EJERCICIOS del TEMA 2: Lenguajes Regulares

Sobre AFDs (autómatas finitos deterministas):

1. Razona la veracidad o falsedad de la siguientes afirmación, apoyándote en la teoría vista en clase. Si M = (Q, Σ, δ, q_0, F) es un autómata finito determinista totalmente especificado con F = Q, entonces L(M) = Σ^*.

2. Sea M = (Q, Σ, δ, q_0, F) un AFD con Q = {q_0,q_1,q_2}, Σ = {**a**, **b**}, F = {q_2} y la función de transición δ:

δ	**a**	**b**
q_0	q_0	q_1
q_1	q_2	q_1
q_2	q_2	q_0

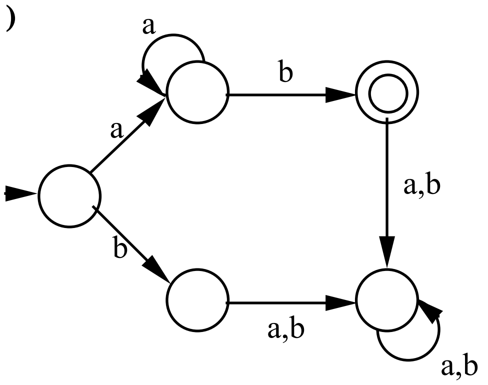
a) Dibuja el autómata M

b) Traza los cómputos de M que procesan las palabras **abaa**, **bbbabb**, **bababa** **bbbaa**

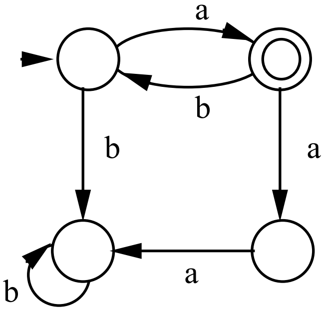
c) ¿Qué palabras de las procesadas en **(b)** son aceptadas por M?

3. Busca tres palabras aceptadas y tres palabras rechazadas por cada uno de los siguientes autómatas mostrando el cómputo que las procesa. Determina cuáles de ellos están totalmente especificados. ¿Sabrías cuál es el lenguaje aceptado por cada uno de ellos?

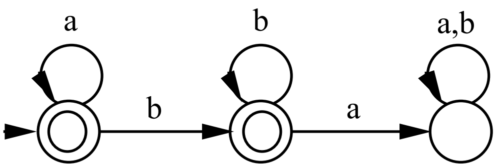
a)



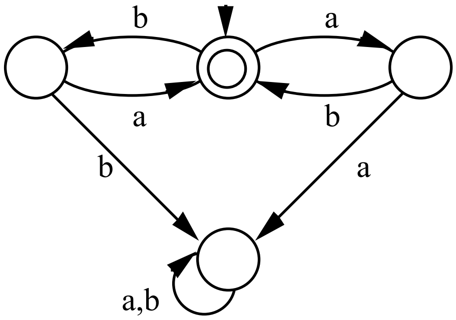
b)



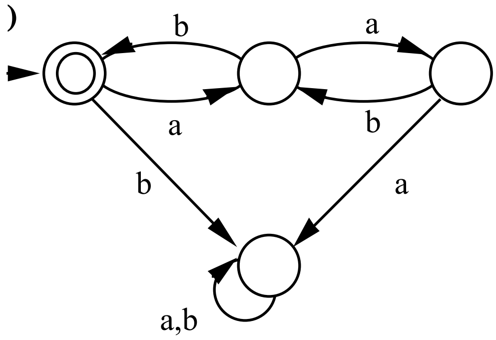
c)



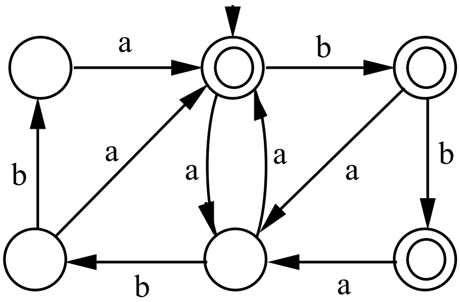
d)



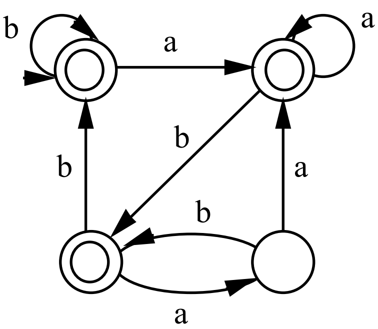
e)



f)



g)



