$

Familia 2

$u = ab^p$ $1+p+k \leq N$
$v = b^k$ $k \geq 1$
$w = b^{N-(p+k)+1}\#b^N$

$z_i = ab^p \ b^{(q)*i} \ b^{N-(p+k)+1}\#b^N = ab^{N+k(i-1)+1} \ \#b^N$

Tomando i=0 se obtiene $z_2 = ab^{N+k(0-1)+1}\#b^N = ab^{N-k+1}\#b^N$

Se observa que: la cantidad de las **b**'s después del **#** no cambia (es "N"), pero las primeras **b**'s deben de ser una cantidad > (mayor estricto) que las segundas **b**'s por la condición de t ≥ q, pero hay que agregarle al menos una **b** (correspondiente a la "s", que vale 1).
Con lo cual la condición de t ≥ q NO se cumple y por lo tanto $z_0 \notin L_3$

Al haber estudiado todas las descomposiciones posibles para la tira $z= ab^{N+1}\#b^N$ que cumplen $|uv|\leq N$ y $|v|\geq 1$, por el contrarrecíproco del Pumping Lema se concluye que el lenguaje L_3 **no es regular**.

Nota: Observar que no se podía tomar la tira que considerara la otra restricción p>r>0 porque si se tomaba $a^N b^N \#$ (o similar) con p=N, r=N-1, s=1, q=t=0; donde se analizaría sólo una familia, la u y v en las a's, como la **s** no tiene relación con nadie y sólo tiene que ser s>0 sumado, la relación p>r siempre puede cumplirse aunque disminuya la cantidad de a's

b) Se intenta construir una gramática libre de contexto, ya que se entiende que el lenguaje no es regular pero si libre de contexto.

Primero observamos que $a^p b^m \# b^t = a^p b^{q+r+s} \# b^t = a^p b^r . b^s . b^q \# b^t$

S → PEQ
P → aPb | A
A → aA | a
E → bE | b
Q → bQb | B
B → bB | #

La gramática no se encuentra simplificada por lo que usamos el algoritmo de simplificación:

Paso 1: Producciones épsilon
No tiene producciones épsilon

Paso 2: Producciones unitarias
Tenemos la producción unitaria P → A por lo que eliminamos esta regla y agregamos las reglas P → aA y P → a, quedando la gramática

S → PEQ
P → aPb | aA | a
A → aA | a
E → bE | b
Q → bQb | B
B → bB | #

Luego tenemos la producción unitaria Q → B por lo que eliminamos esta regla y agregamos las reglas Q → Bb y Q → #, quedando la gramática

S → PEQ
P → aPb | aA | a
A → aA | a
E → bE | b
Q → bQb | bB | #
B → bB | #

Luego no tenemos más producciones unitarias

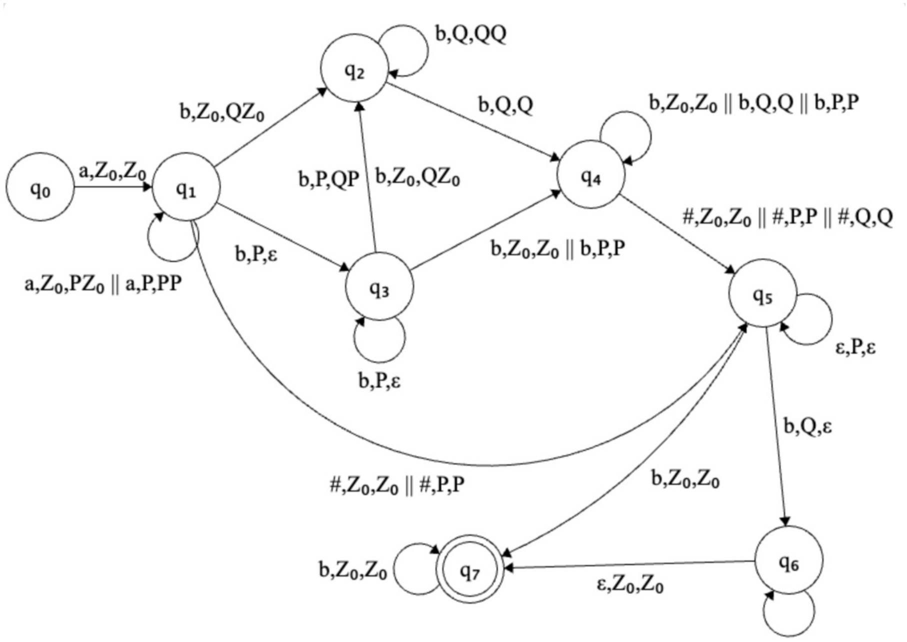
Paso 3: Variables positivas
POS_1 = {P, A, E, Q, B} compuesto por todas las varibales que derivan terminales
luego POS_2 = {P, A, E, Q, B, S} ya que S → PEQ que está compuesto de variables positivas
Todas las variables son positivas

Paso 4: Variables alcanzables
ALC_1={S}
ALC_2={S, P, E, Q}
ALC_3={S, P, E, Q, A, B}
Todas las variables son alcanzables

La gramática simplificada resultante es:

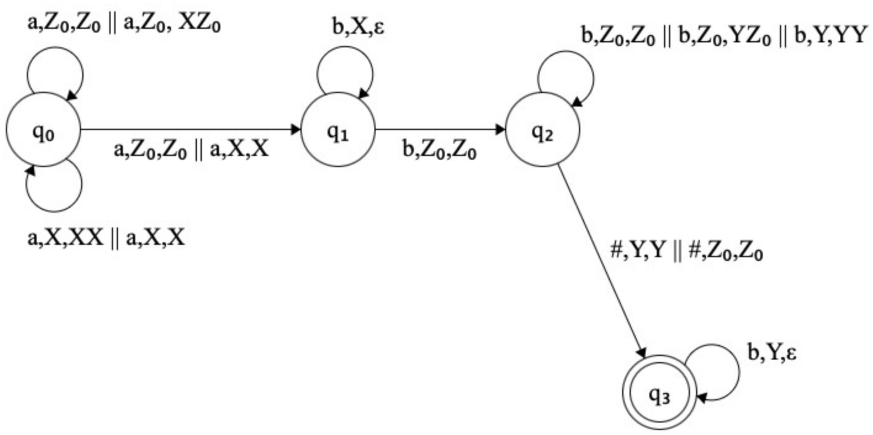
S → PEQ
P → aPb | aA | a
A → aA | a
E → bE | b
Q → bQb | bB | #
B → bB | #

c) Ahora se va a construir un APD para reconocer el lenguaje L_3.



El autómata no es determinista ya que en q_3 teniendo en el tope del stack Z_0 y consumiendo el símbolo 'b' tengo dos transiciones posibles: una al estado q_2 y otra al estado q_3.

Otra versión del APD podría ser:



El autómata no es determinista ya que en q_0 teniendo en el tope del stack Z_0 y consumiendo el símbolo 'a' tengo dos transiciones posibles: una al estado q_0 y otra al estado q_1.

Ejercicio 4

Sea el lenguaje $L_4 = \{ x = 0^k1^k w \ / \ |w| = 2*k, \text{ con } w \in \{0,1\}^* \}$

a) Como la letra dice que el lenguaje no es libre de contexto, se construye una gramática irrestricta.
A continuación las reglas:

S → IS'F
S' → 0AS'1A | ϵ
A0 → 0A
A1 → 1A
AF → F0 | F1
I0 → 0I
I1 → 1I
IF → ϵ

b) De la misma forma, el autómata a dar para reconocer las tiras de L_4 es una MT