ASCE MAGAZINE ISSN: 3073–1178

Revista ASCE Magazine, Periodicidad: Trimestral, Volumen: 5, Número: 2, Año: 2026 páginas 453-472

Doi: https://doi.org/10.70577/asce.v5i2.775

Recibido: 2026-03-22

Aceptado: 2026-04-06

Publicado: 2026-04-22

Autómatas finitos y reconocimiento de lenguajes formales

Finite automata and formal language recognition

Autor

Angel Yael Sarabia Martínez

anguelyaelo@gmail.com

https://orcid.org/0009-0006-8523-2805

Universidad Autónoma del Estado de Hidalgo

Pachuca Hidalgo - México

Como Citar

Sarabia Martinez. A. Y. (2026) Autómatas finitos y reconocimiento de lenguajes formales. ASCE MAGAZINE, 5(2) 453-472

Resumen

Los autómatas finitos son considerados pilares fundamentales para el procesamiento de información estructurada, pues, constituyen la herramienta primordial para el reconocimiento de lenguajes formales, específicamente de aquellos clasificados como regulares. El objetivo de la investigación es analizar los fundamentos teóricos de los autómatas finitos y explicar su función en el reconocimiento de lenguajes formales. Se realizó una revisión bibliográfica con enfoque crítico, descriptivo y analítico; se aplicó el método PRISMA; se recopilaron Artículos Científicos (AC) previamente publicados en bases de datos académicas; se aplicaron filtros como criterios de inclusión y exclusión tomando en cuenta las preguntas y objetivos de investigación. Se comparó a los autómatas finitos y su relación con los lenguajes formales, se conoció que todo autómata finito no determinista tiene un autómata finito determinista equivalente que reconoce el mismo lenguaje, lo que garantiza que el no determinismo, no incrementa el poder expresivo; la robustez de los lenguajes formales se sustenta en sus propiedades de cerradura, pues, permite que los autómatas finitos superen el ámbito teórico para integrarse en aplicaciones críticas de ingeniería. Se concluyó que, los autómatas finitos se consolidan como el nivel más elemental de la jerarquía de Chomsky y las gramáticas regulares; las propiedades de cerradura algebraica permiten integrar estos modelos en aplicaciones críticas como la verificación formal; en investigaciones posteriores, se recomienda priorizar el uso de autómatas finitos deterministas minimizados en el desarrollo de analizadores léxicos para compiladores y motores de búsqueda de patrones.

Palabras clave: Automatización, Determinismo, Lenguaje de programación, Lexicografía, Reconocimiento de caracteres

Abstract

Finite automata are considered fundamental pillars for structured information processing, as they constitute the primary tool for recognizing formal languages, specifically those classified as regular. The objective of this research is to analyze the theoretical foundations of finite automata and explain their function in the recognition of formal languages. A literature review was conducted with a critical, descriptive, and analytical approach; the PRISMA method was applied; previously published scientific articles were compiled from academic databases; and filters were applied as inclusion and exclusion criteria, taking into account the research questions and objectives. A comparison of finite automata and their relationship to formal languages was made. It was found that every non-deterministic finite automaton has an equivalent deterministic finite automaton that recognizes the same language, guaranteeing that non-determinism does not increase expressive power. The robustness of formal languages is based on their closure properties, which allow finite automata to move beyond the theoretical realm and be integrated into critical engineering applications. It was concluded that finite automata are consolidated as the most elementary level of the Chomsky hierarchy and regular grammars. The algebraic closure properties allow these models to be integrated into critical applications such as formal verification. In future research, it is recommended to prioritize the use of minimized deterministic finite automata in the development of lexical analyzers for compilers and pattern recognition engines.

Keywords: Automation, Determinism, Programming language, Lexicography, Character recognition

Introducción

La teoría de la computación ha experimentado una evolución significativa en los últimos años, consolidando a los autómatas finitos como pilares fundamentales para el procesamiento de información estructurada (Postiglione, 2024). Estos modelos matemáticos abstractos, definidos por un conjunto finito de estados y transiciones, constituyen la herramienta primordial para el reconocimiento de lenguajes formales, específicamente de aquellos clasificados como regulares dentro de la jerarquía de Chomsky (Marx, 2021).