4. Construye AFDs que acepten cada uno de los siguientes lenguajes definidos sobre el alfabeto $\Sigma = \{\mathbf{a},\mathbf{b}\}$:

- **a)** L = { $\mathbf{x} \in \Sigma^*$: la longitud de **x** es divisible por 3}
- **b)** L = { $\mathbf{x} \in \Sigma^*$: **aba** no es subpalabra de **x** }
- **c)** L = { $\mathbf{x} \in \Sigma^*$: **x** comienza por **a** y termina por **ab** }
- **d)** L = { $\mathbf{x} \in \Sigma^*$: **x** tiene un número par de **a**'s y un número par de **b**'s }
- **e)** L = { $\mathbf{x} \in \Sigma^*$: **x** tiene tres **a**'s consecutivas }
- **f)** L = { $\mathbf{x} \in \Sigma^*$: toda aparición de la subpalabra **aba** en **x**, o bien va seguida de **bb**, o está al final de la palabra }
- **g)** L = { $\mathbf{x} \in \Sigma^*$: si **x** empieza por **a** no contiene la subpalabra **aa** y si **x** empieza por **b** contiene la subpalabra **aa** }
- **h)** L = { $\mathbf{x} \in \Sigma^*$: **x** tiene un número par de apariciones de la cadena **ab** }
- **i)** L = { $\mathbf{x} \in \Sigma^*$: **ab** es subpalabra de **x** si y sólo si **ba** es subpalabra de **x** }
- **j)** L = { $\mathbf{x} \in \Sigma^*$: **x** está formada por la concatenación de un número arbitrario de cadenas de la forma $\mathbf{yy^R}$, con $|\mathbf{y}|=2$ }
- **k)** L = { $\mathbf{x} \in \Sigma^*$: **x** no contiene ningún prefijo en el que la diferencia entre el número de **a**'s y **b**'s sea mayor que tres (a favor de cualquiera de ellos) }

5. (**Ejercicio especial**) Construye un AFD que acepte el siguiente lenguaje: L = { **x** ∈ {**a**, **b**}* : **x** no contiene la subpalabra **bab** ni el sufijo **aba** }

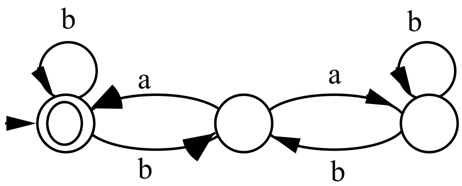
6. (**Ejercicio especial**) Demuestra que la noción de lenguaje aceptado por un AFD no depende del alfabeto particular de éste (siempre que al menos contenga los símbolos involucrados). Es decir, prueba que:

> "Si Σ* ⊃ L y ∏ ⊃ Σ, entonces existe un AFD sobre el alfabeto Σ que acepta L si y solo si, existe otro AFD sobre el alfabeto ∏ que acepta L".

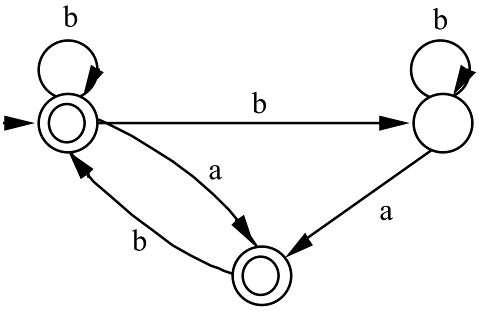
Sobre AFNDs (autómatas finitos no deterministas):

7. Busca tres palabras aceptadas y tres palabras rechazadas por cada uno de los siguientes autómatas no deterministas, mostrando todos los cómputos que las procesan. ¿Sabrías qué lenguaje acepta cada uno de ellos?

