P1-Soluciones-Explicadas-2022.pdf

Aabad

Autómatas y Lenguajes

3º Grado en Ingeniería Informática

Escuela Politécnica Superior
Universidad Autónoma de Madrid

AUTÓMATAS Y LENGUAJES

Examen Parcial, 11 de octubre de 2022

Bloque 1. Expresiones Regulares y Autómatas

1. **Expresiones regulares:**
   a) (2 pts.) Propón de manera razonada una expresión regular para el lenguaje sobre el alfabeto {0,1} que consiste en todas las cadenas en las que todo par de ceros consecutivos es seguido por un par de unos consecutivos.

   Ejemplo de cadenas válidas: λ, 0, 01, 1, 0011, 100110, 1101110011010101

   Ejemplo de cadenas inválidas: 00, 001, 1110010

   Justifica tu respuesta.

a) (01+0011 +1)*(0+λ)

Hay que hacer que, si se quieren poner 2 ceros, obligatoriamente haya 2 unos detrás. Para conseguirlo se da la opción de 3 cadenas que acaban todas en uno, y tras ellas se puede añadir un 0 final.

b) (1 pt.) Considera estas dos expresiones regulares:

   R = a*(ba*ba*)*   S = a*(ba*b)a*

   Demuestra que ambas expresiones no son equivalentes, indicando una cadena que pertenezca al lenguaje de una de estas expresiones, pero no al de la otra.

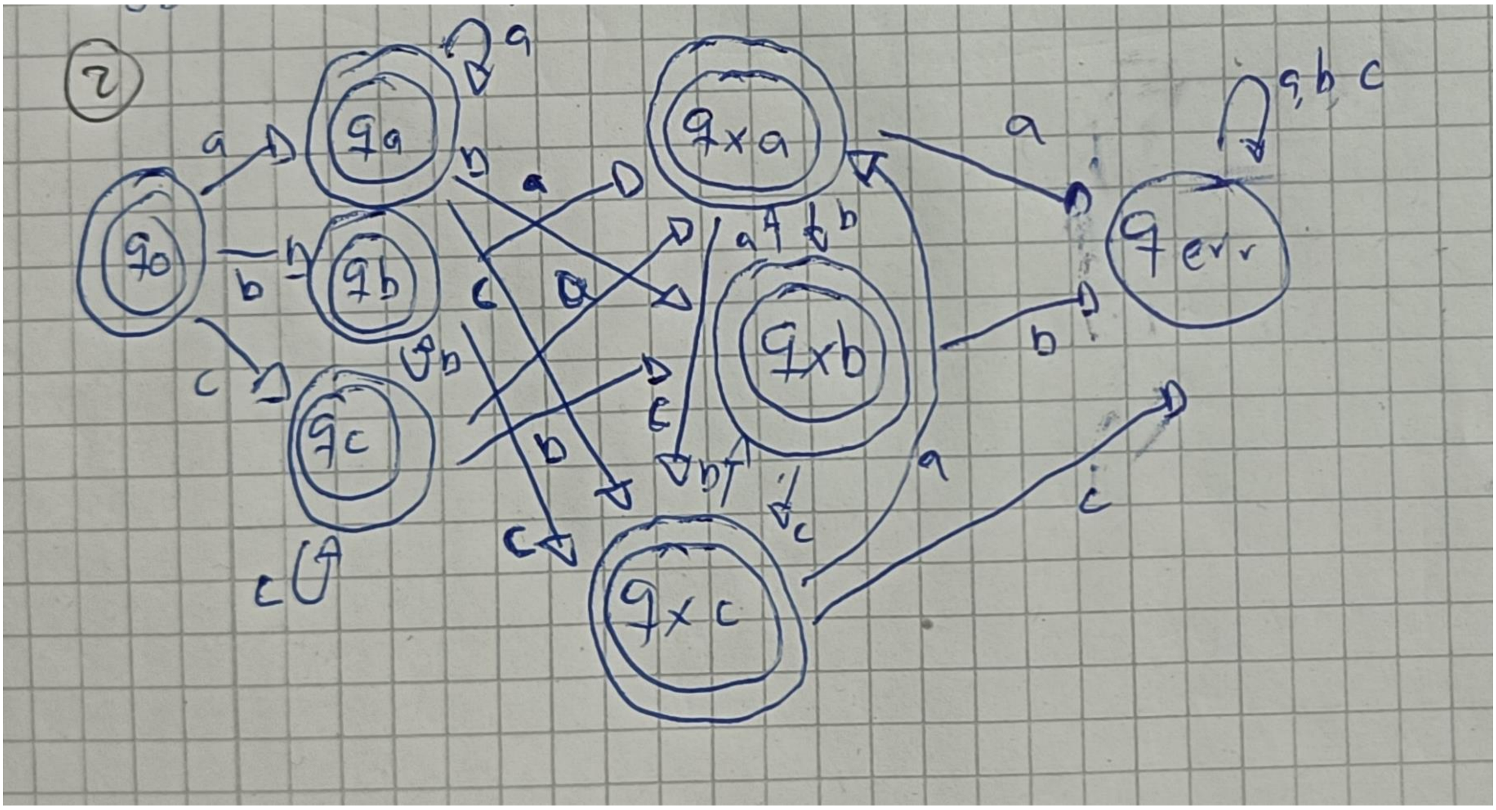
b) Basta ver que bbbb ∈ R pero bbbb ∉ S. Las cadenas de R pueden tener cualquier número par de b`s, mientras que las de S deben tener exactamente 2.