En el ámbito tecnológico actual, la implementación de estos sistemas ha trascendido al campo académico para convertirse en la base de soluciones con el uso de inteligencia artificial, seguridad informática y minería de datos (Figueiras et al., 2026).

Sin embargo, aun cuando existe solidez teórica, la aplicación de los autómatas finitos conserva altos desafíos ante la explosión de datos en gran tamaño, uno de los problemas más recurrentes es la explosión de estados en sistemas complejos, donde el diseño de un autómata no optimizado puede partir de un consumo excesivo de recursos computacionales (Karakonstantis y Xylomenos, 2026).

Asimismo, en el área del procesamiento de lenguaje natural, existe dificultad para modelar la ambigüedad de los lenguajes humanos mediante estructuras rígidas, lo que obliga a integrar modelos deterministas con técnicas de aprendizaje automático, con la finalidad de mejorar la precisión en el reconocimiento de patrones y unidades de significado complejas (Yadav et al., 2021).

Por otra parte, la importancia de los autómatas finitos, se justifica por su capacidad de ofrecer un determinismo absoluto y alta eficiencia en tareas de verificación y filtrado (Figueiras et al., 2026). Su función técnica es indispensable para la construcción de analizadores de lenguajes en compiladores modernos, la detección de peligros en redes, y la extracción de dominios de conocimiento en documentos no estructurados con una precisión superior al 90% (Dora et al., 2025).

Además, su capacidad para ser minimizados matemáticamente permite desarrollar un software más ligero y rápido, siendo esta optimización un área de investigación activa, que combina teoremas clásicos, con algoritmos destinados a garantizar la viabilidad de sistemas de control en tiempo real (Negrini et al., 2024).

Por el contexto expuesto, el objetivo de la investigación es analizar los fundamentos teóricos de los autómatas finitos y explicar su función en el reconocimiento de lenguajes formales, por medio de una revisión bibliográfica.

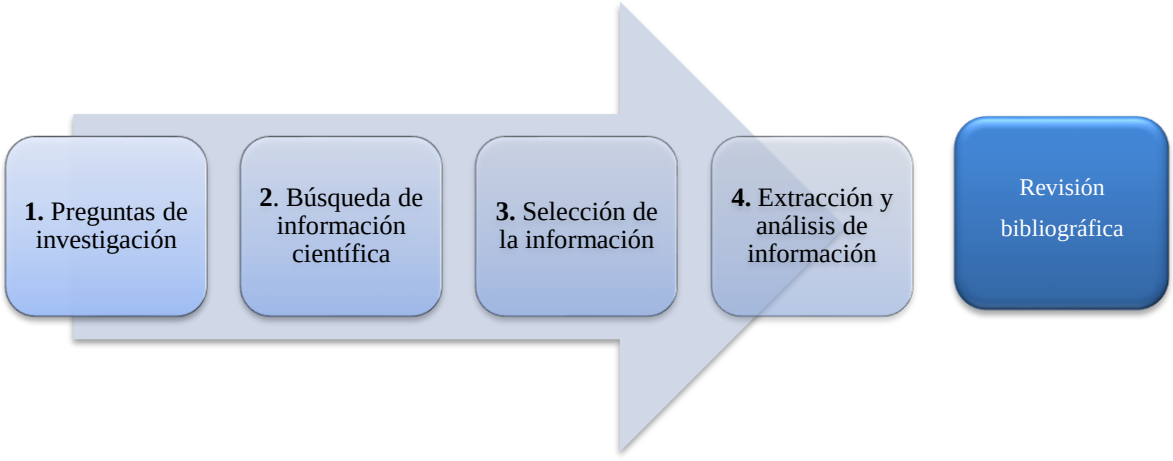
Material y métodos

La revisión bibliográfica fue elaborada al aplicar un enfoque de carácter crítico, descriptivo y analítico; la investigación se realizó por medio del método PRISMA (Preferred Reporting Items for Systematic Reviews and Meta-Analyses), para lo cual, se desarrollaron una serie de procesos, con el objetivo otorgar transparencia y reproducibilidad al artículo, tanto de la elección y recopilación de la información, como de su análisis.