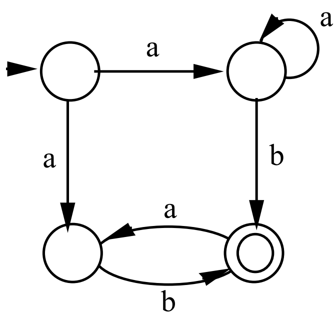
a)



b)



c)



8. Construye AFNDs que acepten los siguientes lenguajes sobre el alfabeto Σ = {**a**,**b**,**c**}

a) L = { **x** ∈ Σ* : **x** tiene algún par de **a**'s separadas por una cadena de símbolos de longitud 4*i, con i ≥ 0 }

b) L = { **x** ∈ Σ* : |**x**| ≥ 5 y el quinto símbolo contado desde el final es **a** }

c) L = { **x** ∈ Σ* : ni **aa** ni **bb** son subcadenas de **x** }

d) L = { **x** ∈ Σ* : **x** tiene **ab** y **ba** como subcadena }

e) L = { **x** ∈ Σ* : **ccc** es sufijo de **x** y en ninguna otra posición de **x** pueden encontrarse dos símbolos iguales seguidos }

f) L = { **x** $\in \Sigma^*$: **x** tiene tres símbolos iguales seguidos }

g) L = { **x** $\in \Sigma^*$: los símbolos de **x** de posición múltiplo de tres son **c**'s }

9. (**Ejercicio especial**) Construye un AFND que acepte el siguiente lenguaje L = { **x** $\in$ {**a**, **b**, **c**}* : **x** empieza y termina por **a** y entre cada aparición de **a** y la siguiente hay un número par de **b**'s o un número impar **c**'s }

10. (**Ejercicio especial**) ¿Cuántos autómatas finitos deterministas y con dos estados pueden construirse sobre el alfabeto {**0**,**1**}? ¿Aceptan todos ellos lenguajes distintos? ¿Y de ellos, cuántos están totalmente especificados? ¿Y si construimos autómatas finitos no deterministas?

11. Sea el lenguaje regular L = { x $\in$ {**a**,**b**}*: $|x|_{\mathbf{a}}$ **mod** 2 = 1 $\wedge$ $|x|_{\mathbf{b}}$ **mod** 3 = 0 $\wedge$ **aba** es subpalabra de x }. Sigue los pasos que se te indican a continuación *sin construir en ningún momento el autómata finito indicado* :

a) Supón que quieres construir un autómata finito M que reconozca a L. Indica cuántos estados tendría M y explica por qué. Enumera dichos estados, explicando para qué sirve cada uno de ellos en tu construcción. Señala cuál sería el estado inicial y cuáles los finales.

b) Elige tres estados cualesquiera e indica cuáles serían sus transiciones y a qué estados nos llevarían.

c) Reconstruye los cómputos asociados a las palabras **aababb** y **babab**.

Sobre GRs (gramáticas regulares) y GLs (gramáticas lineales):

12. Construye gramáticas regulares ó lineales a la derecha que generen cada uno de los siguientes lenguajes sobre el alfabeto terminal Σ = {**a**, **b**, **c**}

a) { **x** $\in \Sigma^*$: $|\mathbf{x}|_{\mathbf{a}}$ mod 2 = 0 }

b) { **x** $\in \Sigma^*$: **x** empieza por **a** y contiene la subpalabra **bbb** }

c) { **x** $\in \Sigma^*$: **x** no contiene tres **b**'s consecutivas }

d) { **x** $\in \Sigma^*$: $|\mathbf{x}|_{\mathbf{a}} + |\mathbf{x}|_{\mathbf{b}}$ mod 3 = 0 }

e) { **x** $\in \Sigma^*$: **x** contiene las subpalabras **aa**, **bb** y **cc** }

f) { **x** $\in \Sigma^*$: **x** no contiene las subpalabras **aba** ni **aca** }

g) { **x** $\in \Sigma^*$: cada aparición de **a** en **x** es precedida de **b** o seguida de **c** }

h) { **x** $\in \Sigma^*$: **x** no contiene ninguna aparición de la subcadena **aa** después de la última aparición del símbolo **c** }

i) { **x** $\in \Sigma^*$: **x** contiene dos **a**’s separadas por un número impar de símbolos }

j) { **x** $\in \Sigma^*$: **x** contiene la subpalabra **ab** pero no contiene la subpalabra **aba** }

De entre ellas, di cuales son regulares y cuales son lineales. Transforma las lineales en regulares.

13. (**Ejercicio especial**) Sea G=(N, Σ, P, S) una gramática lineal a la derecha, con N={S,A,B} y Σ = {**a** ,**b** }, y sean las siguientes palabras: S, A, AB, ε, **aa**A, **aba**S, **bbab**, **a**S**ba**, B**aaab**, **ab**AS, **ab**BA**b**S**b**.

a) ¿Cuáles de ellas podrían ser formas sentenciales de G?

b) Describe la estructura general de una forma sentencial de G.

c) Demuestra por inducción que las formas sentenciales de G no pueden ser distintas de lo descrito en el apartado anterior.

