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 $|F'| = 1$.

$M = (\Sigma, Q, q_0, F, \Delta) \in AFN_{\epsilon}^{\Sigma} \Rightarrow \exists M' = (\Sigma, Q', q_0', F', \Delta') \in AFN_{\epsilon}^{\Sigma}$

tal que $L(M) = L(M')$ y $|F'| = 1$

1- Definición inicial del autómata:

- Partimos de un AF M con:
  - Un conjunto de estados Q
  - Un estado inicial q_0
  - Un conjunto de estados de aceptación F
  - Una funcion de transición Δ

2- Construcción de M'

Introducimos un nuevo estado de aceptación único q_f que no pertenece a Q

Modificamos las transiciones del autómata original:

Para cada estado q pertenece F en el autómata original, agregamos una transición (q,e(epsilon),q_f). Esto permite que cualquier estado de aceptación original pueda llegar al nuevo estado de aceptación q_f mediante una transición vacía(e)

Eliminamos F como conjunto de estados de aceptación, sustituyendolo con F'={q_f}

3- Propiedades preservadas:

Este cambio no altera el lenguaje reconocido por el autómata, ya que cualquier cadena que antes llevaba a un estado de F, ahora puede llegar al único estado q_f por una transición ε.

4- Verificación formal:

El nuevo autómata M' se define como:

$$M' = (\Sigma, Q \cup \{q_f\}, q_0, \{q_f\}, \Delta'\}$$

Donde Δ' es:

$$\Delta' = \Delta \cup \{ (q, \varepsilon, q_f) \mid q \in F \}$$

Dado que ε-transiciones no afectan la aceptabilidad de cadenas, se cumple que L(M')=L(M)

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

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

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 $\hat{L}$.

1. **Operador bit a bit** ($\hat{\cdot}$):
   - $\hat{\varepsilon} = \varepsilon$ (el complemento de la cadena vacía es la cadena vacía).
   - $\hat{0}\alpha = 1\hat{\alpha}$ (el complemento de una cadena que empieza en 0 cambia ese bit a 1 y complementa recursivamente el resto).
   - $\hat{1}\alpha = 0\hat{\alpha}$ (lo mismo para 1).

2. **Operador sobre lenguajes**:

$$\hat{L} = \{\hat{\alpha} : \alpha \in L\},$$

es el lenguaje resultante de aplicar el operador de complemento a todas las cadenas en L.

3. **Propiedad a demostrar**: Si $L \in LR_\Sigma$ (lenguaje regular sobre Σ), entonces $\hat{L} \in LR_\Sigma$.

Estrategia:

Para demostrarlo, construiremos un autómata M' que acepte $\hat{L}$, partiendo de un autómata M que acepte L.

Construcción del autómata M':

1. Sea $M = (\Sigma, Q, q_0, F, \Delta)$, el autómata que acepta L.
   - Q: conjunto de estados.
   - q_0: estado inicial.
   - F: conjunto de estados de aceptación.
   - Δ: función de transición.
2. Construimos $M' = (\Sigma, Q', q_0', F', \Delta')$ tal que:
   - Los estados de M' son los mismos que en M: $Q' = Q$.
   - El conjunto de aceptación F' de M' corresponde a las imágenes de F bajo el operador $\hat{}$, es decir, $F' = \{q \in Q : \hat{q} \text{ es aceptado en } M\}$.
3. Transiciones (Δ'):
   - Para cada transición (q, a, q') en M, añadimos la transición complementaria $(q, \hat{a}, q')$ en M'.
   - Aquí, $\hat{a}$ es el complemento del bit a: $\hat{0} = 1$ y $\hat{1} = 0$.

Demostración de la propiedad:

El lenguaje aceptado por M' es exactamente $\hat{L}$:

- Para cada cadena $\alpha \in L$, existe una secuencia de transiciones en M que lleva desde q_0 a un estado de aceptación en F.
- En M', esa misma secuencia de transiciones, con los bits complementados ($\hat{\alpha}$), llevará desde q_0 a un estado en F'.
- Por lo tanto, M' acepta $\hat{L}$.

Conclusión:

Hemos construido un autómata M' que acepta $\hat{L}$, demostrando que $\hat{L} \in LR_\Sigma$ si $L \in LR_\Sigma$. Por tanto, el operador de complemento bit a bit preserva la regularidad del lenguaje.

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).

Estrategia:

El producto cartesiano de dos autómatas M_1 y M_2, que aceptan L_1 y L_2 respectivamente, genera un nuevo autómata M que acepta la intersección $L_1 \cap L_2$.