Para elaborar la revisión, se recopilaron diferentes Artículos Científicos (AC) previamente publicados en revistas pertenecientes a bases de datos académicas como; Google Académico, Scielo, Springer, y Elsevier; en la figura 1 se exponen los procesos de la investigación.

Figura 1

Descripción de los procesos para el desarrollo de la investigación.



Nota. Diagrama de los procesos del método PRISMA.

Las preguntas de investigación establecidas se exponen en la tabla 1, estas mantienen su enfoque en el cumplimiento de los objetivos de la investigación.

Tabla 1

Descripción de la preguntas y objetivos de la investigación.

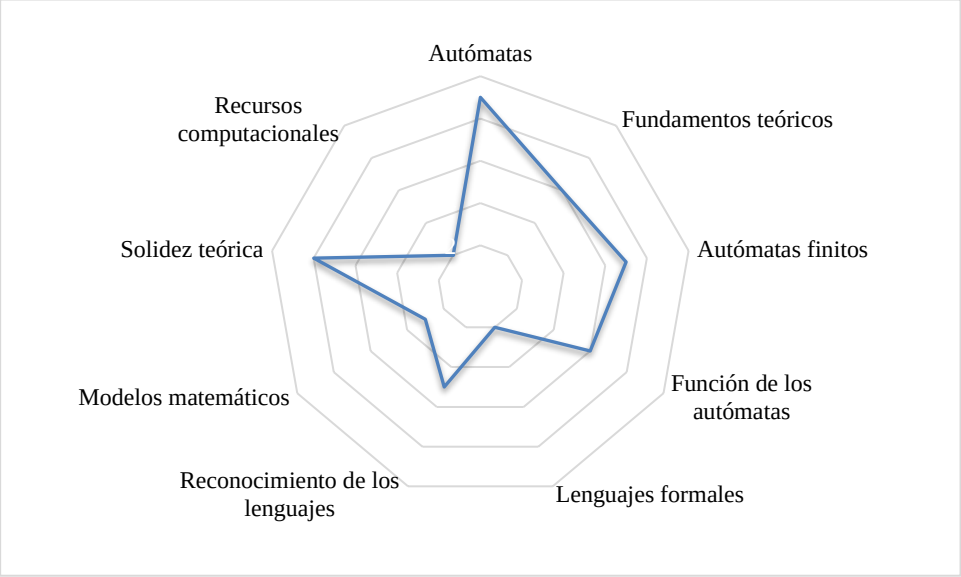
N°	Preguntas	Objetivos
1	¿Cuáles son los fundamentos teóricos autómatas finitos?	Analizar los fundamentos teóricos autómatas finitos.
2	¿Cómo funcionan los autómatas finitos en el reconocimiento de los lenguajes formales?	Explicar la función de los autómatas finitos en el reconocimiento de los lenguajes formales.

Nota. Las preguntas establecidas permiten generar los límites de la investigación.

Para proceder con la búsqueda de la información se trabajó con palabras claves (Figura 2), las cuales mantuvieron relación con las preguntas y objetivos de la investigación (Tabla 1), luego se identificó únicamente la información de AC publicados; en la búsqueda se utilizaron las palabras claves combinadas formando diversas frases. En el mencionado proceso se encontraron 82 AC en total.

Figura 2

Descripción de las palabras claves usadas para la búsqueda.



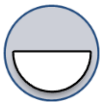
Nota. Con las palabras expuestas y sus respectivas combinaciones se procedió a la búsqueda de información.

