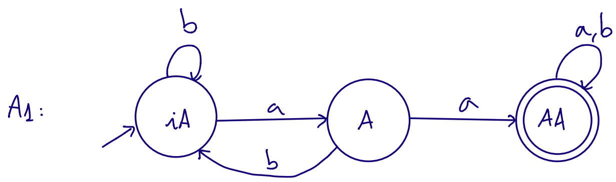
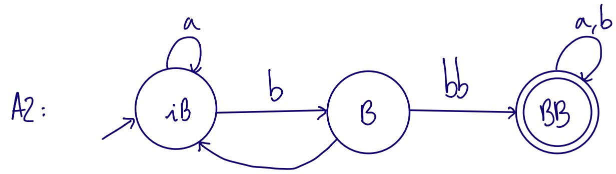
1. Obtén los DFA's mínimos A_1, A_2 para $L_1 = \{xaay \mid x, y \in \{a,b\}^*\}$ y $L_2\{xbby \mid x, y \in \{a,b\}^*\}$, respectivamente, y calcula explícitamente el DFA intersección $A_1 \cap A_2$.

A_1:



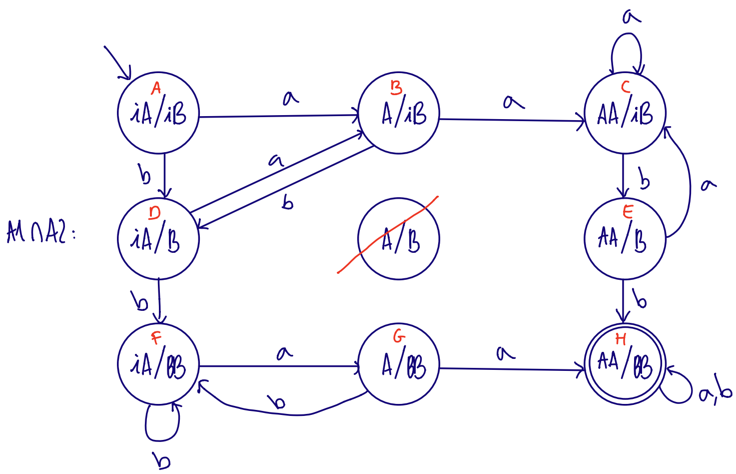
	a	b	
→ iA	A	iA	
A	AA	iA	
AA	AA	AA	+

A_2:



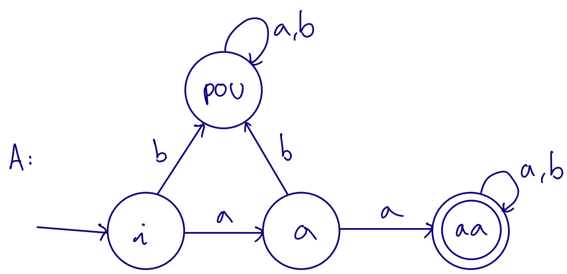
	a	b	
→ iB	iB	B	
B	iB	BB	
BB	BB	BB	+

$A_1 \cap A_2$:

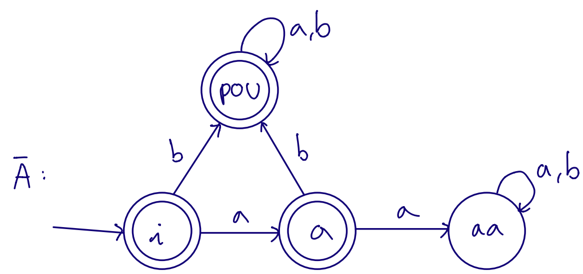


4. Obtén el DFA mínimo A para $L = \{aaw \mid w \in \{a,b\}^*\}$, y calcula explícitamente $\overline{A}$.

A:

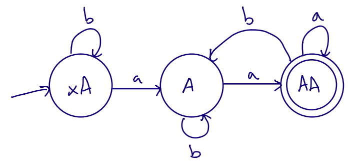


$\overline{A}$:

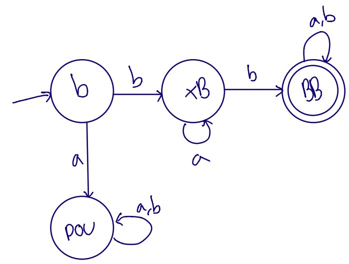


7. Obtén los DFA's mínimos A_1, A_2 para $L_1 = \{xaya \mid x, y \in \{a,b\}^*\}$ y $L_2 = \{bxby \mid x, y \in \{a,b\}^*\}$, respectivamente, y calcula explícitamente el λ-NFA concatenación $A_1 \cdot A_2$, determinízalo y minimízalo.