Construcción:

Sean $M_1 = (\Sigma, Q_1, q_{0,1}, F_1, \Delta_1)$ y $M_2 = (\Sigma, Q_2, q_{0,2}, F_2, \Delta_2)$, los autómatas que aceptan L_1 y L_2. Construimos el autómata $M = (\Sigma, Q, q_0, F, \Delta)$ tal que:

1. **Estados** (Q):
   - Los estados de M son el producto cartesiano de los estados de M_1 y M_2:

$$Q = Q_1 \times Q_2 = \{(q_1, q_2) \mid q_1 \in Q_1, q_2 \in Q_2\}.$$

2. **Estado inicial** (q_0):
   - El estado inicial de M es el par formado por los estados iniciales de M_1 y M_2:

$$q_0 = (q_{0,1}, q_{0,2}).$$

3. **Estados de aceptación** (F):
   - Los estados de aceptación de M son los pares donde ambos componentes son estados de aceptación en M_1 y M_2:

$$F = F_1 \times F_2 = \{(q_1, q_2) \mid q_1 \in F_1 \text{ y } q_2 \in F_2\}.$$

4. **Transiciones** (Δ):
   - Para cada símbolo $a \in \Sigma$, las transiciones de M están definidas como:

$$\Delta((q_1, q_2), a) = (\Delta_1(q_1, a), \Delta_2(q_2, a)).$$

   - Esto significa que M realiza una transición simultánea en M_1 y M_2 al leer el mismo símbolo a.

Propiedad del lenguaje aceptado:

El nuevo autómata M acepta una cadena w si y solo si:

1. w lleva al autómata M_1 desde $q_{0,1}$ a un estado en F_1, y
2. w lleva al autómata M_2 desde $q_{0,2}$ a un estado en F_2.

Por construcción, esto implica que M acepta exactamente las cadenas en $L_1 \cap L_2$.

Ejemplo concreto:

Supongamos:

- $L_1 = \{w \mid w \text{ contiene un número par de 0s}\}$,
- $L_2 = \{w \mid w \text{ contiene un número par de 1s}\}$.

Autómatas individuales:

1. M_1 tiene dos estados (q_{even}, q_{odd}) y alterna al leer 0.
2. M_2 tiene dos estados (q_{even}, q_{odd}) y alterna al leer 1.

Producto cartesiano: El autómata del producto tendrá los estados:

$$Q = \{(q_{even}, q_{even}), (q_{even}, q_{odd}), (q_{odd}, q_{even}), (q_{odd}, q_{odd})\},$$

con transiciones combinadas. Los estados de aceptación serán:

$$F = \{(q_{even}, q_{even})\}.$$

Este autómata aceptará cadenas que tengan un número par de 0 y un número par de 1, es decir, $L_1 \cap L_2$.

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

Estrategia:

Dado un autómata M que acepta L, construiremos un nuevo autómata M^R que acepte L^R. La construcción de M^R invierte las transiciones de M y ajusta los estados inicial y de aceptación.

Construcción de M^R:

Sea $M = (\Sigma, Q, q_0, F, \Delta)$ el autómata que acepta L. Construimos $M^R = (\Sigma, Q', q_0', F', \Delta')$ como sigue:

1. **Estados (Q'):**
   - Los estados de M^R son los mismos que los de M: $Q' = Q$.
2. **Estado inicial (q_0'):**
   - En M^R, el nuevo estado inicial será el conjunto de estados de aceptación originales F, ya que en L^R, las cadenas deben comenzar desde donde terminaban en L.
3. **Estados de aceptación (F'):**
   - El nuevo conjunto de estados de aceptación será el estado inicial de M: $F' = \{q_0\}$.
4. **Transiciones (Δ'):**
   - Invertimos todas las transiciones de M. Para cada transición $(q, a, q') \in \Delta$, agregamos la transición invertida (q', a, q) en Δ'.

Propiedad del lenguaje aceptado:

- Una cadena w pertenece a L^R si y solo si w es la inversa de una cadena en L.
- Dado que las transiciones están invertidas en M^R, el autómata sigue las transiciones de M en orden inverso, llevando desde un estado de aceptación F hasta el estado inicial q_0.

Ejemplo:

Supongamos $L = \{ab, ba\}$. Su reflexión es $L^R = \{ba, ab\}$.

Autómata M:

- Estados: $Q = \{q_0, q_1, q_2\}$,
- Estado inicial: q_0,
- Estado de aceptación: q_2,
- Transiciones: $\Delta = \{(q_0, a, q_1), (q_1, b, q_2), (q_0, b, q_1), (q_1, a, q_2)\}$.

Autómata M^R:

- Estados: $Q' = Q$,
- Estado inicial: q_2,
- Estado de aceptación: $F' = \{q_0\}$,
- Transiciones: $\Delta' = \{(q_1, a, q_0), (q_2, b, q_1), (q_1, b, q_0), (q_2, a, q_1)\}$.

Este nuevo autómata acepta $L^R = \{ba, ab\}$.