Luego, se compararon todos los AC identificados, tomando en cuenta los títulos, el idioma, y los nombres de los autores, para curar la información y descartar aquellos trabajos que se encuentren repetidos. En dicho proceso se redujo la cantidad de AC de 82 a 55, al descartar 27 AC.

Al aplicar el método PRISMA, se usaron los filtros de información mediante el uso de los criterios de inclusión y exclusión descritos en la figura 3, una vez más, se generó una reducción en la cantidad total de los AC base, pues, al tomar en cuenta los criterios mencionados se conservaron únicamente 26 investigaciones para continuar con los procesos.

Figura 3

Criterios de inclusión y exclusión aplicados en el estudio.



Criterios de Inclusión

1. AC relacionados con los autómatas finitos

2. AC vinculados con el reconocimiento de lenguajes formales

3. AC que describan la función de los autómatas finitos sobre el reconocimiento de lenguajes formales

4. AC publicados en el periodo 2020 a 2026

5. AC de revistas científicas indexadas

Criterios de Exclusión

1. AC no relacionados con los autómatas finitos

2. AC desvinculados con el reconocimiento de lenguajes formales

3. AC que no describan la función de los autómatas finitos sobre el reconocimiento de lenguajes formales

4. AC publicados en periodos diferentes de 2020 a 2026

5. AC no publicados en revistas científicas indexadas

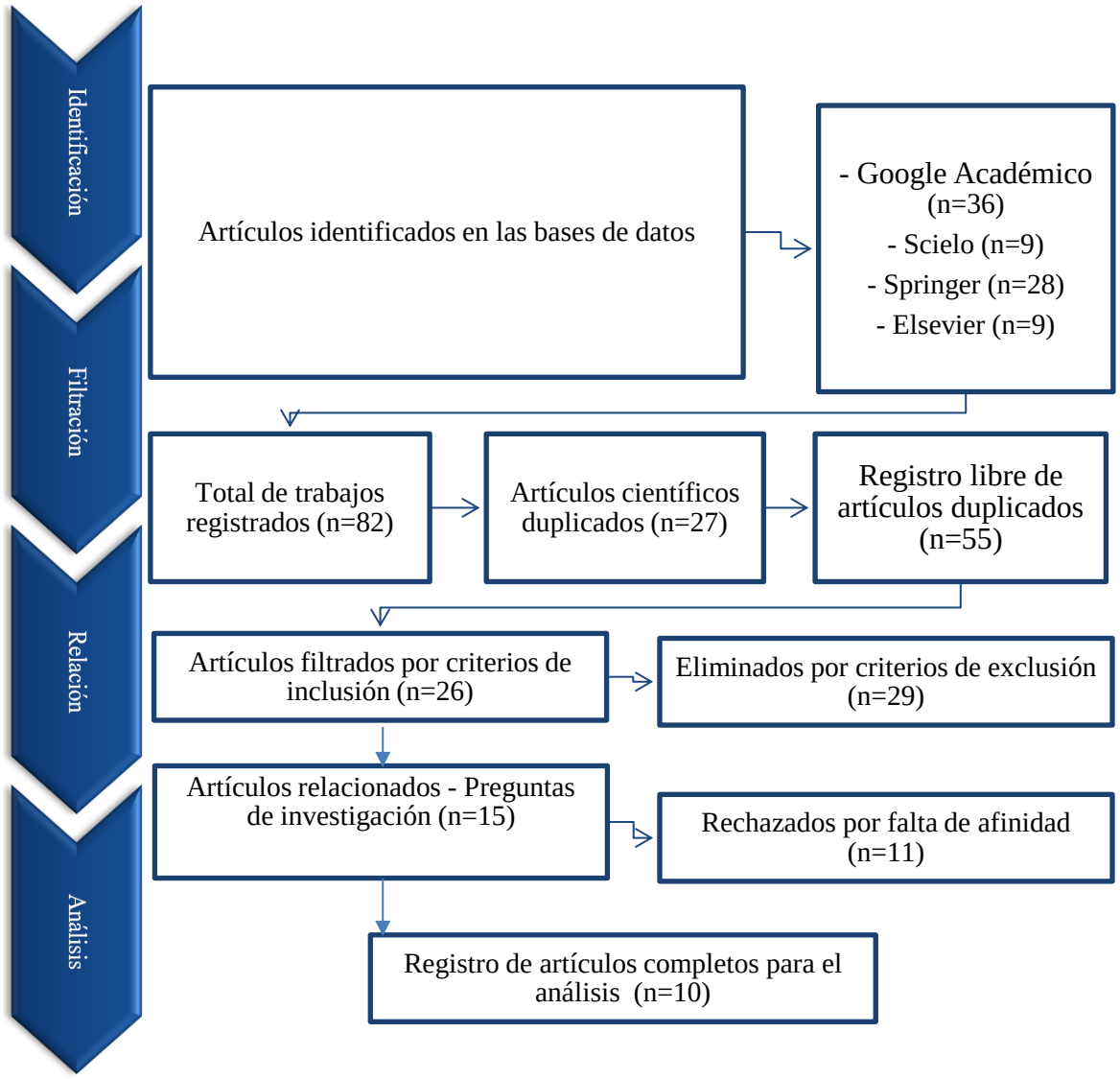
Nota. Los criterios expuestos fueron relacionados directamente con los objetivos establecidos.