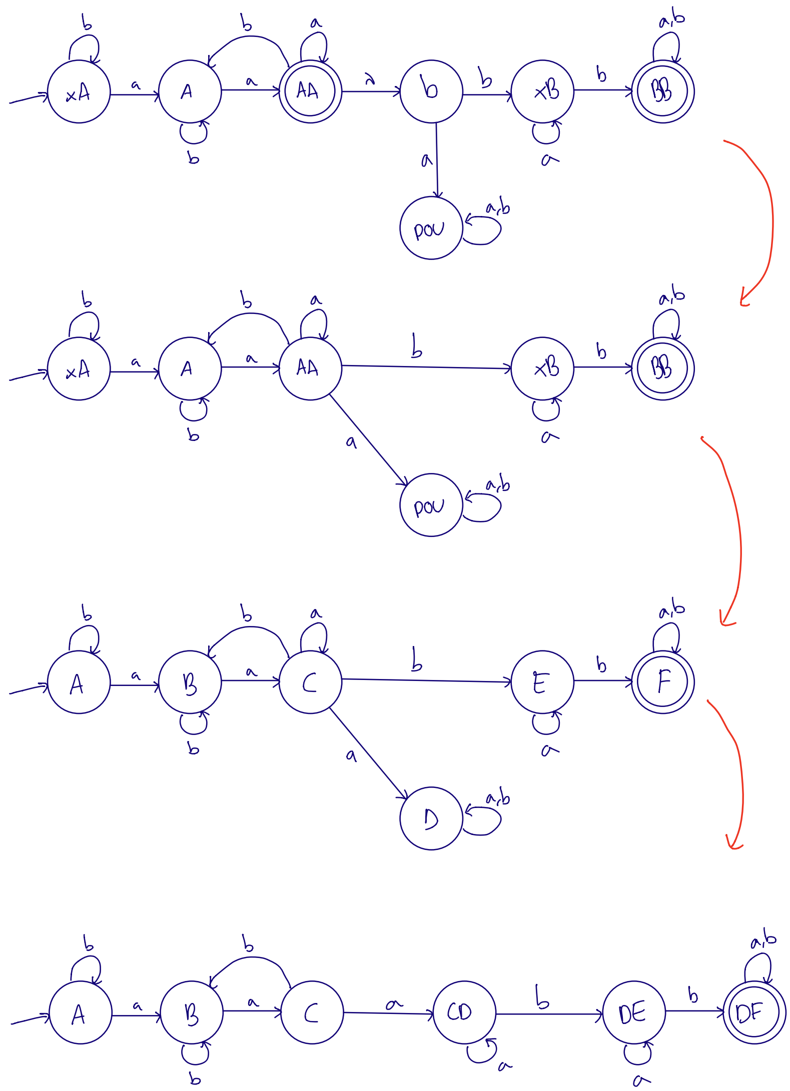
A1 :

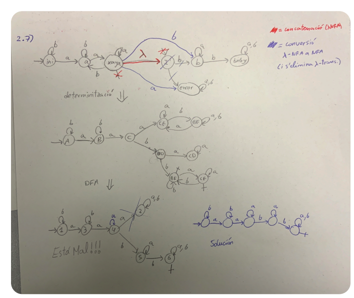


A2 :



A1·A2 :





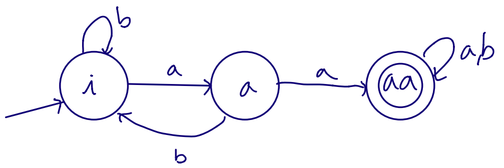
SOLUCION

	a	b
A	B	A
B	C	B
C	CD	B
CD	CD	DE
DE	DE	DF
DF	DF	DF

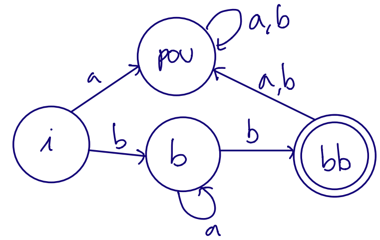
+

8. Obtén los DFA's mínimos A_1, A_2 para $L_1 = \{xaay \mid x, y \in \{a,b\}^*\}$ y $L_2 = \{bxb \mid x \in \{a,b\}^*\}$, respectivamente, y calcula explícitamente el NFA concatenación $A_1 \cdot A_2$, determinízalo y minimízalo.

