**** SOLUCIONES ** EXAMEN DE AUTÓMATAS (2° Grado en Informática, Enero-2018)**

**Apellidos, nombre:** **DNI:** **GRUPO:** 1 / 2 / 3 / PCEO
**¿Has presentado las prácticas?** señala NO / SI / Año anterior

**Instrucciones:** este ENUNCIADO debe entregarse al salir (los folios en sucio se entregan, pero aparte).
Todos los apartados y subaparatados deben ir claramente indicados. Las faltas ortográficas restan puntos. Un examen desordenado y poco legible resta puntos.

---

**Parte I: PREGUNTAS TIPO TEST.** Total 3 puntos sobre 10: respuesta bien +0.3 y respuesta mal -0.15.

1. [ENE2018] Sean los lenguajes $A = \{ww|w \in \{a,b\}^*\}$ y $B = \{w \in \{a,b\}^*\}$. Indica la respuesta **verdadera**:
   a) $A \cup B = B$. V: $A \subset B$
   b) $A \cap B = B \cup A$.
   c) $\lambda \notin B^+$.
2. [ENERO2018] Sea $M_1$ un AF y $M_{min}$ el autómata mínimo equivalente completo, indica la respuesta **verdadera**:
   a) $M_1$ obligatoriamente tiene más estados pero podría tener el mismo número de estados finales.
   b) El número de estados finales de $M_{min}$ es siempre menor, exceptuando el estado trampa.
   c) $M_1$ podría tener menos estados que $M_{min}$ independientemente del estado trampa. V: $M_1$ podría ser AFND
3. [ENERO2018] En relación a las funciones de transición de los $AFND$, indica la respuesta **verdadera**:
   a) La función se define como: $\delta : Q \times V \cup \{\lambda\} \rightarrow Q^*$
   b) Al ser no determinista, puede devolver diferentes resultados según una probabilidad.
   c) A diferencia de la función de transición de los $AFD$, devuelve un conjunto de estados. V: por definición
4. [ENERO2018] Dada la expresión regular $ER = (a^*b^*)^*a^*c^*$, indica la respuesta **verdadera**:
   a) $L(ER) = \{(a^jc^k)^ic^i \mid i,j,k \geq 0\}$.
   b) $L(ER) = \{wc^i \mid w \in \{a,b\}^*, i \geq 0\}$. V: $(a^*b^*)^*a^*c^* = (a|b)^*a^*c^* = (a^*b)^*a^*a^*c^* = (a^*b)^*a^*c^* = (a|b)^*c^*$
   c) $L(ER) = \{(a^jc^k)^lc^i \mid i,j,k,l \geq 0\}$.
5. [$ENE$2018] Considerando las equivalencias entre expresiones regulares, indica la respuesta **verdadera**:
   a) $(0|1)((0|1)1)^* = ((0|1)1)^*(0|1)$.
   b) $0(\lambda|00^*) = 00^*$. V: $0(\lambda|00^*) \overbrace{=}^{R13} 00^*$
   c) $0^*(10^*)^* = (10|0)^*$.
6. [ENERO2018] Dado el lenguaje $L_1 = \{a^ib^{i+2} \mid i \geq 0\}$, indica la respuesta **verdadera**:
   a) Es posible encontrar una gramática regular $G_R$ para que $L(G_R) = L_1$.
   b) Es posible diseñar un $AF$ con menos de 5 estados que acepte $L_1$.
   c) No podemos encontrar una expresión regular $ER$ para que $L(ER) = L_1$. V: $L_1$ no es regular
7. [ENERO2018] Sea la $GLC$ $G_1$ cuyas reglas son: $\{S \rightarrow aS \mid aA \mid \lambda, A \rightarrow aA \mid S \mid \lambda\}$, indica la respuesta **verdadera**:
   a) Es una gramática regular y ambígua.
   b) $G_1$ no es regular y, por tanto, no existe un $AF$ que acepte sus cadenas.
   c) Podemos generar la misma sentencia a partir de dos derivaciones a la derecha distintas. V: cadena aa
8. [ENERO2018] Sean dos $GLC$ $G_1$ y $G_2$, cuyas variable iniciales son $S_1$ y $S_2$, para la concatenación de ambas se requiere:
   a) Añadir una nueva regla $S \rightarrow S_1S_2$. V: definición
   b) Añadir las reglas $S_1 \rightarrow SS_2$ y $S_2 \rightarrow SS_1$.
   c) Añadir la regla $S \rightarrow S_1 \mid S_2$.
9. [ENERO2018] Sea $G_1$ una gramática libre de contexto propia. Podemos afirmar que:
   a) Podemos encontrar tanto una autómata de pila que acepta por estado final como por pila vacía. V: $G_1$ es GLC
   b) Al ser propia no encontraremos reglas unitarias.
   c) Al ser propia tiene un menor número de símbolos no terminales que una equivalente no propia.
10. [ENERO2018] Dado el lenguaje $L = \{ww^Rw\}$, indica la respuesta **verdadera**:
    a) Puede aceptarse por un $AP$ no determinista pero no un $AP$ determinista.
    b) Puede aceptarse por un $AP$ determinista y por tanto ser generado por una $GLC$.
    c) No puede aceptarse por un $AP$. V: $L$ no es libre de contexto