Luego de aplicar los criterios descritos, se leyeron los resúmenes de los AC, se relacionó la relación o no del trabajo y la investigación planteada, considerando los objetivos establecidos (Tabla 1), en este proceso se logró reducir a 15 los AC.

Como último filtro, se leyó de forma completa los AC, por lo que quedaron excluidas aquellas investigaciones que no tuvieron expuestos todos sus componentes, como resultado se generó una matriz de 10 AC, para la elaboración de la revisión bibliográfica. Con la finalidad de exponer los procesos elaborados de forma cuantitativa al aplicar el método PRISMA, se presenta la figura 4.

Figura 4

Descripción cuantitativa de la aplicación del método PRISMA.



Nota. Inicialmente se identificó un total de 82 AC, luego de aplicar los filtros del método establecido se obtuvo una base final de 10 AC.

Resultados

Al aplicar el método PRISMA, se desarrolló una matriz (Tabla 2) en la cual se describe la información básica de los AC considerados idóneos para la elaboración de la revisión bibliográfica, en donde se busca analizar los fundamentos teóricos de los autómatas finitos y explicar su función en el reconocimiento de lenguajes formales.

Tabla 2

Descripción cualitativa de los AC seleccionados por el método PRISMA.

#	Cita	Revista	Indexación	Título	Método	Conclusiones
1	(Nagy, 2022)	Lecture Notes in Computer Science	Springer	From Finite Automata to Fractal Automata – The Power of Recursion	Revisión bibliográfica	Los lenguajes regulares pueden representarse mediante autómatas finitos (aceptores de estados finitos), expresiones regulares y diagramas de sintaxis (en su forma especial) que permiten alternativas, opciones, concatenación e iteración.
2	(Morazán & Minić, 2024)	EPTCS	Google académico	Visualización de expresiones regulares a partir de autómatas de estados finitos	Revisión bibliográfica	Existe una estrecha relación entre los autómatas finitos y las expresiones regulares como bases del estudio de los lenguajes formales. Muestra que ambos modelos son equivalentes y permiten describir los mismos lenguajes desde diferentes enfoques. El uso de visualizaciones ayuda a interpretar conceptos abstractos de manera más clara. En conjunto, esto refuerza su importancia tanto en el aprendizaje como en aplicaciones prácticas de la informática.
3	(Yue et al., 2019)	Nature Computational	Springer	Poder de reconocimiento de lenguaje y concisión de los autómatas afines.	Estudio transversal	Se resalta cómo los autómatas afines amplían la capacidad de reconocimiento en comparación con los autómatas finitos tradicionales, ofreciendo representaciones más compactas para ciertos lenguajes formales. Esto evidencia que, aunque los modelos clásicos siguen siendo fundamentales. Asimismo, se refuerza la importancia de estudiar la eficiencia y expresividad dentro de la teoría de autómatas.

