

# Modelos de Computación Examen VIII



*Escuela Técnica Superior de Ingenierías  
Informática y de Telecomunicación*

**Los Del DGIIM**, [losdeldgiim.github.io](https://losdeldgiim.github.io)

Doble Grado en Ingeniería Informática y Matemáticas  
Universidad de Granada



Esta obra está bajo una Licencia Creative Commons Atribución-NoComercial-SinDerivadas 4.0 Internacional (CC BY-NC-ND 4.0).

Eres libre de compartir y redistribuir el contenido de esta obra en cualquier medio o formato, siempre y cuando des el crédito adecuado a los autores originales y no persigas fines comerciales.

# Modelos de Computación Examen VIII

Los Del DGIIM, [losdeldgiim.github.io](https://losdeldgiim.github.io)

José Juan Urrutia Milán  
Asrturo Olivares Martos

Granada, 2024-2025

**Asignatura** Modelos de Computación

**Curso Académico** 2024-25.

**Grado** Doble Grado en Ingeniería Informática y Matemáticas.

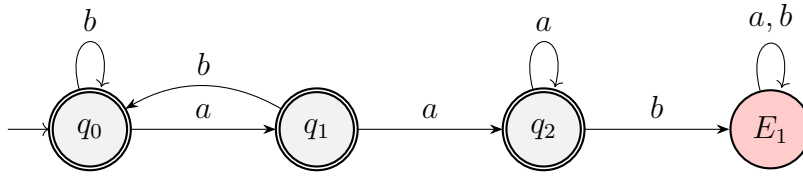
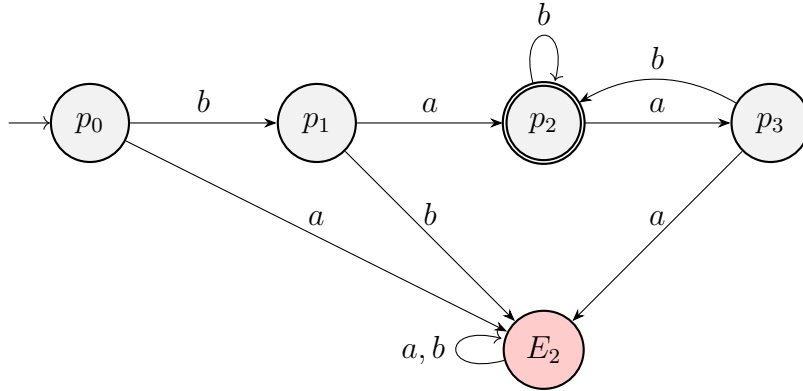
**Grupo** A1.

**Profesor** Marios Kountouris.

**Descripción** Parcial Temas 3 y 4.

**Fecha** 11 de diciembre de 2024.

**Duración** 60 minutos.

Figura 1: Autómata Finito Determinista para  $L_1$ .Figura 2: Autómata Finito Determinista para  $L_2$ .

**Ejercicio 1.** Sobre el alfabeto  $A = \{a, b\}$ , se pide:

1. Construir un Autómata Finito Determinista para el lenguaje:

$$L_1 = \{u \in \{a, b\}^* \mid u \text{ no contiene la subcadena "aab"}\}$$

El AFD se puede ver en la Figura 1.

2. Construir un Autómata Finito Determinista para el lenguaje  $L_2$  dado por la expresión regular:

$$ba(ab + b)^*$$

El AFD se puede ver en la Figura 2.

3. Construir un Autómata Finito Determinista que acepte el lenguaje  $L_1 \cap L_2$  y minimizarlo.

Para ello, usaremos el autómata producto. Los estados finales son:

$$F = \{q_0p_2, q_1p_2, q_2p_2\}$$

En primer lugar, y para que la obtención del autómata sea más sencilla, vemos que todos los estados de la forma  $q_iE_2, E_1p_j, E_1E_2$  con  $i \in \{0, 1, 2\}$  y  $j \in \{0, 1, 2, 3\}$  son estados indistinguibles, puesto que de ellos nunca vamos a poder salir (al no poder salir de  $E_1E_2$ ). Como ninguno de ellos es final, son indistinguibles y los fusionamos en un único estado de error  $E$ . El AFD producto al que llegamos, habiendo previamente fusionado dichos estados, se puede ver en la Figura 3.

La minimización del autómata producto se muestra en la Tabla 1.

