Práctico 4: Equivalencia entre Lenguajes Regulares y Autómatas Finitos (Teorema de Kleene)

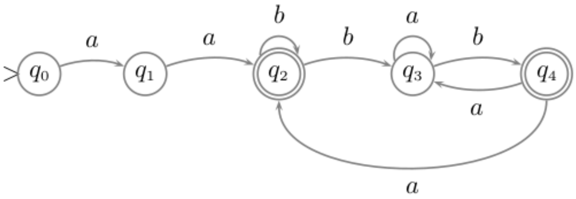
Año 2024

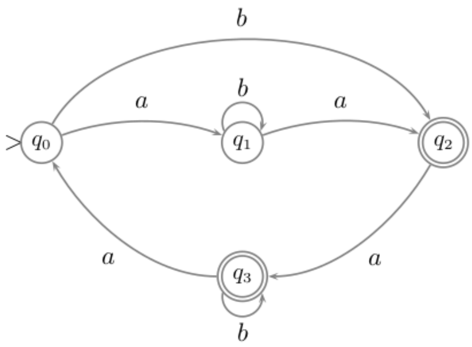
Ejercicio 1. Sean $L_1 = a^*b^*$ y $L_2 = ((a+b)(a+b))^*$, construir recursivamente utilizando el Teorema de Kleene (ida) un AF que acepta $L_1 \cup L_2$, L_1L_2 y L_1^*.

Ejercicio 2. Hallar las soluciones de las siguientes ecuaciones utilizando el Lema de Arden:

- $X = aX + bX$
- $X = aX + b^*ab + bX + a^*$
- $X = ab^2X + aX + a^*b + b^*a$

Ejercicio 3. Para cada uno de los AF's que se muestran a continuación, hallar su expresión regular equivalente utilizando el Teorema de Kleene (vuelta):





Ejercicio 4. Probar que todo AF tiene un AF equivalente con exactamente un único estado de aceptación: si $M = (\Sigma, Q, q_0, F, \Delta) \in AFN\epsilon^\Sigma$, entonces $\exists M' = (\Sigma, Q', q_0', F', \Delta') \in AFN\epsilon^\Sigma$ tal que $L(M) = L(M')$ y $|F'| = 1$.

Ejercicio 5. Probar que si $L \in LR^\Sigma$, entonces $\overline{L} \in LR^\Sigma$ dando un AF que acepta $\overline{L}$.

Ejercicio 6. Sea $\Sigma = \{0,1\}$ un alfabeto binario, entonces definimos recursivamente el siguiente operador unario sobre cadenas de Σ^*:

$$\hat{\epsilon} = \epsilon$$
$$\widehat{0\alpha} = 1\hat{\alpha}$$
$$\widehat{1\alpha} = 0\hat{\alpha}$$

Notar que el operador $\hat{}$ es el complemento bit a bit de la cadena. Por último, para todo $L \subseteq \{0,1\}^*$, definimos $\hat{L} = \{\hat{\alpha} : \alpha \in L\}$. Probar que si $L \in LR^\Sigma$, entonces $\hat{L} \in LR^\Sigma$ dando un AF que acepta $\hat{L}$.

Ejercicio 7. Probar que si $L_1 \in LR^\Sigma$ y $L_2 \in LR^\Sigma$, entonces $(L_1 \cap L_2) \in LR^\Sigma$, dando un AF que acepta $(L_1 \cap L_2)$. (Ayuda: pensar en el producto cartesiano de dos autómatas).

Ejercicio 8. Probar que si $L \in LR^\Sigma$, entonces $L^R \in LR^\Sigma$ dando un AF que acepta L^R.