4	(Yue et al., 2019)	Wiley Focus	Google académico	Aceptabilidad lingüística de autómatas finitos basados en la teoría del producto semitensor de matrices	Revisión bibliográfica	El lenguaje en autómatas finito, permite representar de manera más estructurada las transiciones y el comportamiento del sistema. A partir de ello, se fortalece la comprensión de los lenguajes formales desde una perspectiva algebraica. En conjunto, se amplían las herramientas teóricas para estudiar la capacidad y funcionamiento de los autómatas finitos.
5	(Kumar et al., 2025)	Conferencia Internacional de 2025 sobre Control Inteligente, Computación y Comunicaciones (IC3)	Google académico	La intersección de algoritmos, teoría de autómatas y lenguajes formales en la teoría computacional	Revisión bibliográfica	Se evidenció cómo los algoritmos, la teoría de autómatas y los lenguajes formales se integran para comprender los procesos computacionales. Los autómatas finitos se consideran como modelos clave para el reconocimiento y procesamiento de lenguajes. Asimismo, los lenguajes formales generan la base para estructurar y analizar estos sistemas de manera rigurosa.
6	(Kamble et al., 2024)	Conferencia Internacional sobre Tecnologías Emergentes (ICETSIS),	Google académico	Revisión de la aplicación de la teoría de autómatas en el procesamiento del lenguaje natural	Revisión bibliográfica	La teoría de autómatas, en particular los autómatas finitos, tiene un papel relevante en el procesamiento del lenguaje natural. Estos modelos permiten representar y analizar estructuras de lenguaje de manera formal y eficiente. A su vez, los lenguajes formales proporcionan el marco necesario para describir patrones y reglas del lenguaje.
7	(Gambhire et al., 2026)	Ciencias Matemáticas Discretas	Google académico	Aprovechamiento de los lenguajes formales y la teoría	Revisión sistemática	Los lenguajes formales y la teoría de autómatas, permiten representar patrones lingüísticos de forma

		y Criptografía		de autómatas en el procesamiento del lenguaje natural (PLN)		estructurada y eficiente. Además, facilitan el desarrollo de sistemas capaces de reconocer y manipular información textual. En conjunto, se reafirma su importancia como herramientas fundamentales en la intersección entre teoría computacional y aplicaciones de procesamiento del lenguaje.
8	(Nagy, 2022)	Revista Internacional de Fundamentos de la Informática	Google académico	Autómatas fractales: recursión en lenguajes libres de contexto y en lenguajes libres de contexto deterministas y lineales	Revisión bibliográfica	La recursividad en lenguajes libres de contexto amplía las posibilidades de representación más allá de los autómatas finitos. Aun así, estos últimos siguen siendo fundamentales para comprender las bases de los lenguajes formales más simples. La relación entre distintos tipos de lenguajes evidencia la necesidad de modelos adecuados según el nivel de complejidad. En conjunto, se refuerza la importancia de los autómatas finitos como punto de partida en la teoría de lenguajes formales.
9	(Yokomori & Okubo, 2021)	Menbranes	Springer	Teoría de los autómatas de reacción: una revisión.	Revisión bibliográfica	Los autómatas de reacción amplían el marco tradicional de los autómatas finitos al agregar dinámicas inspiradas en sistemas interactivos. Los autómatas finitos continúan siendo la base para comprender el reconocimiento de lenguajes formales. La comparación entre modelos evidencia diferencias en capacidad expresiva y alcance.
10	(Urbat & Schröder, 2020)	Biblioteca Digital de ACM	Google académico	Aprendizaje de autómatas: un enfoque algebraico	Revisión bibliográfica	El aprendizaje de autómatas, desde un enfoque algebraico, permite inferir modelos a partir de datos y

	observaciones. Los autómatas finitos se consolidan como estructuras clave para representar y reconocer lenguajes formales. En conjunto, se evidencia una conexión sólida entre teoría, aprendizaje y modelado dentro de la computación.