A1.

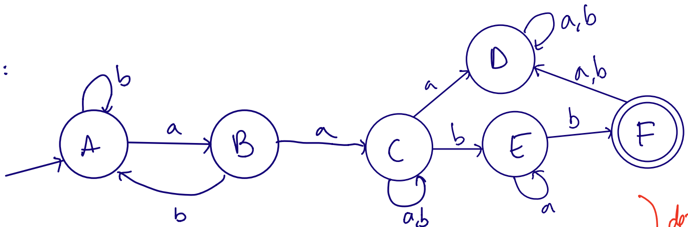


A2:



A1·A2:

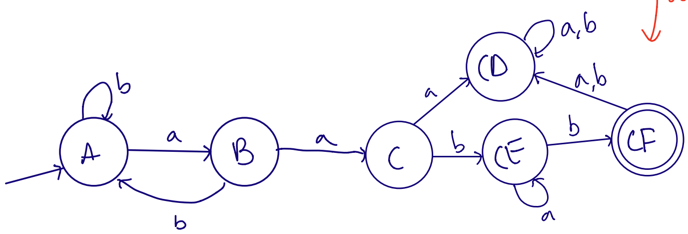
NFA

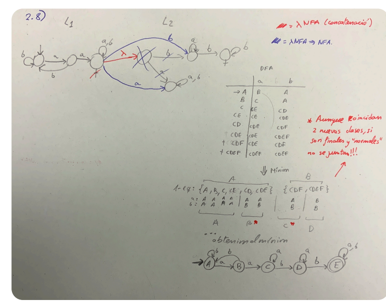


DFA	a	b
A	B	A
B	B	C
C	CD	CE
CD	CD	CD
CE	CE	CF
CF	CD	CD

determinístic

DFA

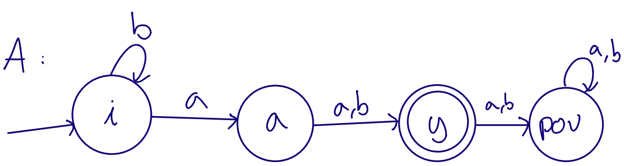




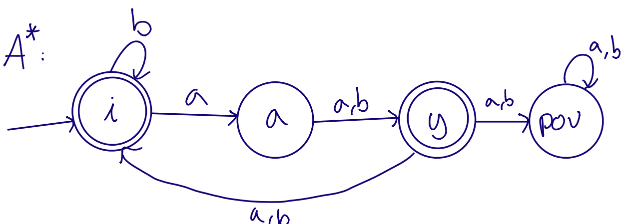
SOLUCION

10. Obtén el DFA mínimo A para $L = \{xay \in \{a,b\}^* \mid |y| = 1\}$, y calcula explícitamente el NFA estrella A^*, determinízalo y minimízalo.

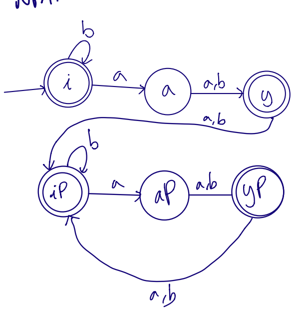
A:



A^*:

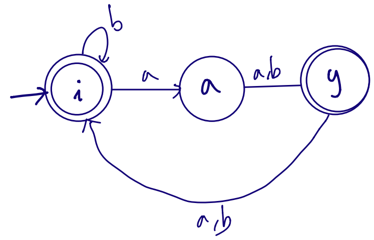


NFA:



	a	b	
i	a	i	+
a	y	y	
y	ipou	ipou	+
ipou	apou	ipou	+
apou	ypou	ypou	
ypou	ipou	ipou	+

