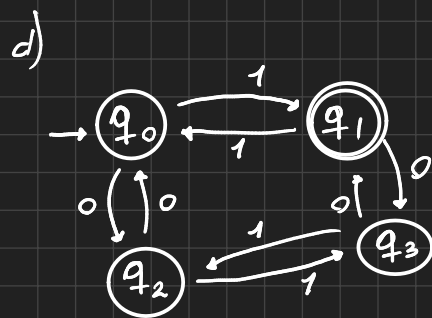
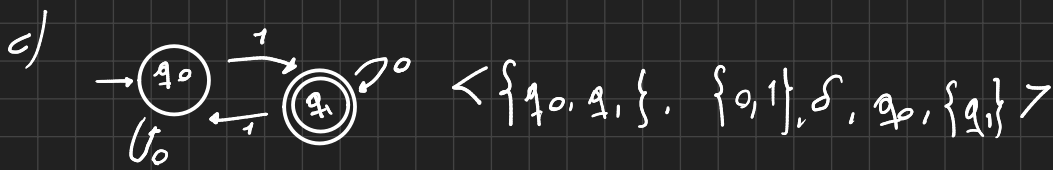
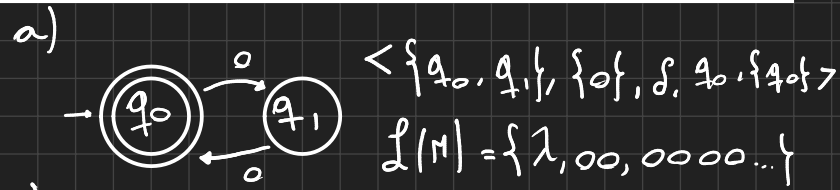
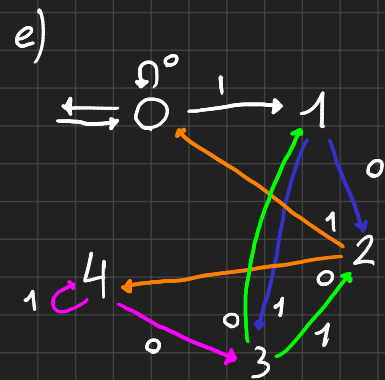


Ejercicio 1. Construir autómatas finitos para los siguientes lenguajes:

- Cadenas sobre $\Sigma = \{0\}$ de longitud par.
- Cadenas sobre $\Sigma = \{0, 1\}$ con cantidad par de ceros.
- Cadenas sobre $\Sigma = \{0, 1\}$ con cantidad impar de unos.
- Cadenas sobre $\Sigma = \{0, 1\}$ con cantidad par de ceros y cantidad impar de unos.
- Cadenas sobre $\Sigma = \{0, 1\}$ que, interpretadas como un número binario, sean congruentes a cero módulo 5.¹