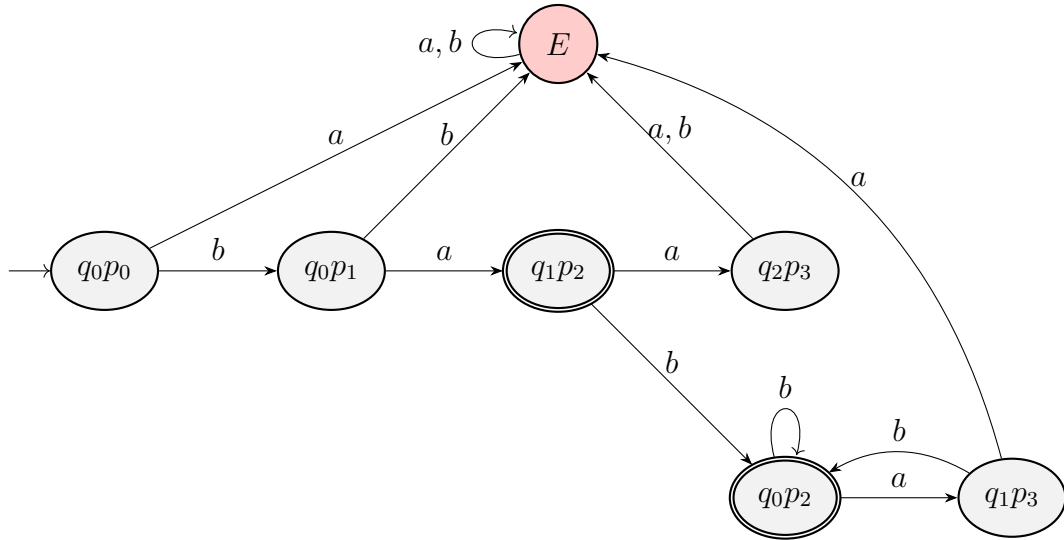
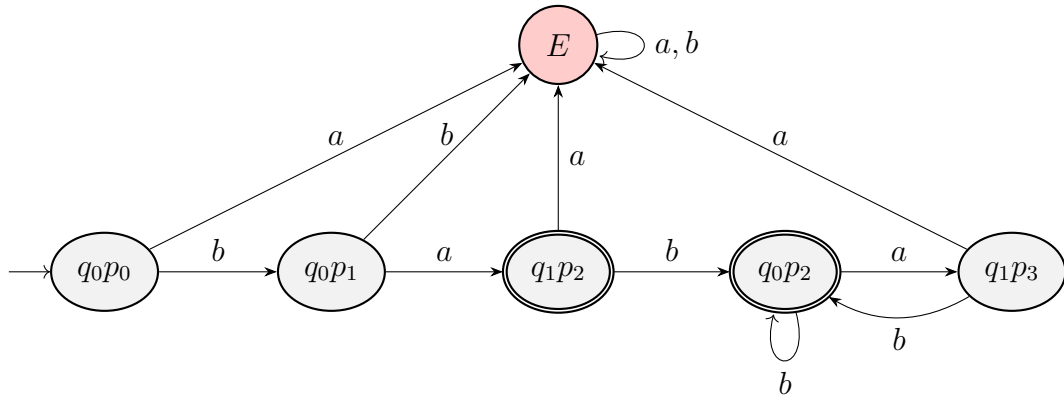


Figura 3: Autómata Finito Determinista para  $L_1 \cap L_2$ .

$q_0p_1$	×					
$q_1p_2$	×	×				
$q_2p_3$	×	×	×			
$q_0p_2$	×	×	×	×		
$q_1p_3$	×	×	×	×	×	
$E$	×	×	×		×	×
	$q_0p_0$	$q_0p_1$	$q_1p_2$	$q_2p_3$	$q_0p_2$	$q_1p_3$

Tabla 1: Tabla de minimización del AFD producto.

Figura 4: Autómata Finito Determinista mínimo para  $L_1 \cap L_2$ .

Por tanto, hemos detectado que  $q_2p_3$  también es de error. Es decir, notando por  $\equiv$  a la relación de indistinguibilidad, tenemos que:

$$q_2p_3 \equiv E$$

Por tanto, el autómata mínimo es el que se muestra en la Figura 4.