14. (**Ejercicio especial**) Una Gramática cuasi-regular a la derecha se define como una gramática G = (N, Σ, P, S), donde las reglas de P responden a alguna de las 3 siguientes formas: $A \to \varepsilon$, $A \to \mathbf{a}B$, $A \to \mathbf{a}$, con $A,B \in N$, $\mathbf{a} \in \Sigma$. Considera el siguiente algoritmo:

<u>Entrada</u>**:** G = (N, Σ, P, S) gramática cuasi-regular a la derecha

<u>Salida</u>**:** INCOGNITA

<u>Procedimiento</u>**:** Construir INCOGNITA = (N_1, Σ, δ, S, F) donde

$N_1 = N \cup \{Z\}$

$F = \{Z\} \cup \{A : A \to \varepsilon \in P\}$

$B \in \delta(A,\mathbf{a}) \Leftrightarrow A \to \mathbf{a}B \in P$

$Z \in \delta(A,\mathbf{a}) \Leftrightarrow A \to \mathbf{a} \in P$

a) Aplica el algoritmo a la gramática siguiente: G=({S,A,B}, {**a**,**b**,**c**}, P, S) siendo P el conjunto de producciones

$S \to \mathbf{a}A \mid \mathbf{a}B$ $\quad A \to \mathbf{b}A \mid \mathbf{b}$ $\quad B \to \mathbf{c}B \mid \mathbf{c}$

b) ¿A qué modelo pertenece INCÓGNITA?. Razona tu respuesta.

c) Justifica la siguiente afirmación: "Todo lenguaje generado por una gramática regular (a derecha) se puede generar mediante una gramática cuasi-regular (a derecha) y viceversa".

Sobre ERs (expresiones regulares):

15. Di si son verdad o no las siguientes afirmaciones:

a) **baa** $\in$ L(**a*****b*****a*****b***)

b) L(**b*****a***) $\cap$ L(**a*****b***) = L(**a*** $\cup$ **b***)

c) L(**a*****b***) $\cap$ L(**c*****d***) = $\varnothing$

d) **abcd** $\in$ L((**a**(**cd**)***b**)*)

e) Las expresiones regulares $\alpha = \boldsymbol{\varepsilon}$ y $\beta=(\boldsymbol{\varepsilon}\cup\boldsymbol{\varnothing})^*$ son equivalentes.

16. Para las siguientes expresiones regulares, escribe todas las palabras de longitud menor o igual que seis pertenecientes al lenguaje que generan, y describe dicho lenguaje en cada caso:

a) (**10**)* $\cup$ (**01**)*

b) (**10** $\cup$ **01**)*

c) (**11** $\cup$ **0**)*(**00** $\cup$ **1**)*

d) (**1** $\cup$ **01** $\cup$ **001**)*(ε $\cup$ **0** $\cup$ **00**)

e) [**00** $\cup$ **11** $\cup$ (**01** $\cup$ **10**)(**00** $\cup$ **11**)*(**01** $\cup$ **10**)]*

17. Sea Σ ={**a**,**b**}. Escribe expresiones regulares para los siguientes lenguajes:

a) cadenas con al menos tres **a**'s

b) cadenas con a lo sumo tres **a**'s

c) cadenas con un número de **a**'s divisible por 3

d) cadenas con al menos una aparición de la subcadena **aaa**

e) cadenas en las que las **a**'s van agrupadas como mínimo de tres en tres.

18. Sea Σ ={**a**,**b**}. Construye una expresión regular para el lenguaje formado por las cadenas que <u>no contienen</u> la subcadena **aaa.** Se debe hacer primero el AFD, obtener a partir de él la GRD y, resolviendo las ecuaciones correspondientes, llegar a la expresión regular.

19. (**Ejercicio especial**) Sea Σ ={**a**,**b**}. Construye una expresión regular para el lenguaje formado por las cadenas que contienen <u>exactamente una</u> aparición de la subcadena **aaa.** Se debe hacer primero el AFD, obtener a partir de él la GRD y, resolviendo las ecuaciones correspondientes, llegar a la expresión regular.