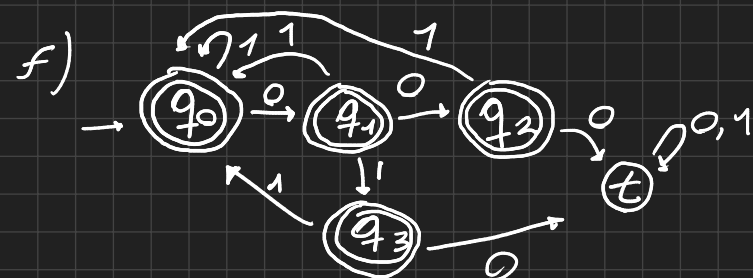
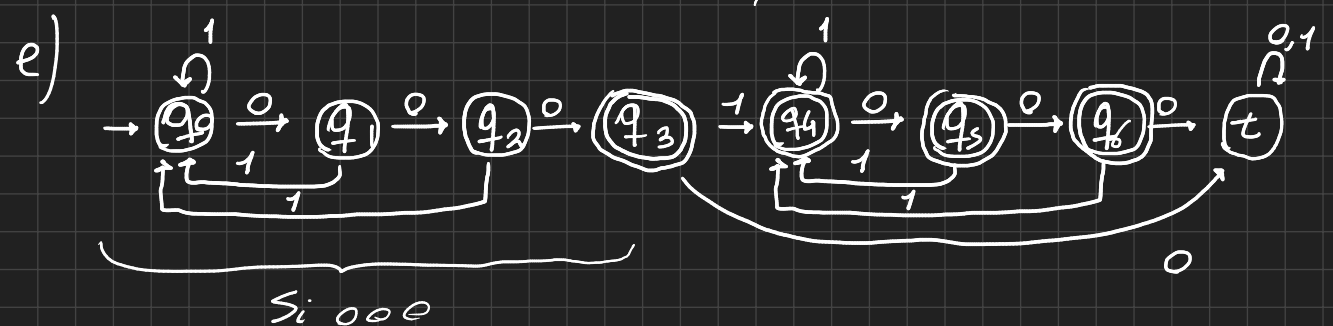
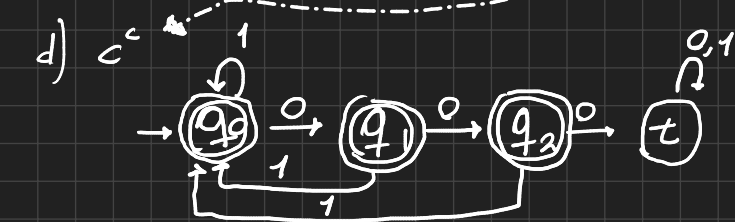
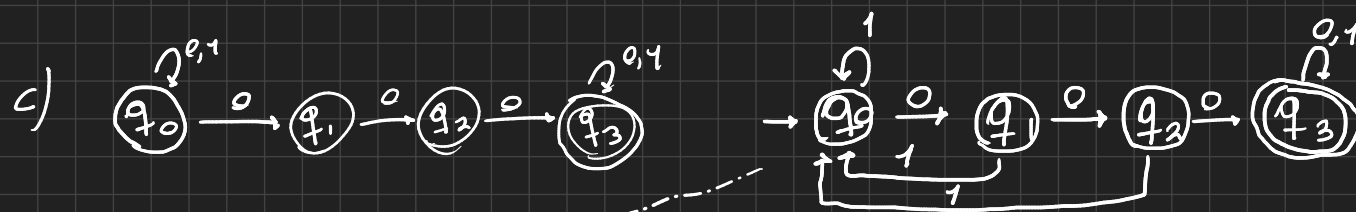
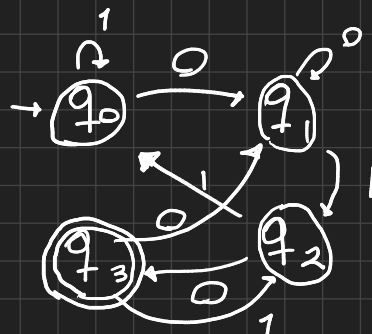
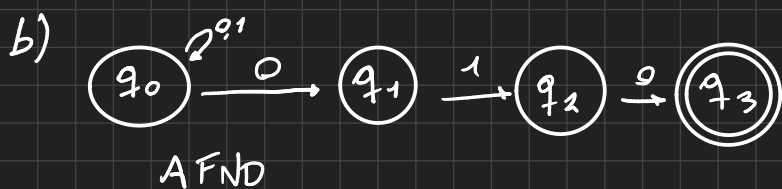
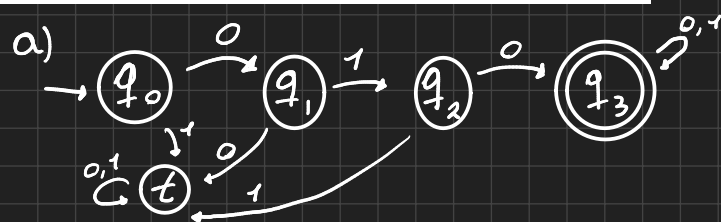


0
1 0 1
1 0 1 0
1 1 1 1
1 0 1 0 0
1 1 0 0 1
1 1 1 1 0
1 0 0 0 1 1



Agrega algo al final en base Um 2.
0 en 2 1 en $(\cdot 2) + 1$. Cada estado
en Um resto 0 en inicio y final

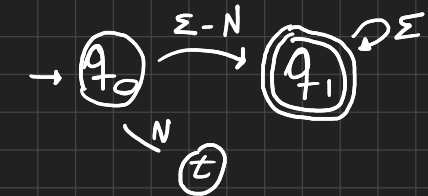
- Ejercicio 2.** Construir autómatas finitos para los siguientes lenguajes sobre $\Sigma = \{0, 1\}$:
- Cadenas que comiencen con 010.
 - Cadenas que terminen con 010.
 - Cadenas que contengan la subcadena 000.
 - Cadenas que no contengan la subcadena 000.
 - Cadenas que contengan la subcadena 000 exactamente una vez (la cadena 0000 no pertenece a este lenguaje).
 - Cadenas que no contengan la subcadena 000 ni la 010.