Nota. Los AC descritos fueron considerados idóneos para la elaboración de la investigación.

En la tabla 3, se presenta una comparación de los autómatas finitos y su relación con los lenguajes formales, cabe recalcar que, los autómatas finitos constituyen el nivel más básico dentro de la jerarquía de Chomsky, siendo estrictamente equivalentes a las expresiones regulares y a las gramáticas regulares. Su principal limitación radica en la ausencia de memoria estructurada, lo que impide el reconocimiento de dependencias anidadas o de largo alcance. Sin embargo, su eficiencia, cerradura algebraica y propiedades de decidibilidad los convierten en herramientas fundamentales para el análisis léxico, verificación de sistemas y diseño de compiladores.

Tabla 3

Descripción de los autómatas finitos

Tipo de autómata	Definición formal	Capacidad computacional	Relación con lenguajes formales	Características principales
Autómata Finito Determinista (AFD)	5-tupla (Q, Σ, δ, q 0 ,F) con función de transición determinista	Limitada a memoria finita (sin almacenamiento adicional)	Reconoce lenguajes regulares	Transición única por símbolo; fácil implementación; cierre bajo operaciones (unión, intersección, complemento); decidibilidad completa
Autómata Finito No Determinista (AFN)	5-tupla (Q, Σ, δ, q 0 ,F) con función de transición no determinista	Equivalente al AFD en poder expresivo	Reconoce lenguajes regulares	Permite múltiples transiciones y transiciones ε; más compacto que AFD; equivalente mediante transformación

Nota. La descripción de la simbología se realiza en los párrafos complementarios.

A continuación, se amplía su descripción desde una perspectiva técnica:

Definición formal y estructura

Un AFD se define como una 5-tupla donde:

- Q es un conjunto finito de estados
- Σ es el alfabeto de entrada
- $\delta: Q \times \Sigma \rightarrow Q$ es la función de transición
- $q_0 \in Q$ es el estado inicial
- $F \subseteq Q$ es el conjunto de estados de aceptación

En el caso del AFN la función de transición se generaliza como:

$$\delta : Q \times \Sigma \rightarrow 2^Q$$

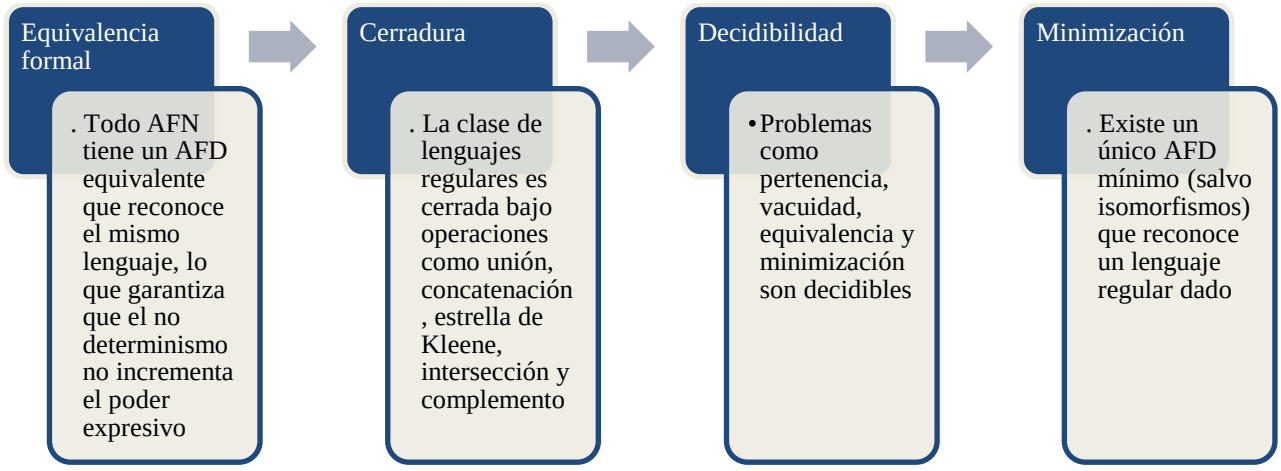
permitiendo múltiples estados sucesores.