2. Autómatas finitos (3 pts.)

Diseña un autómata finito, no necesariamente determinista, para el lenguaje formado por palabras de cualquier longitud, con los símbolos “a”, “b” y “c”, y que cumplan **alguna** de las siguientes condiciones:

1. No hay dos símbolos iguales contiguos.
2. Todos los símbolos son iguales.

Nota: cualquier palabra de longitud 0 o 1 también cumple cualquiera de las condiciones anteriores.



He supuesto que la cadena vacía también es válida (en caso contrario q0 no sería estado final).

qa, qb y qc son los estados en los que acabarán aquellas cadenas en las que todos los símbolos son iguales. Entre los estados qxa, qxb y qxc se encuentran las cadenas que no tengan 2 símbolos iguales contiguos, en cuanto haya 2 contiguos el único estado en el que se encontrará será el de error, del que no saldrá y no es final. Por lo tanto sólo serán aceptadas las cadenas que cumplan las codiciones del enunciado.

3. De ER a AF (2 pts.) Considera la expresión regular *((ab+b)*a)** de palabras sobre el alfabeto {a, b}. Diseña un autómata finito para esta expresión **siguiendo el método general de reducción** de expresiones regulares a AFN-λ visto en clase. Para ello, muestra la construcción para cada una de las subexpresiones *a*, *b*, *ab*, etc. y el autómata final. No simplifiques ninguno de estos subautómatas, sigue el procedimiento general.

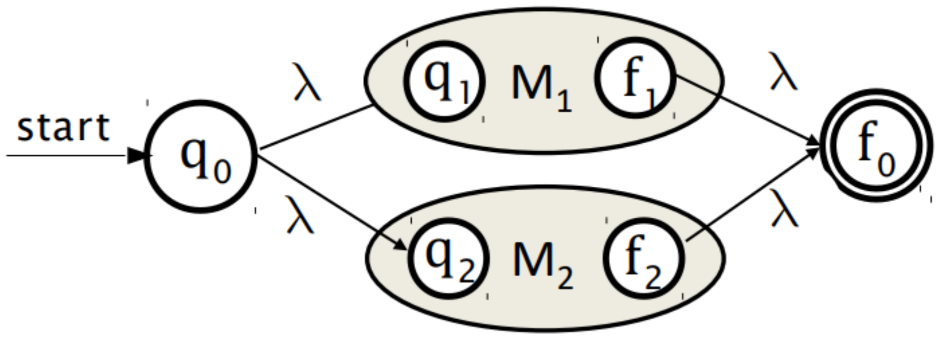
Para seguir el algoritmo hay que dividirlo en las distintas subcadenas que lo componen:

1) a
2) b
3) ab
4) ab+a
5) (ab+a)*
6) (ab+b)*a
7) ((ab+b)*a)*

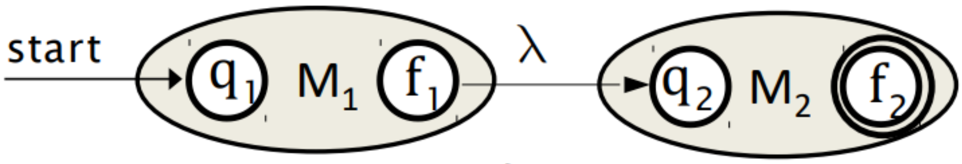
Reducción 1: de expresiones regulares a AFN-λ

Distinguimos tres casos para construir *M*:

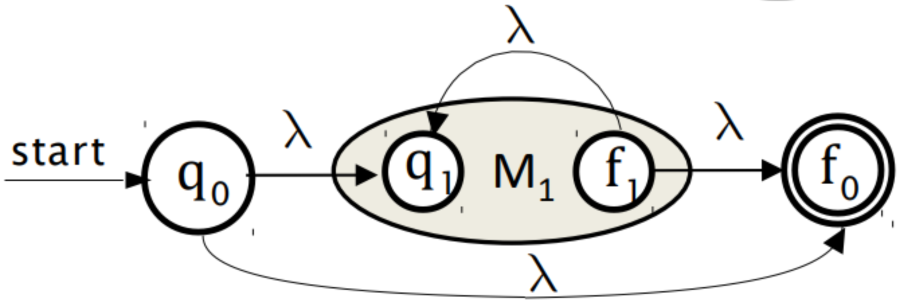
1.
R = $\alpha + \beta$



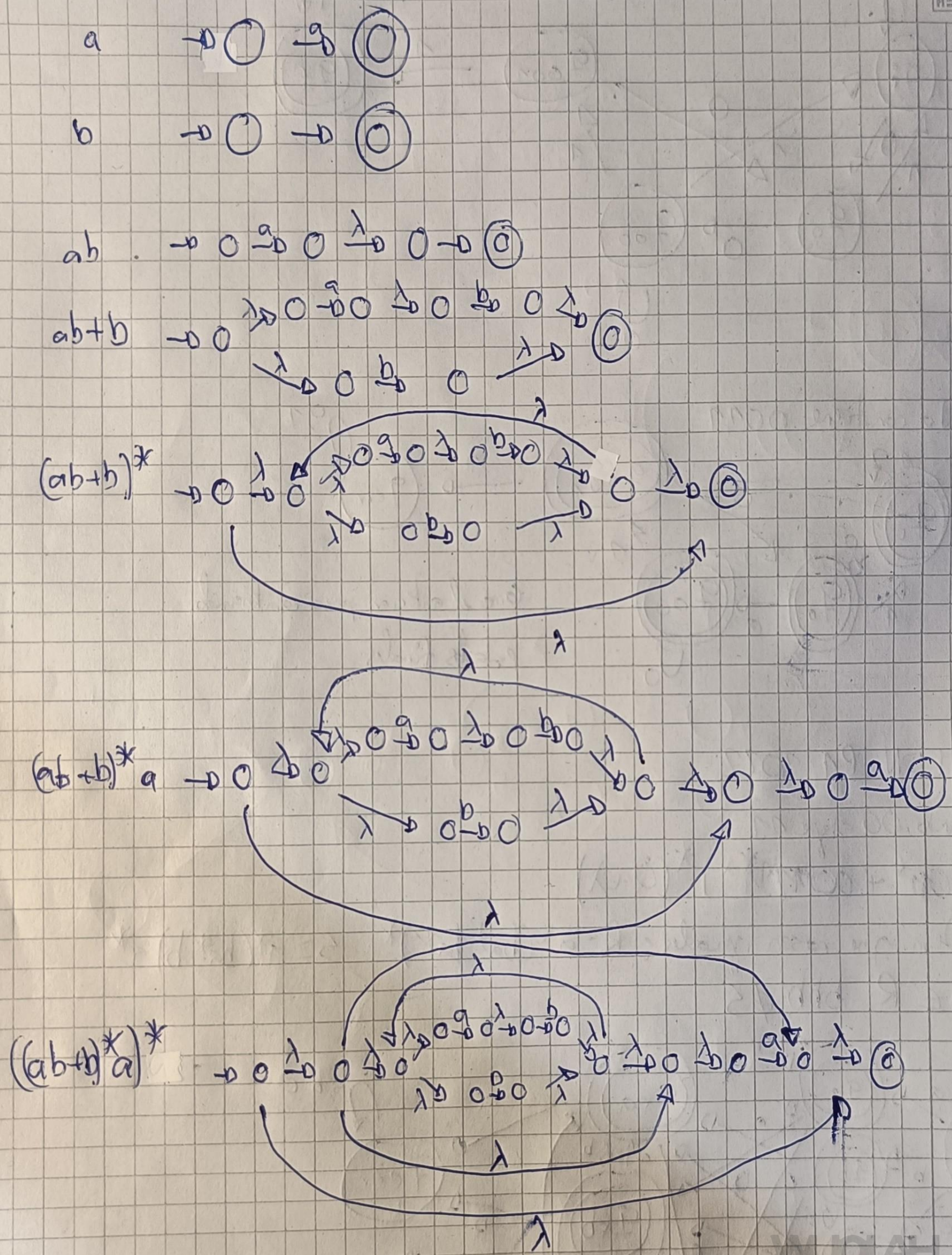
2.
R = $\alpha.\beta$



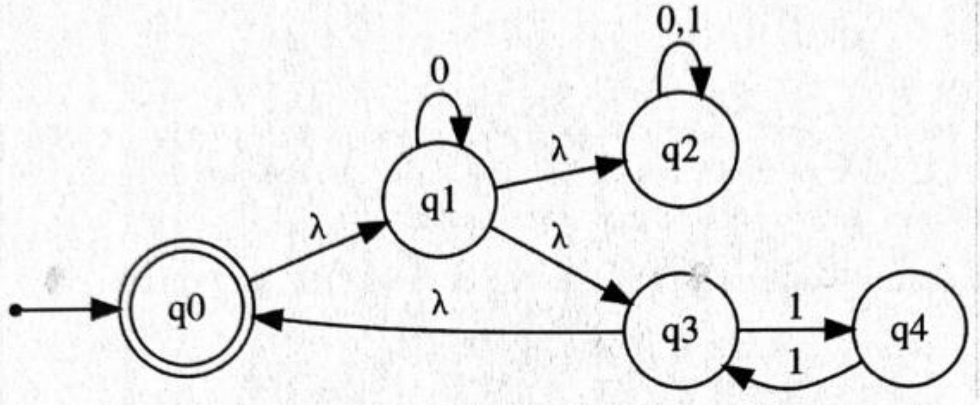
3.
R = α^*



③ AFN-λ para $((ab+b)^*a)^*$



4. De AFN a AFD (2 pts.) Dado el siguiente Autómata:



Crea el AF Determinista equivalente, siguiendo cualquier de los métodos vistos en clase. Refleja en el nombre de los estados del AFD el nombre de los estados del AFND de los que proceden.

Se saca la tabla de estados de autómata no determinada a autómata determinado, marcando como finales todos aquellos conjuntos de estados que incluyan algún final, en este caso q0.

(4) AFND-λ a AFD

	λ	0	1
q0	q1	∅	∅
q1	q2,q3	q1	∅
q2	∅	q2	q2
q3	q0	∅	q4
q4	∅	∅	q3

	0	1
*{q0,q1,q2,q3}	{q0,q1,q2,q3}	{q4,q2}
{q2,q4}	{q2}	{q2,q3,q0,q1}
{q2}	{q2}	{q2}