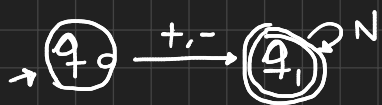
Ejercicio 3. Construir autómatas finitos para los siguientes lenguajes:

- Identificadores de cualquier longitud que comiencen con una letra o guión y contengan letras, dígitos o guiones.
- Constantes enteras con signo.
- Constantes enteras con signo opcional.
- Constantes reales con signo. Ejemplos: +123.456, -55.0, +00.430.
- Constantes reales con signo opcional y partes enteras y fraccionarias opcionales. Ejemplos: los anteriores más 123.456, -55., +.43. → ?
- Constantes reales con notación exponencial opcional. Ejemplos: los anteriores más -55.E5, +.43E-6.

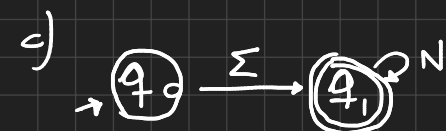
a) $\Sigma = \{a \dots z, -, 1 \dots 9\}$ $N \subseteq \Sigma$ $N = \{0, 1 \dots 9\}$



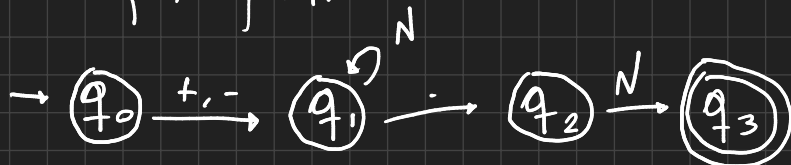
b) $N = \{0, 1, 2, 3, 4, 5, 6, 7, 8, 9\}$ $\Sigma = N \cup \{-, +\}$



Δ sobre que 58-751+1+-+3 o +--+-
 En involucro y el signo en -3
 +5

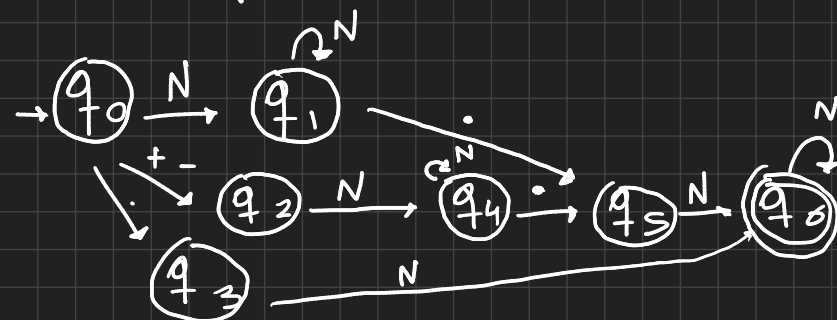


d) $\Sigma = \{+, -, \cdot\} \cup N$



Δ sumi que +15. no es válido
 ni -8.
 Pero -00 > .00 N'

e) Poner que .q3 es válido



Por ser AFD falta
 con los q faltan sean a un
 t pero no o queda Fec

f) $\Sigma' = \Sigma + \{E\}$ E maybe no des de b como ...N5?

Ejercicio 4. Dado un autómata finito para \mathcal{L} , indicar cómo construir autómatas finitos para los siguientes lenguajes. Indicar en cada caso si es necesario que el autómata de entrada sea determinístico o no, y de qué tipo es el autómata resultante.

- \mathcal{L}^c , el complemento de \mathcal{L} .
- \mathcal{L}^* , la clausura de Kleene de \mathcal{L} .
- \mathcal{L}^r , la reversa de \mathcal{L} .
- $\text{Ini}(\mathcal{L}) = \{\alpha \mid \exists \beta \text{ tal que } \alpha\beta \in \mathcal{L}\}$, los prefijos de \mathcal{L} .
- $\text{Fin}(\mathcal{L}) = \{\alpha \mid \exists \gamma \text{ tal que } \gamma\alpha \in \mathcal{L}\}$, los sufijos de \mathcal{L} .
- $\text{Sub}(\mathcal{L}) = \{\alpha \mid \exists (\beta, \gamma) \text{ tales que } \gamma\alpha\beta \in \mathcal{L}\}$, las subcadenas de \mathcal{L} .
- $\text{Máx}(\mathcal{L}) = \{\alpha \in \mathcal{L} \mid \forall \omega \in \Sigma^+, \alpha\omega \notin \mathcal{L}\}$, las cadenas maximales de \mathcal{L} .
- $\text{Mín}(\mathcal{L}) = \{\alpha \in \mathcal{L} \mid \text{ningún prefijo propio de } \alpha \text{ pertenece a } \mathcal{L}\}$, las cadenas minimales de \mathcal{L} . Es decir, $\text{Mín}(\mathcal{L}) = \{\alpha \in \mathcal{L} \mid \nexists (\omega_1, \omega_2) \text{ tales que } \alpha = \omega_1\omega_2 \wedge \omega_1 \in \mathcal{L} \wedge \omega_2 \neq \lambda\}$.
- $\mathcal{L}_T = \{\alpha \in \Sigma^* \mid \exists (\omega_1 \in \mathcal{L}, \omega_2 \in \Sigma^*) \text{ tales que } \alpha = \omega_1\omega_2\} = \mathcal{L}.\Sigma^*$.