**Parte II: PROBLEMAS.** Total 7 puntos.

1. $\boxed{\text{2 puntos}}$ Sean los lenguajes $L_1$ y $L_2$ definidos como:

   - $L_1$: cadenas de aes y bes cuyo prefijo y sufijo es $ab$.
   - $L_2$: cadenas de aes y bes que, al menos, contienen una subcadena $ab$.

   *a*) **(0.5p)** Define formalmente los lenguajes $L_1$ y $L_2$ (por comprensión y/o usando operadores de lenguajes).

   *b*) **(1.0p)** Diseña un $AFD$ que acepte el lenguaje $\overline{L_1}$.

   *c*) **(0.5p)** Diseña una $ER$ que defina el lenguaje $L_1 \cap L_2$

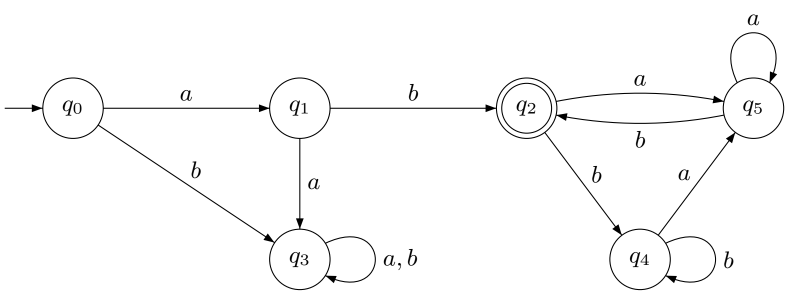
**SOLUCION:**

a)

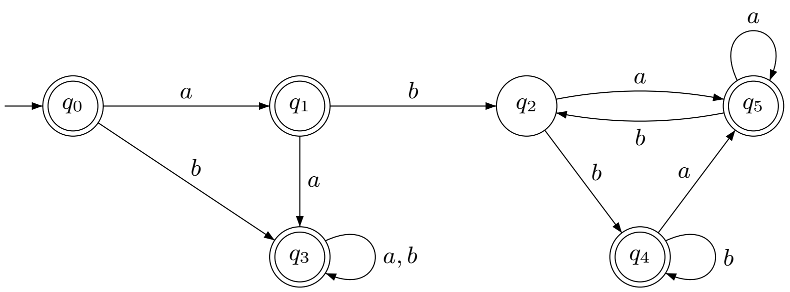
$L_1 = \{ab\} \cup \{abwab \mid w \in \{a,b\}^*\}$
$L_2 = \{xaby \mid x, y \in \{a,b\}^*\}$

b)

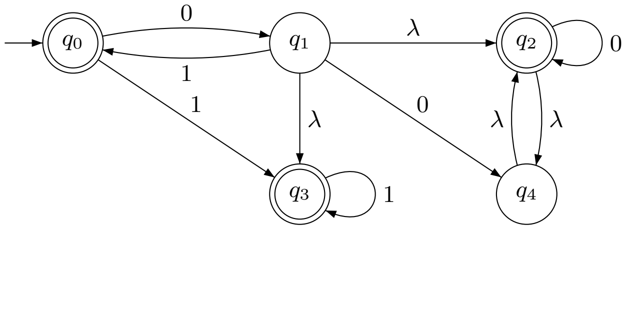
Se diseña el autómata que acepta el lenguaje $L_1$:




Es un autómata completo, y podemos complementarlo para obtener el de $\overline{L_1}$:




2. $\boxed{\text{2 puntos}}$ Dado el siguiente autómata $M_2$, obtén el **autómata finito determinista** equivalente utilizando el algoritmo visto en clase.




c)

Teniendo en cuenta que $L_1 \cap L_2 = L_1$, la expresión regular es $ER = ab(\lambda|(a|b)^*ab)$.