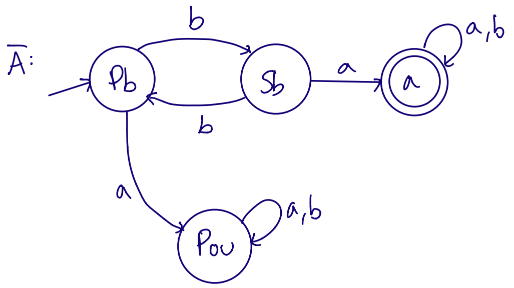
$\Longrightarrow$



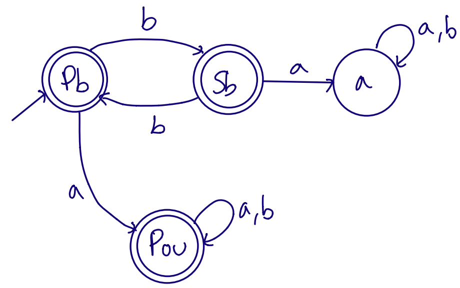
13. Obtén el DFA mínimo A para $L = \{w \in \{a,b\}^* \mid \forall w_1, w_2 (w = w_1 a w_2 \Rightarrow |w_1|_b \in \dot{2})\}$, y calcula explícitamente el NFA reverso A^R, determinízalo y minimízalo.

$\overline{L} = \{ w \in \{a,b\}^* \mid \exists_{w_1, w_2} \quad w = w_1 \cdot a \cdot w_2 \wedge |w_1|_b \in \dot{2}+1 \}$

$\overline{A}$:

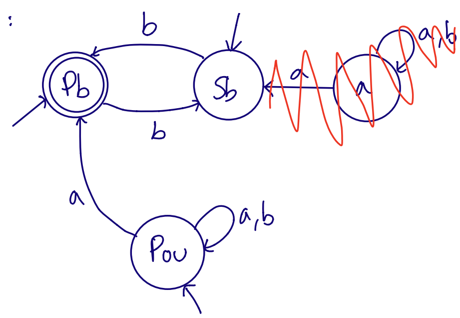


$\Longrightarrow$ A:



$\Downarrow$

A^R:



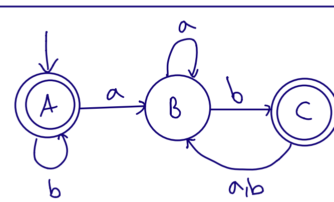
determine

	a	b	
→ PbSbPou	PbPou	PbSbPou	+
PbPou	PbPou	SbPou	+
SbPou	PbPou	PbPou	

|||

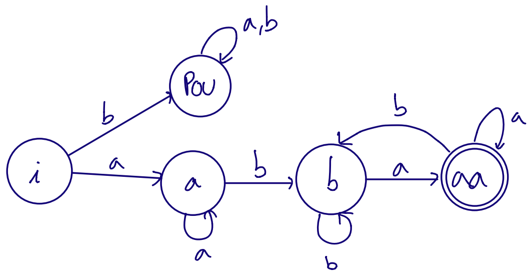
	a	b	
→ A	B	A	+
B	B	C	+
C	B	B	

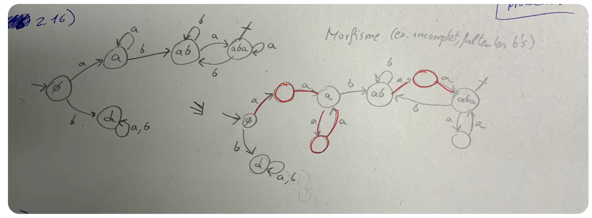
A^R: $\equiv$



16. Obtén el DFA mínimo A para $L = \{axbya \mid x, y \in \{a,b\}^*\}$, y dado el morfismo definido por $\sigma(a) = aa$, $\sigma(b) = ba$, calcula explícitamente el NFA imagen $\sigma(A)$, determinízalo y minimízalo.

A:





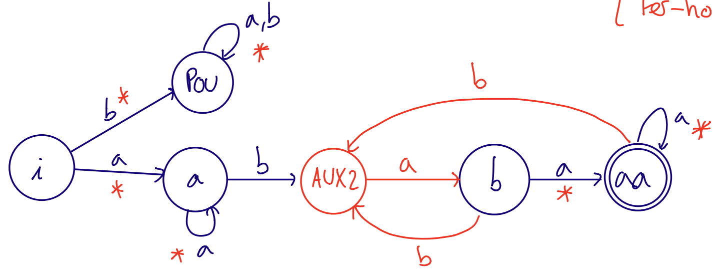
Solucion inacabada (morfismo aplicado solo a las A)

* [no cal més estats, desapareixerien al fer-ho mínim]

$\sigma(A)$:

$\sigma(a) = aa$

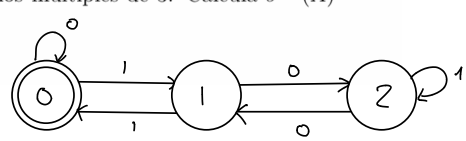
$\sigma(b) = ba$



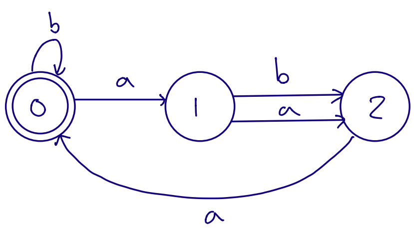
19. Sea A el DFA mínimo que reconoce a los números binarios múltiples de 3. Calcula $\sigma^{-1}(A)$ tomando como σ los morfismos:

(a) $\sigma(a) = 01,\ \sigma(b) = 0,\ \sigma(c) = \lambda$.

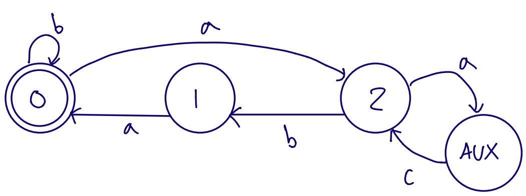
A:



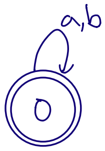
$\sigma^{-1}(A)$:



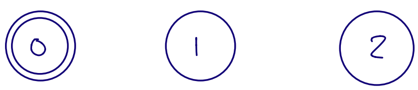
(b) $\sigma(a) = 10,\ \sigma(b) = 0,\ \sigma(c) = \lambda$.

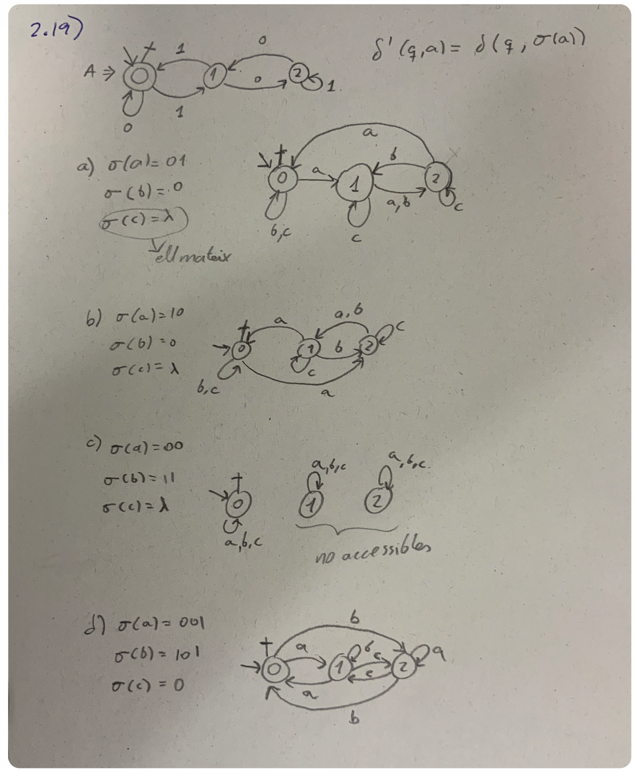


(c) $\sigma(a) = 00,\ \sigma(b) = 11,\ \sigma(c) = \lambda$.



(d) $\sigma(a) = 001,\ \sigma(b) = 101,\ \sigma(c) = 0$.





Solucion

20. Diseña un algoritmo de coste razonable para encontrar los estados no accesibles de un DFA de entrada.

2.20) Mètode DFS: $M = \langle Q, \Sigma, \delta, q_0, F \rangle$ [Considerem M com a graf dirigit]

$Q = \{q_0, q_1, \dots, q_n\}$

```
accessibles = vector<estats>(0);
S una pila;
vector<bool> visited(n, false);
for (q0 --> qn, anomenem qi al tractat){
    s.push(qi);
    while (not s.empty()){
        qj = s.top();
        s.pop();
        if (not visited[qj]){
            visited[qj] = true;
            s.push(estats adj. a qj)
            accessibles.insert(qj)
        }
    }
}
estats_no_accesibles = Q - accessibles;
```

cost = $O(n+m)$

n = nº estats

m = nº transicions

porque quiero saber los que son del comp. conex de q_0!!! Los no conexos no son accesibles.