Sea M el autómata dado y M' el resultado

a) M debe ser determinístico, lo que invertirá los estados finales con no finales

b) $M' = \textcircled{q_0} \xrightleftharpoons[\lambda]{\lambda} M$ dice que en la forma rápida, en AFND

c) Como que M puede ser AFD o AFND. $M' = M$ con las flechas invertidas y final y no final invertidos. Siempre M' en AFND

4. **Prefijos ($\text{Ini}(\mathcal{L})$):** Make all states from which an accept state is reachable accepting.

5. **Suffixes ($\text{Fin}(\mathcal{L})$):** Add new start state with ϵ -transitions to all states (NFA).

6. **Substrings ($\text{Sub}(\mathcal{L})$):** Combine prefix and suffix operations.

7. **Maximal strings ($\text{Máx}(\mathcal{L})$):** Remove transitions from accept states (ensure no extensions).

8. **Minimal strings ($\text{Mín}(\mathcal{L})$):** Accept states are those with no path to another accept state via non-empty string.

9. **Right quotient (\mathcal{L}_T):** Add ϵ -transitions from accept states to a new universal state (NFA). ?

Ejercicio 5. Dados autómatas finitos para \mathcal{L}_1 y \mathcal{L}_2 indicar cómo construir autómatas finitos para los siguientes lenguajes, con las mismas consideraciones que en el ejercicio anterior:

- $\mathcal{L}_1 \cup \mathcal{L}_2$
- $\mathcal{L}_1 \cap \mathcal{L}_2$
- $\mathcal{L}_1 \cdot \mathcal{L}_2$
- $\mathcal{L}_1 \setminus \mathcal{L}_2$



b) $\mathcal{L}_1 \cap \mathcal{L}_2 = \overline{\overline{\mathcal{L}_1} \cap \overline{\mathcal{L}_2}} = \overline{\mathcal{L}_1 \cup \mathcal{L}_2}$



c) 1. $\textcircled{q_m}$ es un estado final de \mathcal{L}_1 , dejó de serlo y con una transición instantánea conecta al punto de entrada de \mathcal{L}_2

d) $\mathcal{L}_1 \setminus \mathcal{L}_2 = \mathcal{L}_1 \cap \Delta_2^c = \overline{\overline{\mathcal{L}_1} \cup \mathcal{L}_2} = \overline{\mathcal{L}_1 \cup \mathcal{L}_2}$

Ejercicio 6. Demostrar que para todo autómata finito determinístico su relación de transición \vdash cumple:

- Determinismo:** $((q, \alpha) \vdash (r, \lambda) \wedge (q, \alpha) \vdash (s, \lambda)) \implies r = s$
- Concatenación:** $((q, \alpha) \vdash (q_1, \lambda) \wedge (q_1, \beta) \vdash (r, \lambda)) \implies (q, \alpha\beta) \vdash (r, \lambda)$
- Siempre toma un estado:** $(q, \alpha\beta) \vdash (r, \lambda) \implies \exists q_1 ((q, \alpha) \vdash (q_1, \lambda) \wedge (q_1, \beta) \vdash (r, \lambda))$
- Linealidad:** $(q, \alpha) \vdash^n (r, \lambda) \iff |\alpha| = n$
- Invariancia:** $(q, \alpha) \vdash (q, \lambda) \implies \forall i \in \mathbb{N} ((q, \alpha^i) \vdash (q, \lambda))$

$\overline{\mathcal{L}_1}$ y $\overline{\mathcal{L}_2}$ son complementos de \mathcal{L}_1 y \mathcal{L}_2 y use \mathcal{L} como el autómata...

