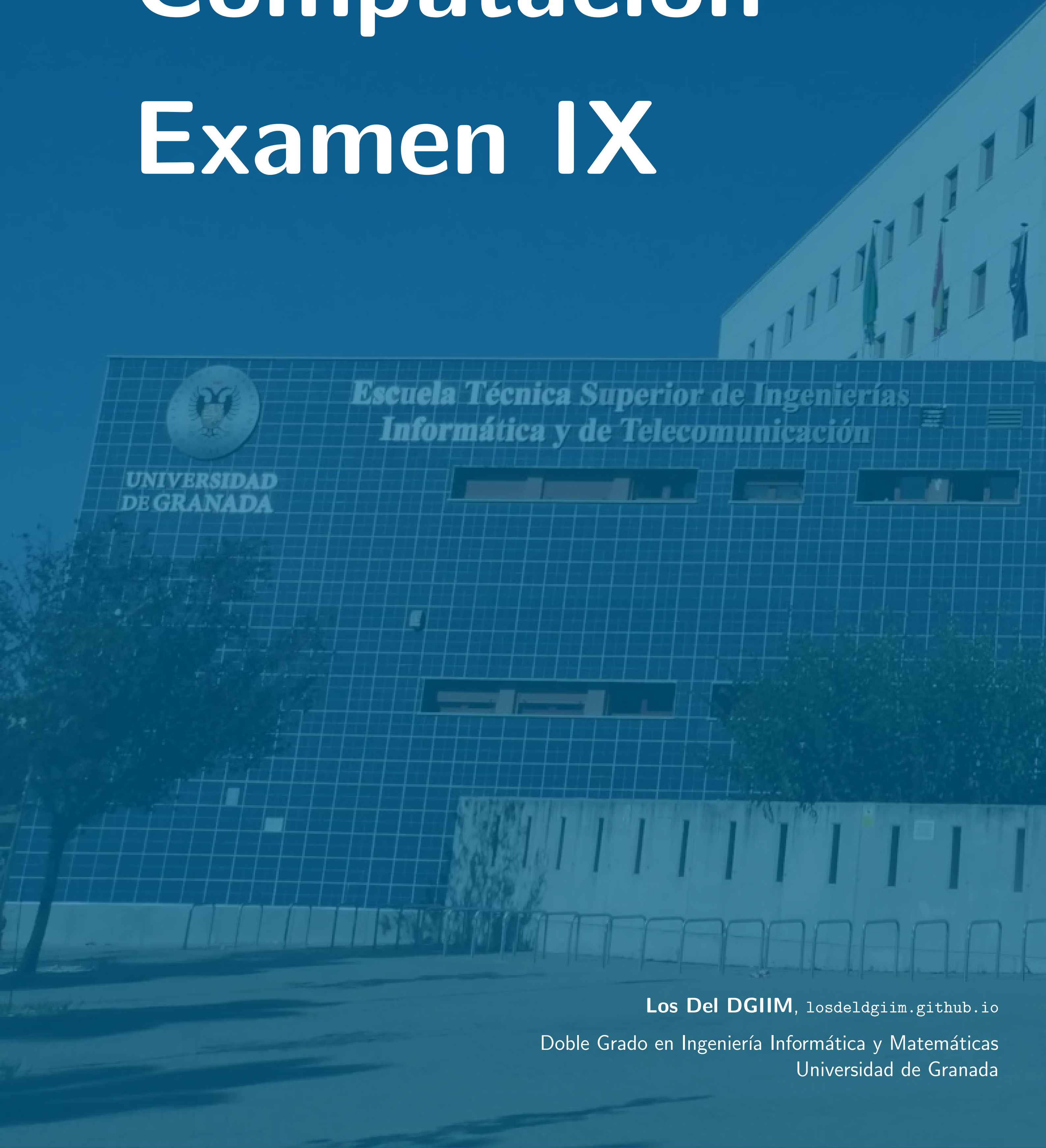
Modelos de Computación Examen IX



Los Del DGIIM, `losdeldgiim.github.io`

Doble Grado en Ingeniería Informática y Matemáticas
Universidad de Granada

Modelos de Computación
Examen IX

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 A2.

Profesor Serafín Moral Callejón.

Descripción Parcial Temas 3 y 4.

Fecha 12 de diciembre de 2024.

Duración 60 minutos.

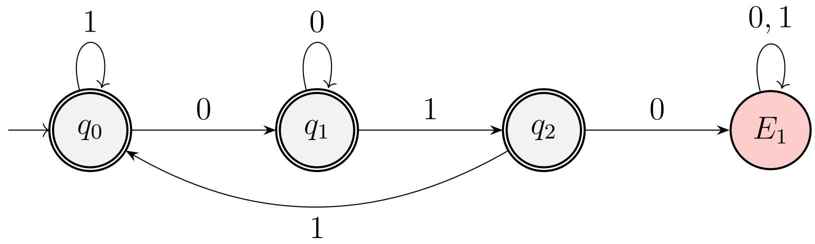


Figura 1: Autómata Finito Determinista para L_1.