**Ejercicio 2.** Sea la gramática  $G = (\{S\}, \{a, b, c\}, P, S)$  con  $P$  el conjunto que contiene las producciones:

$$S \rightarrow abSba \mid baSab \mid ababSbaba \mid babaSabab \mid ccc$$

1. Demuestra que  $\mathcal{L}(G)$  no es regular.

Veamos en primer lugar el valor de  $\mathcal{L}(G)$  mediante doble inclusión:

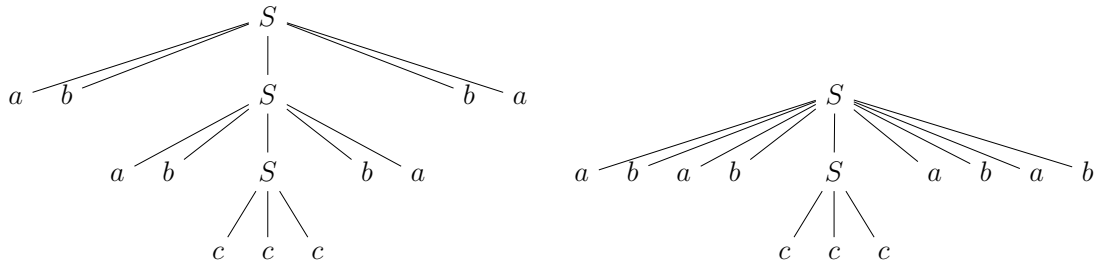
$$\mathcal{L}(G) = \{ucccu^{-1} \mid u \in \{ab, ba\}^*\}$$

- $\subseteq$ ) Sea  $w \in \mathcal{L}(G)$ . Entonces, existe una sucesión de pasos de derivación desde  $S$  tal que  $S \xRightarrow{*} w$ . En cada paso de derivación, vamos obteniendo la palabra  $u$  y  $u^{-1}$ , y para finalizar es necesario usar  $S \rightarrow ccc$ . Por tanto,  $w = ucccu^{-1}$  con  $u \in \{ab, ba\}^*$ .
- $\supseteq$ ) Sea  $w = ucccu^{-1}$  con  $u \in \{ab, ba\}^*$ . Entonces, con las 2 primeras reglas de  $P$  podemos obtener  $uSu^{-1}$ , y finalmente con la quinta regla obtenemos  $ucccu^{-1}$ .

Veamos ahora que no es regular usando el recíproco del lema de Bombeo. Para cada  $n \in \mathbb{N}$ , sea  $z = (ab)^n c^3 (ba)^n$ , con  $|z| = 4n + 3 \geq n$ . Sea ahora una descomposición  $z = uvw$  de  $z$ , con  $u, v, w \in \{a, b, c\}^*$  y  $|uv| \leq n$  y  $|v| \geq 1$ .

Como  $|uv| \leq n$ , tenemos que  $v$  es una subcadena propia de  $(ab)^n$ . Por tanto, bombeando con  $i = 0$ , tenemos que queda  $uv^0w = xc^3(ba)^n$ , con  $x \in \{a, b\}^*$  subcadena propia de  $(ab)^n$ ,  $|x| = |(ab)^n| - |v| = 2n - |v| < 2n$ .

Por tanto, como  $|x| \neq |(ba)^n|$ , tenemos que  $(ba)^n \neq x^{-1}$ , y por tanto tenemos que  $uv^0w \notin \mathcal{L}(G)$ . Por el recíproco del lema de Bombeo,  $\mathcal{L}(G)$  no es regular.

Figura 5: Árboles de derivación para  $(ab)^2 c^3 (ba)^2$ .

2. Demuestra que  $G$  es una gramática ambigua.

Para ello, basta con ver que la palabra  $(ab)^2 c^3 (ba)^2 \in \mathcal{L}(G)$  tiene más de un árbol de derivación. En concreto, los árboles de derivación que se pueden obtener son los que se muestran en la Figura 5.

3. Dar una gramática no ambigua que genere el lenguaje  $\mathcal{L}(G)$ .

Consideramos la gramática  $G' = (\{S'\}, \{a, b, c\}, P', S')$  con  $P'$  el conjunto que contiene las producciones:

$$S' \rightarrow abS'ba \mid baS'ab \mid ccc$$

Es directo ver que  $\mathcal{L}(G') = \mathcal{L}(G)$ , y que  $G'$  es una gramática no ambigua, puesto que dependiendo de la palabra tan solo podrá derivar de una forma, ya que cada regla de producción comienza por un símbolo terminal distinto.