TECNOLOGICO NACIONAL DE MEXICO

CAMPUS ENSENADA

LENGUAJES Y AUTÓMATAS I

Xenia Padilla

TAREA 02 - PROGRAMACIÓN DE AFD

Amador Peralta Cesar Iván 22760238

Ing. Sistemas Computacionales

Ensenada, Baja California, México a 28 de octubre del 2025

Introducción

Este reporte documenta el diseño de cuatro Autómatas Finitos Deterministas (AFD) distintos sobre el alfabeto \Sigma = \ {a, b\}. Cada AFD está diseñado para reconocer un lenguaje regular específico, basado en las siguientes propiedades de las cadenas de entrada:

1. Contener la subcadena "aba" (El Perrito Guardián).
2. Tener una longitud que sea múltiplo de 3 (El Gatito de los Tres Pasos).
3. Contener un número par de símbolos 'a' (El Loro que Contaba 'a's Pares).
4. Terminar con la subcadena "bb" (El Conejo que Termina en 'bb').

Cada AFD se define formalmente como una quíntupla M = (Q, \Sigma, \delta, q_0, F), donde:

- **Q** es el conjunto finito de estados.
- \Sigma es el alfabeto de entrada (\ {a, b\}).
- \delta: Q \times \Sigma \to Q$ es la función de transición.
- q_0 \in Q es el estado inicial.
- F \subseteq Q es el conjunto de estados de aceptación (finales).

Desarrollo

EJERCICIO 1: El Perrito Guardián (Subcadena "aba")

Este AFD necesita recordar la parte más larga del prefijo de "aba" que se ha leído hasta el momento.

- Q = \ {q_0, q_1, q_2, q_{aba}\}

    q_0: No se ha visto ninguna parte de "aba" o la última letra fue 'b' (reseteando).

    q_1: Visto un prefijo de "a".

    q_2: Visto un prefijo de "ab".

    q_{aba}: Estado de Aceptación. Se ha visto "aba". Una vez en este estado, cualquier otra entrada lo mantiene allí.

- q_0: Estado Inicial
- F = \{q_{aba}\}

δ	a	b
q_0	q_1	q_0
q_1	q_1	q_2
q_2	q_{aba}	q_0
q_{aba}	q_{aba}	q_{aba}

EJERCICIO 2: El Gatito de los Tres Pasos (Longitud Múltiplo de 3)

Este AFD utiliza sus estados para contar la longitud de la cadena módulo 3.

- $Q = \{q_0, q_1, q_2\}$

    q_0: Longitud $\equiv 0 \pmod 3$.

    q_1: Longitud $\equiv 1 \pmod 3$.

    q_2: Longitud $\equiv 2 \pmod 3$.

- q_0: Estado Inicial
- $F = \{q_0\}$ (Acepta la cadena vacía ε ya que $|\varepsilon|=0$, y $0 \equiv 0 \pmod 3$)

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

EJERCICIO 3: El Loro que Contaba 'a's Pares (Número Par de 'a's)

Este AFD utiliza dos estados para rastrear la paridad del número de 'a's leídas.

- $Q = \{q_{par}, q_{impar}\}$
    - q_{par}: Se ha leído un número **par** de 'a's.
    - q_{impar}: Se ha leído un número **impar** de 'a's.
- q_{par}: Estado Inicial
- $F = \{q_{par}\}$ (El cero es un número par, por lo que ε es aceptada).

δ	**a**	**b**
q_{par}	q_{impar}	q_{par}
q_{impar}	q_{par}	q_{impar}

EJERCICIO 4: El Conejo que Termina en 'bb'

Este AFD debe recordar los últimos dos símbolos leídos para verificar el sufijo "bb".

- $Q = \ \{q_0, q_1, q_2, q_{bb}\}$
        - q_0: No se ha visto 'b' o la última letra fue 'a' (sin progreso).
        - q_1: Última letra leída fue 'b'.
        - q_2: Últimas dos letras leídas fueron 'bb' (Estado de Aceptación).
- q_0: Estado Inicial
- $F = \{q_2\}$

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

Minimización y Determinismo

Es crucial notar que todos los autómatas diseñados son deterministas (AFD), ya que para cada estado y cada símbolo del alfabeto, existe una y solo una transición. Esto cumple con la definición formal $M = (Q, \Sigma, \delta, q_0, F)$.

En el diseño, se ha buscado la minimización de estados para reducir la complejidad. Por ejemplo, en el caso de El Loro que Contaba 'a's Pares, solo dos estados son suficientes para diferenciar entre paridad. De manera similar, en el caso de El Conejo que Termina en 'bb', los estados rastrean eficientemente el prefijo más largo del sufijo "bb" que se ha visto, logrando la mínima cantidad necesaria de estados (cuatro). El estado q_{aba} en el Perrito Guardián es un estado trampa de aceptación o estado sumidero de aceptación, ya que una vez alcanzado, la cadena siempre será aceptada, independientemente de la entrada posterior.

Conclusión

El diseño de Autómatas Finitos Deterministas es fundamental en la Teoría de la Computación, siendo la base para la comprensión de los lenguajes regulares. Los cuatro AFDs presentados demuestran cómo un modelo matemático simple, con una memoria finita representada por sus estados, puede reconocer propiedades específicas en las cadenas de entrada. El éxito de estos diseños se verifica por su correcta aceptación de las cadenas positivas (\checkmark Acepta) y rechazo de las cadenas negativas (\times Rechaza).

Fuentes Bibliográficas

- Hopcroft, J. E., Motwani, R., & Ullman, J. D. (2007). *Introduction to Automata Theory, Languages, and Computation* (3rd ed.). Pearson Education.
- Materiales y ejercicios de Teoría de Autómatas y Lenguajes Formales (Universidad, OCW, Recursos en línea).