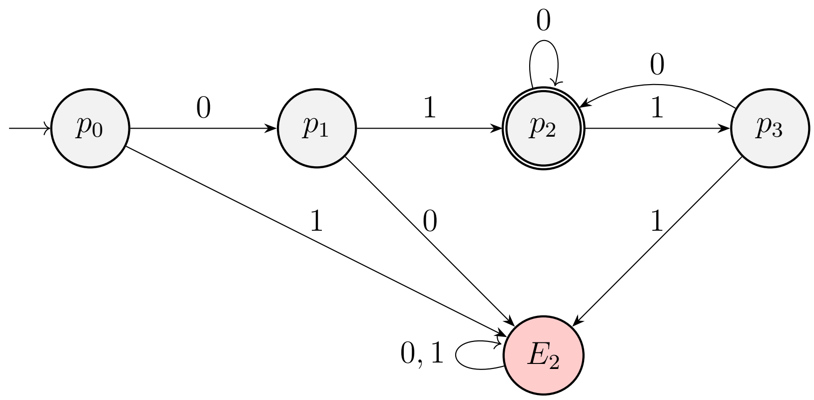


Figura 2: Autómata Finito Determinista para L_2.

Ejercicio 1. Sobre el alfabeto $A = \{0, 1\}$, se pide:

1. Construir un Autómata Finito Determinista para el lenguaje:

$$L_1 = \{u \in \{0, 1\}^* \mid u \text{ no contiene la subcadena “010”}\}$$

   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:

$$01(10 + 0)^*$$

   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.

   En primer lugar, todos los estados de error son indistinguibles, ya que desde ellos no se puede salir y no se puede alcanzar ninguno final. Notando por E al estado de error, tenemos que:

$$E = \{(q_i E_2, E_1 p_j) \mid i \in \{0, 1, 2\}, j \in \{0, 1, 2, 3\}\}$$

   Por tanto, el AFD que acepta $L_1 \cap L_2$ es el que se muestra en la Figura 3. No obstante, notando por $\equiv$ la relación de equivalencia de estados indistinguibles, vemos que:

$$E \equiv q_2 p_3$$

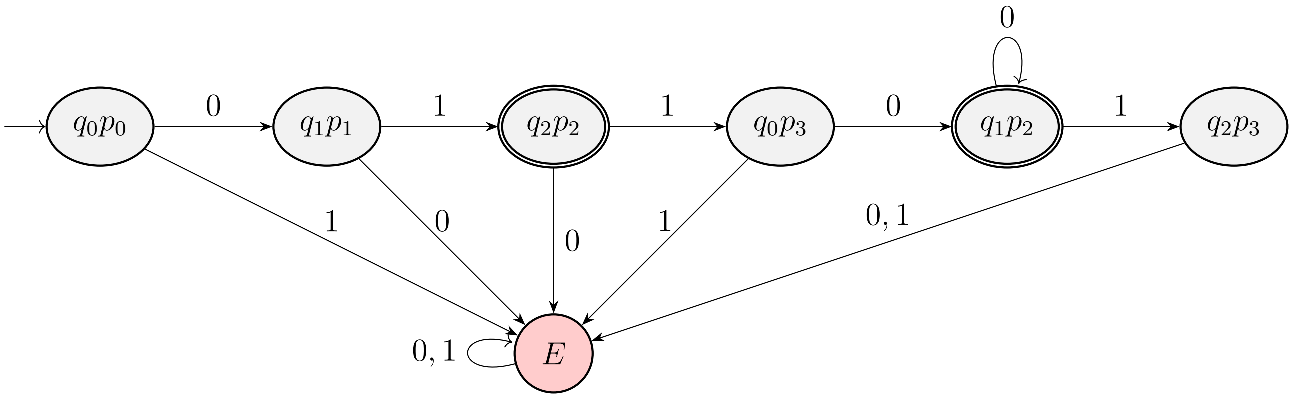


Figura 3: Autómata Finito Determinista para $L_1 \cap L_2$.