20. Construye expresiones regulares para denotar los siguientes lenguajes:

a) { $\mathbf{w} \in \{\mathbf{a}, \mathbf{b}\}^*$: **w** acaba en **ab** }

b) { $\mathbf{w} \in \{\mathbf{a}, \mathbf{b}\}^*$: $|\mathbf{w}|_{\mathbf{a}} \neq 1$ } .

c) { $\mathbf{w} \in \{\mathbf{a}, \mathbf{b}\}^*$: **w** tiene un número par de **a**'s y termina por **ab** }

d) { $\mathbf{w} \in \{1,0\}^*$: **w** no contiene **101** como subpalabra }

e) { $\mathbf{w} \in \{1,0\}^*$: **w** comienza por **101** y termina por **101** }

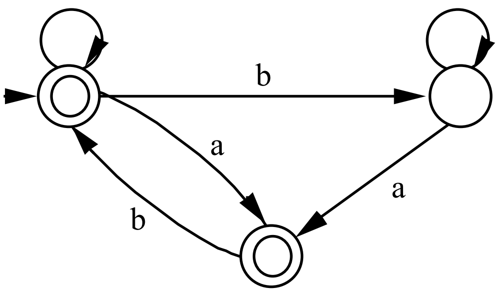
f) { $\mathbf{w} \in \{\mathbf{a}, \mathbf{b}, \mathbf{c}\}^*$: entre cada dos **a**'s de **w** hay un número de **c**'s múltiplo de 3}

21. En la definición inductiva de las expresiones regulares, la regla que afirma que ε es una expresión regular se puede eliminar, ya que con el resto de las reglas es posible construir una expresión α tal que L(α) = {ε}. ¿Por qué?

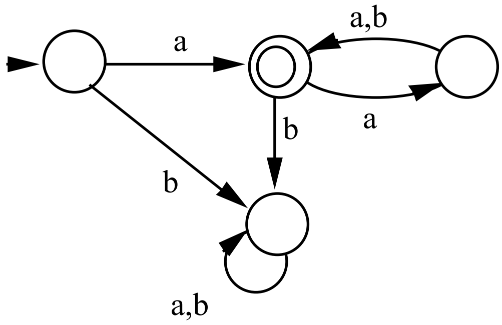
Sobre ERs equivalentes a autómatas:

22. Construye expresiones regulares equivalentes a los siguientes AF's:

a)



b)



Sobre ERs equivalentes a gramáticas:

23. Construye expresiones regulares equivalentes a las gramáticas lineales a la derecha cuyas reglas de producción figuran a continuación:

a)
S → **a**S | **b**S | **aba**A | **ababa**B | **abababa**C | **abababa**
A → **a**A | **b**A | **aba**B | **ababa**C | **ababa**
B → **a**B | **b**B | **aba**C | **aba**
C → **a**C | **b**C | **a** | **b**

b)
S → **a** | **a**A | **b**B
A → **b** | **b**A
B → **a** | **a**A | **a**C
C → **b** | **b**A | **a**D
D → **a**C | **b**D

Ejercicios teóricos sobre ERs:

24. (**Ejercicio especial**) Asumamos como ya demostrado que para cualesquiera expresiones regulares $\alpha_1, ..., \alpha_n, \beta_1, ..., \beta_n$ se cumple que:

i) $(\alpha_1 \cup \alpha_2 \cup ... \cup \alpha_n)^* = (\alpha_1{}^*\alpha_2{}^* ... \alpha_n{}^*)^*$

ii) $(\alpha_1 \cup \alpha_2 \cup ... \cup \alpha_n)\ (\beta_1 \cup \beta_2 \cup ... \cup \beta_m) = \alpha_1\beta_1 \cup \alpha_1\beta_2 \cup ... \cup \alpha_1\beta_m \cup \alpha_2\beta_1 \cup ... \cup \alpha_n\beta_1 \cup ... \cup \alpha_n\beta_m$

Se dice que una expresión regular está en *forma normal disyuntiva* si tiene la forma $\alpha_1 \cup \alpha_2 \cup ... \cup \alpha_n$, con $n \geq 1$ y $\alpha_1, ..., \alpha_n$ son expresiones regulares en las que no aparece el símbolo "$\cup$".

a) Utilizando los resultados anteriores **i)** y **ii)** demuestra que para toda expresión regular α existe otra equivalente **fnd(α)** en forma normal disyuntiva.

b) Escribe la forma normal disyuntiva de las siguientes expresiones regulares:

$(\varepsilon \cup (0 \cup 1)^*100)0^*$

$((10^* \cup 01^*)\ (00^* \cup 11^*))^*$

25. Sea α una expresión regular cualquiera:

a) Diseña un algoritmo recursivo para decidir si α es equivalente a la expresión regular $\varnothing$.

b) Describe otro algoritmo para obtener a partir de α otra expresión regular equivalente **sinvac(α)** que, o bien no contiene la subexpresión $\varnothing$, o es exactamente la expresión regular $\varnothing$ (inevitable cuando α es equivalente a $\varnothing$).

c) Aplica el algoritmo hecho en el apartado **(b)** a la expresión regular: $(1\ \varnothing\ (11)^*)^*(0\ \varnothing\ 0 \cup 01)^*(\varnothing\ (1 \cup 0)^* \cup 10)$.

26. Sea α una expresión regular que no contiene ninguna aparición de la subexpresión $\varnothing$.

a) Diseña un algoritmo recursivo para decidir si la palabra vacía pertenece o no a $L(\alpha)$.

b) Describe otro algoritmo para obtener a partir de α otra expresión regular equivalente **sineps(α)**, que sigue sin contener apariciones de $\varnothing$, y que tiene la forma ε, β ó $\beta \cup \varepsilon$, donde β es una expresión regular que no contiene apariciones de ε (es decir, en **sineps(α)** eliminamos todas las apariciones de ε salvo una en caso de que sea estrictamente necesaria para mantener la equivalencia).

c) Aplica el algoritmo encontrado en **(b)** a la expresión regular $(((\mathbf{1} \cup \boldsymbol{\varepsilon})^*\boldsymbol{\varepsilon})\, \mathbf{01} \cup \boldsymbol{\varepsilon}\, \mathbf{1} \cup \boldsymbol{\varepsilon})^*\, (\mathbf{01} \cup \boldsymbol{\varepsilon})\, (\mathbf{10} \cup \boldsymbol{\varepsilon}) \cup \boldsymbol{\varepsilon}^*$.

d) Supongamos ahora que, en la definición de expresión regular, se suprimen las claúsulas que aseguran que $\varnothing$ y $\boldsymbol{\varepsilon}$ son expresiones regulares. A la luz de los resultados de este ejercicio y el precedente ¿cuáles serían los lenguajes regulares que no podrían ser denotados por las mismas? ¿Y si sólo quitamos una de las claúsulas?

Sobre ε-AFND's y su relación con las ER's:

27. Construye ε-AFND's que acepten los siguientes lenguajes sobre $\Sigma = \{\mathbf{a}, \mathbf{b}, \mathbf{c}\}$

a) L = { **x** $\in \Sigma^*$: las apariciones de **a** en **x** no pueden ir seguidas de **c**, las apariciones de **b** en **x** no pueden ir seguidas de **a**, y las apariciones de **c** en **x** no pueden ir seguidas de **b** }

b) L = { **x** $\in \Sigma^*$: **x** no puede tener ninguna **a** ocupando una posición posterior a la de una **b** y **x** no contiene **c**'s aisladas }

c) L = { **x** $\in \Sigma^*$: **x** está formada por uno ó más grupos de **a**'s separados por cadenas de 1, 2 ó 3 **b**'s }

d) L = { **x** $\in \Sigma^*$: **x** puede descomponerse como concatenación de tres subpalabras **y**, **z** y **w** cumpliendo que en **y** cada grupo de **a**'s tiene longitud par, en **z** cada grupo de **b**'s tiene longitud par y en **w** cada grupo de **c**'s tiene longitud par }

28. Construye autómatas finitos que reconozcan los lenguajes denotados por las siguientes expresiones regulares:

a) $\mathbf{a^*bb^*(a \cup b)ab^*}$

b) $\mathbf{b((aab^* \cup a^4)b)^*a}$

c) $\mathbf{(a^+b^*a^+b^*)^+}$

d) $\mathbf{(((b^*a)^*a)^*a)^*a}$

e) $\mathbf{(a^2)^*(b^3)^*(c^4)^*(a^4)^*(b^3)^*(c^2)^*}$

29. (**Ejercicio especial**) **a)** Construye un **autómata**, M, que rechace todas las palabras de longitud impar que tengan los dos últimos símbolos iguales. Debe aceptar, por ejemplo, las palabras **abaa**, **a**, **aba** y rechazar **abb**, **babbbaaaa.** Puedes construirlo directamente (con las explicaciones oportunas) u obtenerlo del apartado siguiente mediante algún algoritmo de transformación.

b) Diseña una expresión regular que genere el mismo lenguaje del apartado anterior . Puedes diseñarla directamente (con las explicaciones oportunas) u obtenerla del apartado anterior mediante algún algoritmo de transformación.

