

Modelos de Computación Examen VI



UNIVERSIDAD
DE GRANADA

*Escuela Técnica Superior de Ingenierías
Informática y de Telecomunicación*

Los Del DGIIM, losdelgiim.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 VI

Los Del DGIIM, losdeldgiim.github.io

Arturo 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 1 y 2.

Fecha 6 de noviembre de 2024.

Duración 60 minutos.

Ejercicio 1. Sea una gramática $G = \{V, T, P, S\}$ con $V = \{S, A, B\}$, $T = \{1, 0\}$ y las reglas de producción:

$$\begin{aligned} S &\rightarrow A101B \\ A &\rightarrow 1A1 \mid \varepsilon \\ B &\rightarrow 0B \mid 1B \mid \varepsilon \end{aligned}$$

1. ¿Cuál es el lenguaje generado por esta gramática?

En primer lugar, vemos que A genera palabras de la forma $(11)^*$, y B genera palabras de la forma $(0+1)^*$. Por tanto, la expresión regular asociada a $\mathcal{L}(G)$ es:

$$(11)^* \mathbf{101} (0+1)^*$$

El lenguaje por tanto es:

$$\mathcal{L}(G) = \{1^{2n}101w \in \{1, 0\}^* \mid n \in \mathbb{N} \cup \{0\}, w \in \{0, 1\}^*\}$$

2. Obtén una gramática lineal por la izquierda que genere el mismo lenguaje.

Sea $G' = \{V', T, P', S'\}$ con $V' = \{S', A'\}$ y las reglas de producción:

$$\begin{aligned} S' &\rightarrow S'0 \mid S'1 \mid A'101 \\ A' &\rightarrow A'11 \mid \varepsilon \end{aligned}$$

Ejercicio 2. Sea un lenguaje L que debe cumplir las condiciones:

- Contener exactamente dos a 's
- Contener al menos una b entre las dos a 's (después de la segunda a no hay restricciones y pueden aparecer más b 's)

Obtener:

1. La expresión regular.

$$b^* \mathbf{ab^+ab^*}$$

2. El AFND con transiciones nulas.

El AFND con transiciones nulas que se obtiene algorítmicamente a partir de la expresión regular viene representado en la figura 1.

Este AFND se puede simplificar aún más eliminando las transiciones nulas, como se muestra en la figura 2.

3. El AFD.

El AFND de la Figura 2 sigue sin ser determinista, puesto que hay transiciones que no están definidas para todos los símbolos del alfabeto. Aplicamos el algoritmo para obtener el AFD, que se muestra en la Figura 3.

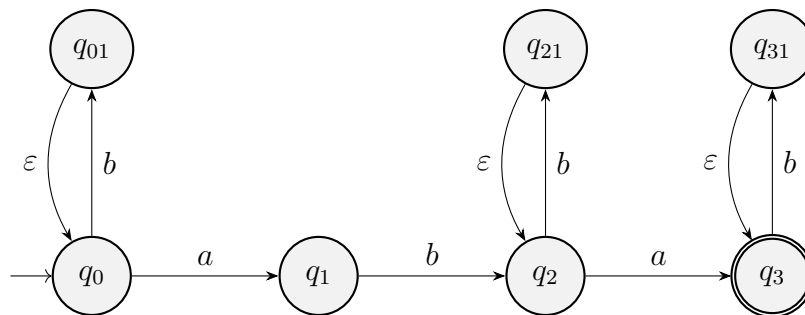


Figura 1: AFND con transiciones nulas que acepta el lenguaje del Ejercicio 2.

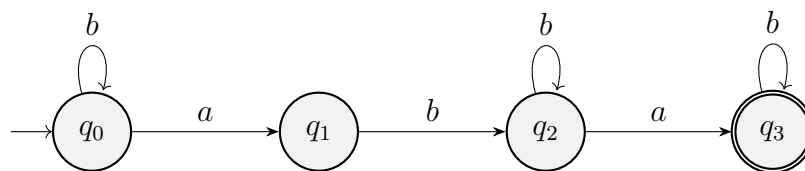


Figura 2: AFND simplificado que acepta el lenguaje del Ejercicio 2.

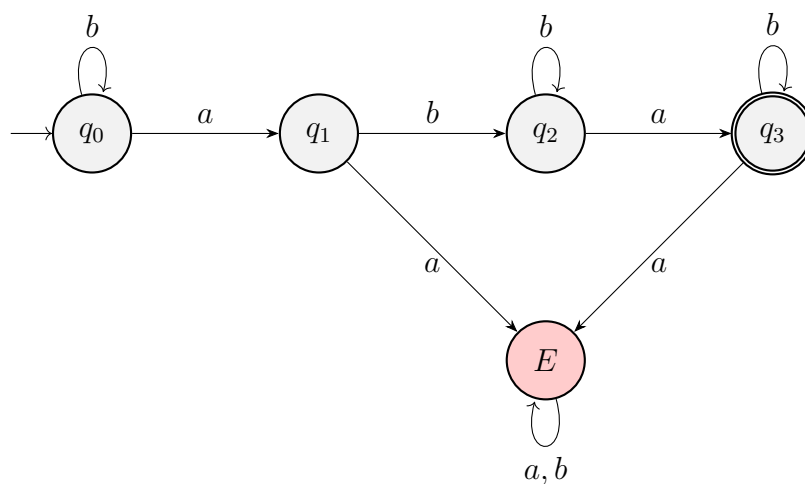


Figura 3: AFD que acepta el lenguaje del Ejercicio 2.