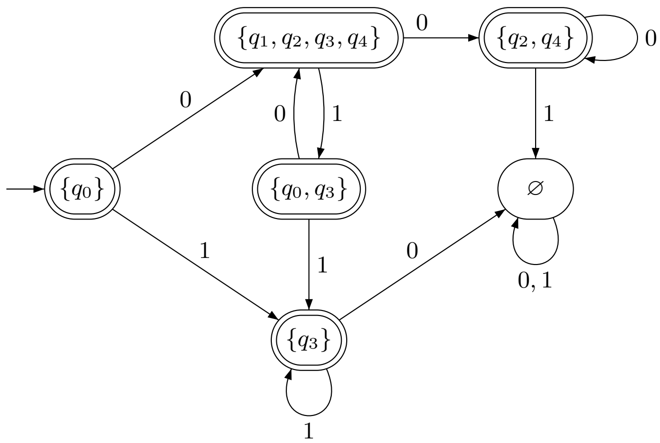
**SOLUCION:**

Calculamos la función de lambda-clausura (LC):

- $LC(q_0) = \{q_0\}$
- $LC(q_1) = \{q_1, q_2, q_3, q_4\}$
- $LC(q_2) = \{q_2, q_4\}$
- $LC(q_3) = \{q_3\}$
- $LC(q_4) = \{q_2, q_4\}$

Aplicamos el resto del algoritmo para calcular la función de transición y nuevos estados. El resultado es el siguiente AFD:

| | 0 | 1 |
|---|---|---|
| $\rightarrow \#\{q_0\}$ | $\{q_1, q_2, q_3, q_4\}$ | $\{q_3\}$ |
| $\#\{q_1, q_2, q_3, q_4\}$ | $\{q_2, q_4\}$ | $\{q_0, q_3\}$ |
| $\#\{q_3\}$ | $\varnothing$ | $\{q_3\}$ |
| $\#\{q_2, q_4\}$ | $\{q_2, q_4\}$ | $\varnothing$ |
| $\#\{q_0, q_3\}$ | $\{q_1, q_2, q_3, q_4\}$ | $\{q_3\}$ |
| $\varnothing$ | $\varnothing$ | $\varnothing$ |




3. $\boxed{\text{2 puntos}}$ Sea la gramática $G_3$ con las siguientes reglas de producción:

$$\begin{aligned} S &\rightarrow aA \mid cC \mid AC \\ A &\rightarrow aaA \mid \lambda \\ B &\rightarrow bB \mid bbA \\ C &\rightarrow cC \mid D \\ D &\rightarrow \lambda \end{aligned}$$

*a*) **(0.5p)** Define formalmente el lenguaje $L(G_3)$ (por comprensión y/o usando operadores de lenguajes).

*b*) **(1.5p)** Obtén una gramática $\lambda$-libre equivalente a $G_3$ usando el algoritmo visto en clase.

**SOLUCION:**

a)

El lenguaje $L(G_3)$ se puede considerar la unión de tres lenguajes, uno por cada una de las reglas de $S$:

$$L(G_3) = L_1 \cup L_2 \cup L_3$$

donde $L_1 = \{a^{2i+1} \mid i \geq 0\}$, $L_2 = \{c^j \mid j \geq 1\}$ y $L_3 = \{a^{2i}c^j \mid i, j \geq 0\}$.

b)

Aunque no es necesario, se puede hacer un paso previo de eliminación de símbolos inútiles. La variable B es inalcanzable, de modo que podemos trabajar con el resto de la gramática:

$$\begin{aligned} S &\rightarrow aA \mid cC \mid AC \\ A &\rightarrow aaA \mid \lambda \\ C &\rightarrow cC \mid D \\ D &\rightarrow \lambda \end{aligned}$$

Las variables anulables son $V_{anu} = \{A, D, C, S\}$. Eliminamos las $\lambda$-reglas:

$$\begin{aligned} S &\rightarrow aA \mid cC \mid AC \\ A &\rightarrow aaA \\ C &\rightarrow cC \mid D \end{aligned}$$

A continuación se modifica la gramática para que los símbolos anulables aparezcan o se eliminen en el lado derecho de las reglas, en todas las combinaciones posibles, pero sin añadir ninguna $\lambda$-regla salvo para $S$:

$$
\begin{aligned}
&S \to aA \mid a \mid cC \mid c \mid AC \mid C \mid A \mid \lambda \\
&A \to aaA \mid a \\
&C \to cC \mid c \mid D
\end{aligned}
$$

La gramática anterior ya es $\lambda$-libre. Podría darse un último paso para eliminar el símbolo inútil $D$ (improductiva):

$$
\begin{aligned}
&S \to aA \mid a \mid cC \mid c \mid AC \mid C \mid A \mid \lambda \\
&A \to aaA \mid aa \\
&C \to cC \mid c
\end{aligned}
$$

4. $\boxed{\text{1 punto}}$ Dado el lenguaje $L_4 = \{a^{2n}b^{n+1} \mid n > 0\}$, diseña un autómata de pila **determinista** con aceptación por **estado final** que acepte $L_4$.

**SOLUCION:**