Sobre equivalencias entre autómatas:

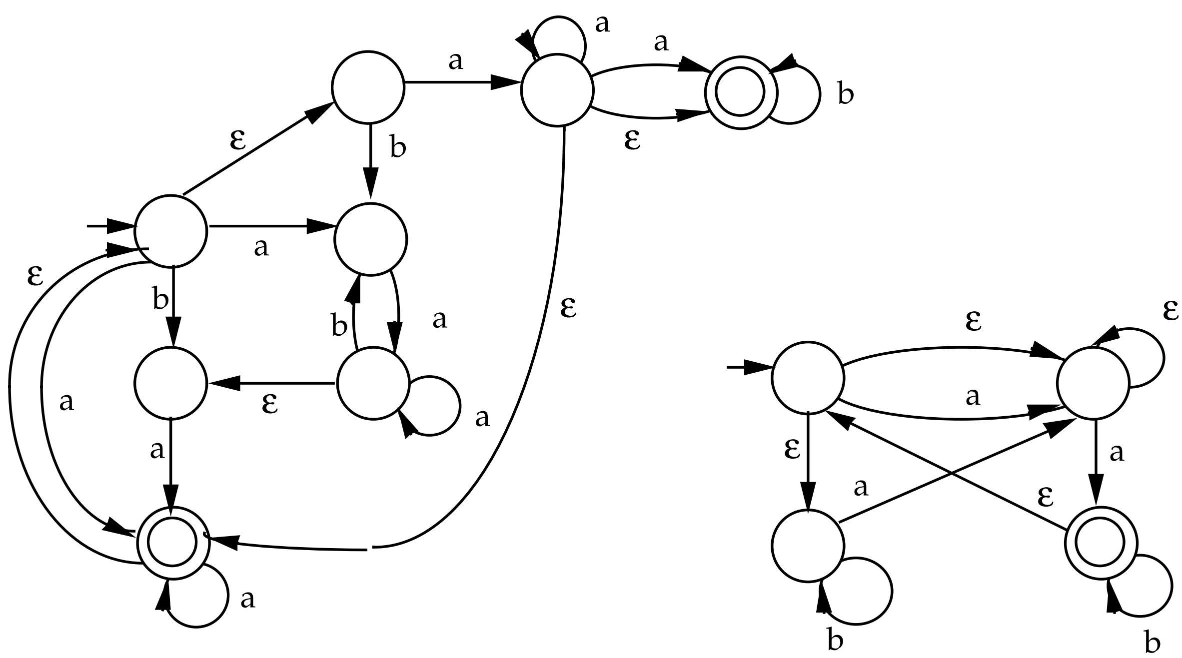
30. Construye AFD's equivalentes a los siguientes AFND's:

a) $(\{ q_0, q_1, q_2, q_3 \}, \{ \mathbf{0} , \mathbf{1} \}, \delta_1, q_0, \{ q_3 \})$

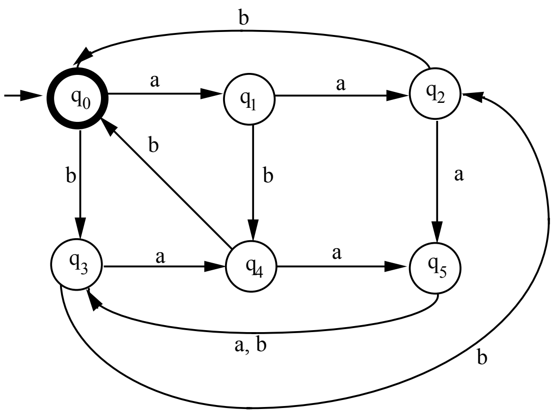
b) $(\{ q_0, q_1, q_2, q_3 \}, \{ \mathbf{0} , \mathbf{1} \}, \delta_2, q_0, \{ q_1,q_3 \})$