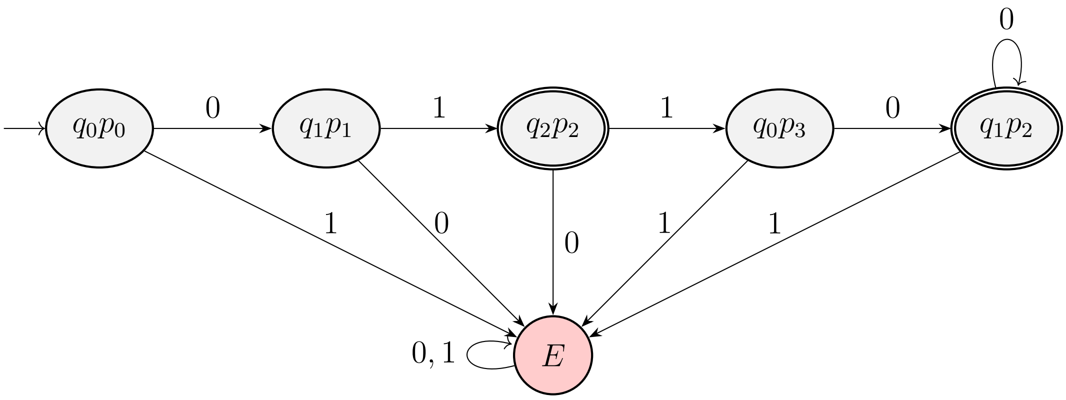


Figura 4: Autómata Finito Determinista minimal para $L_1 \cap L_2$.

ya que, desde q_2p_3, vamos a E con cualquier símbolo. Por tanto, el AFD minimizado es el que se muestra en la Figura 4.

Además, como no hay ciclos, sería fácil comprobar que el autómata de la Figura 4 es minimal.

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

$$S \to aaSbb \mid bbSaa \mid aaaSbbb \mid bbbAaaa \mid ccc$$

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

   Lo haremos mediante el Lema de Bombeo. Para cada $n \in \mathbb{N}$, consideramos la palabra $z = a^{2n}c^3b^{2n} \in \mathcal{L}(G)$, con $|z| = 4n + 3 \geqslant n$. Para cada descomposición $z = uvw$ con $|uv| \leqslant n$ y $|v| \geqslant 1$, tenemos que:

   $$u = a^k, \ v = a^l, \ w = a^{2n-k-l}c^3b^{2n} \qquad \text{con } 0 \leqslant k + l \leqslant n, \ l \geqslant 1$$

   Bombeando con $i = 2$, obtenemos la palabra:

   $$uv^2w = a^{2n+l}c^3b^{2n} \notin \mathcal{L}(G)$$

   la cual sabemos que no pertenece a $\mathcal{L}(G)$, ya que $2n + l \neq 2n$ por ser $l \geqslant 1$, y todas las palabras de $\mathcal{L}(G)$ tienen la misma longitud antes de c^3 y después.

   Por tanto, por el recíproco del Lema de Bombeo, $\mathcal{L}(G)$ no es regular.

2. Demuestra que G es una gramática ambigua.

   Sea la palabra $z = a^6c^3b^6 \in \mathcal{L}(G)$. Podemos derivarla de dos formas distintas, como se muestra en la Figura 5. Por tanto, G es ambigua.

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

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

   $$\begin{aligned} S' &\to S_A \mid S_B \mid c^3 \\ S_A &\to aaAbb \\ S_B &\to bbBaa \\ A &\to aAb \mid S_B \mid c^3 \\ B &\to bBa \mid S_A \mid c^3 \end{aligned}$$

   Es directo ver que $\mathcal{L}(G') = \mathcal{L}(G)$ y que G' es una gramática no ambigua.

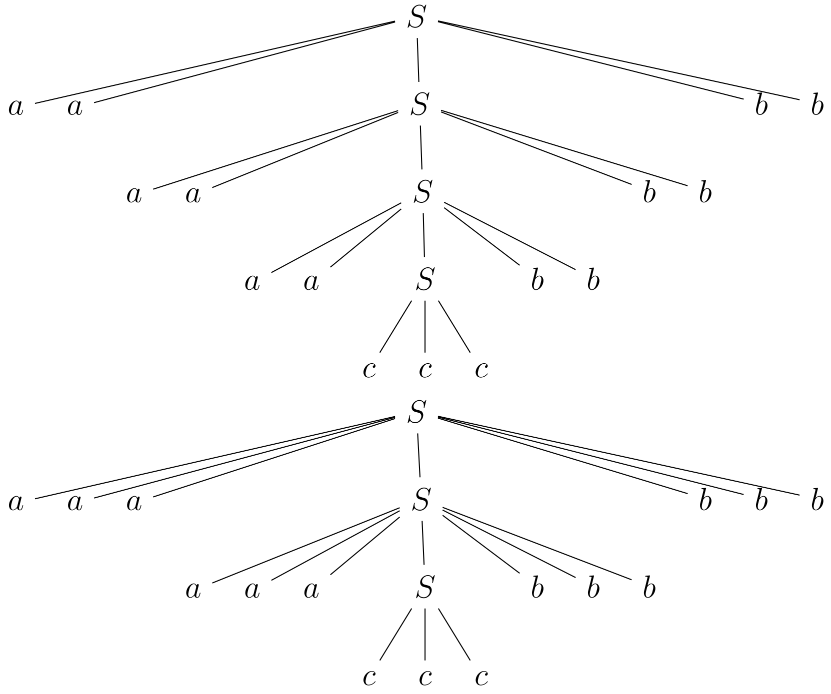


Figura 5: Árboles de derivación para $z = a^6c^3b^6$.