Solucion

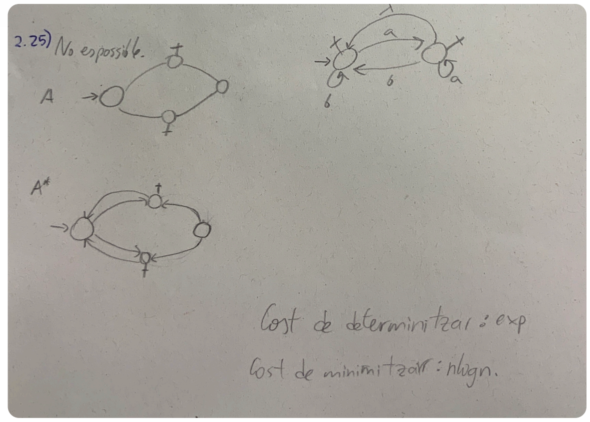
21. Diseña un algoritmo de coste razonable para decidir si un DFA de entrada acepta alguna palabra.

22. Diseña un algoritmo de coste razonable para decidir si un DFA de entrada acepta infinitas palabras.

24. Cuál es el coste temporal de las siguentes operaciones de DFA's:

(a) intersección.
(b) unión.
(c) complementario.
(d) concatenación (incluyendo determinización).
(e) estrella (incluyendo determinización).
(f) reverso (incluyendo determinización).
(g) aplicación de morfismo (incluyendo determinización).
(h) aplicación de morfismo inverso.

25. Por qué en nuestra definición de A^* añadimos un nuevo estado inicial, que es también aceptador, en el caso particular en que se cumpla $\lambda \notin \mathcal{L}(A)$? En este caso, no bastaría con poner el propio estado inicial de A como aceptador?



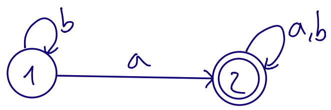
27. Propón un algoritmo de coste razonable para saber si dos DFA's de entrada reconocen el mismo lenguaje.

29. Justifiqueu la veracitat o falsetat de les següents afirmacions per a DFAs mínims A, A_1, A_2, NFAs B, B_1, B_2, B_3 i morfisme σ. En cas que les operacions que apareixen hagin estat definides només per a DFAs, assumiu la seva extensió natural a NFAs.

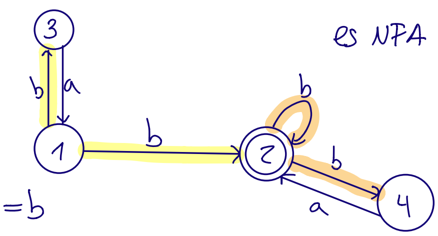
(a) $A_1 \cap A_2$ és DFA mínim.
(b) $A_1 \cup A_2$ és DFA mínim.
(c) $\bar{A}$ és DFA mínim.
(d) $\sigma(A)$ és DFA.
(e) $\sigma^{-1}(A)$ és DFA mínim.
(f) $\bar{\bar{A}} = A$.
(g) $(B^R)^R = B$.
(h) $(B^*)^* = B^*$.
(i) $(B_1 B_2) B_3 = B_1 (B_2 B_3)$.
(j) $(B_1 B_2)^R = B_2^R B_1^R$.
(k) $(B^R)^* = (B^*)^R$.
(l) En el cas en que A^R també sigui DFA, llavors podem concloure que és mínim.

d) $\sigma(A)$ es un DFA → FALS

A:



$\sigma(A)$:



$\sigma(a) = b$

$\sigma(b) = ba$

31. Dado un lenguaje L, definimos $\texttt{Prefijos}(L)$ como el lenguaje $\{w | \exists x : (wx \in L)\}$. Dado un DFA A, cómo se puede construir un DFA $\texttt{Prefijos}(A)$ que cumpla $\mathcal{L}(\texttt{Prefijos}(A)) = \texttt{Prefijos}(\mathcal{L}(A))$.

32. Dado un lenguaje L, definimos $\texttt{Sufijos}(L)$ como el lenguaje $\{w | \exists x : (xw \in L)\}$. Dado un DFA A, cómo se puede construir un DFA $\texttt{Sufijos}(A)$ que cumpla $\mathcal{L}(\texttt{Sufijos}(A)) = \texttt{Sufijos}(\mathcal{L}(A))$.

35. Sigui $B_n = \{a^k \mid k \text{ és un múltiple de } n\}$. Demostreu que per a cada $n \geq 1$, el llenguatge B_n és regular.