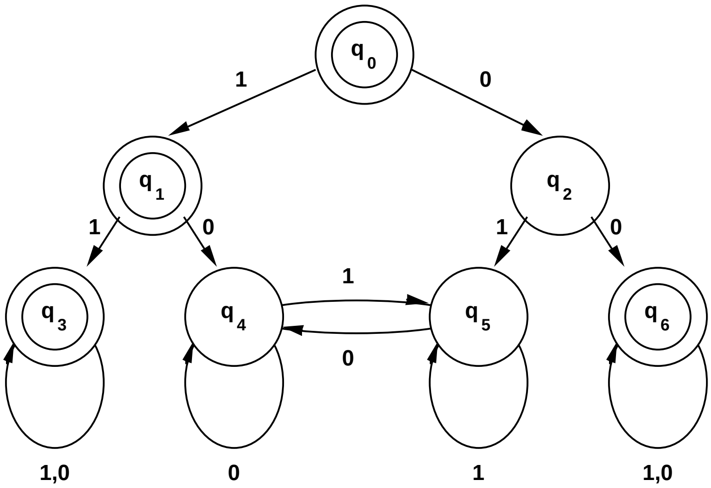
δ_1	**0**	**1**	δ_2	**0**	**1**
q_0	q_0,q_1	q_0	q_0	q_1,q_3	q_1
q_1	q_2	q_2	q_1	q_2	q_1,q_2
q_2	q_3	-	q_2	q_3	q_0
q_3	q_3	q_3	q_3	-	q_0

31. Construye AFD's equivalentes a los siguientes ε-AFND's:



32. Dado los siguientes autómatas finitos construye el autómata finito mínimo equivalente a cada uno de ellos:





33. Sea el lenguaje L = {$\mathbf{a}^{100+3i}\mathbf{b}^j$: i, j ≥ 0} definido sobre el alfabeto Σ = {**a**, **b**}.

a) Demuestra que L es regular, dando una expresión regular α que lo denote.

b) Describe como sería un AFD M, totalmente especificado, equivalente a α. Procura que sea el autómata mínimo. Toma un estado cualquiera de M, y comprueba que no es equivalente/indistinguible de ninguno de los demás estados.

Nota: No es necesario ni conveniente que apliques los algoritmos de transformación o minimización. De hecho no hace falta que escribas el autómata M completo.

34. Supón que te dan dos expresiones regulares α y β, y que te piden que compruebes si son o no equivalentes. Explica el proceso que seguirías para resolverlo.

35. Di si son ciertas o falsas las siguientes afirmaciones, razonando tu respuesta de forma breve pero convincente.

a) Tenemos un AFD sobre el alfabeto {**a**,**b**} que contiene dos estados no finales, p y q, con las siguientes transiciones:

	a	**b**
p	p	q
q	p	q

Sin conocer el resto del autómata podemos asegurar que estos estados son necesariamente indistinguibles.

b) Si M = (Q, Σ, δ, q_0, F) es el autómata mínimo equivalente al AFD, N = (P, Σ, γ, p_0, G), el cardinal de F y G ha de ser el mismo.

36. Demuestra que no son regulares sobre el alfabeto $\Sigma = \{ \mathbf{a} , \mathbf{b} \}$:

a) $\{ \mathbf{w}\overline{\mathbf{w}} \in \Sigma^*$: $\overline{\mathbf{w}}$ es la palabra que resulta de cambiar cada aparición en **w** de **a** por **b** y viceversa }

b) $\{ \mathbf{w} \in \Sigma^*: \mathbf{w} = \mathbf{w}^R \wedge |\mathbf{w}|_a = |\mathbf{w}|_b \}$

c) $\{ \mathbf{a}^n\mathbf{b}^m: m \le n \le 2*m \}$

d) $\{ \mathbf{a}^i\mathbf{b}^j\mathbf{a}^k: j = \max(i,k) \}$

e) $\{ \mathbf{a}^i\mathbf{b}^j\mathbf{a}^k : i, j, k >0 \wedge (i \le j \vee j \ge k \vee i = k) \}$

f) $\{ \mathbf{a}^i\mathbf{b}^j\mathbf{a}^k : i, j, k >0 \wedge (i = j \vee i = k \vee j = k) \}$

g) { $\mathbf{w} \in \Sigma^*$: **w** tiene al menos un prefijo con más **b**'s que **a**'s }

37. Considera el alfabeto { M, D, C, L, X, V, I } y el lenguaje de los números romanos. Demuestra que es un lenguaje regular construyendo un autómata finito con transiciones vacías que lo reconozca. Recuerda que, por ejemplo, VIIII no es un número romano, y que debemos escribir IX, en su lugar. Puede resultarte útil construir la expresión regular y luego el ε-AFND correspondiente.