Propiedades fundamentales

Los autómatas finitos presentan un conjunto de propiedades algebraicas y computacionales relevantes, como se muestra en la figura 5.

Figura 5

Propiedades algebraicas y computacionales de los autómatas finitos



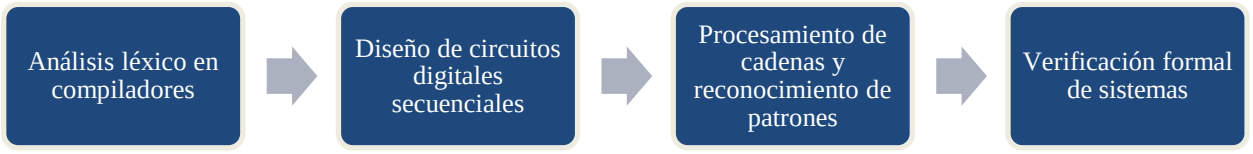
Cabe considerar que, los autómatas finitos son equivalentes a expresiones y gramáticas regulares. Esta equivalencia establece que cualquier lenguaje reconocido por un autómata finito puede describirse mediante una expresión regular y viceversa, lo que resulta fundamental en el diseño de analizadores léxicos.

Debido a la ausencia de memoria auxiliar, los autómatas finitos no pueden, reconocer lenguajes con dependencias anidadas (ejemplo $a^n b^n$), y manejar conteos arbitrarios o estructuras recursivas;

dichas limitaciones motivan el uso de modelos más potentes como los autómatas de pila. A su vez, los autómatas finitos tienen aplicaciones directas como se muestra en la figura 6.

Figura 6

Aplicaciones de los autómatas finitos



Desde un punto de vista formal, los autómatas finitos representan el nivel más restringido de cómputo dentro de la teoría de lenguajes formales; sin embargo, su robustez matemática, eficiencia algorítmica y capacidad de modelado los convierten en una herramienta esencial tanto en teoría como en aplicaciones prácticas.

Por otra parte, un lenguaje puede contener un número infinito de cadenas definidas sobre un conjunto finito de símbolos o alfabetos Σ. Sea Σ* el conjunto de todas las cadenas finitas de símbolos en Σ, incluyendo la cadena de longitud cero (ε). Por lo tanto, un lenguaje es un subconjunto de Σ* para algún alfabeto Σ. Esto hace que los lenguajes naturales y los lenguajes de programación queden incluidos en esta definición formal.

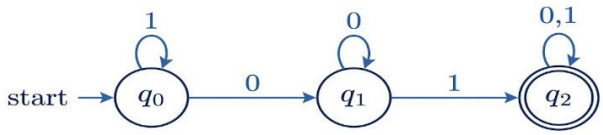
Los tipos de lenguajes proporcionan propiedades útiles del lenguaje definido. Si llamamos C a la clase del lenguaje definido especialmente con cierto tipo de descripción, entonces podemos responder si la pertenencia a la clase C se preserva bajo diversas operaciones. Otra cuestión es si un lenguaje en la clase C puede ser reconocido de manera simple y rápida, de modo que se pueda desarrollar un compilador para ese lenguaje particular en la clase C.

Funciones de transcripción del autómata finito

A continuación, en la figura 7, se detallan todas las funciones de transición del autómata finito determinista.

Figura 7

Funciones de transcripción del autómata finitos determinista

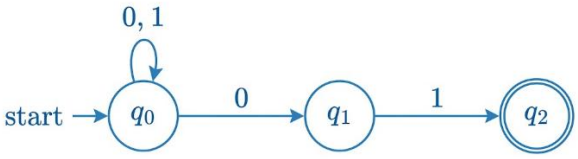


El estado q0 tiene dos funciones posibles: δ(q0, 1) = q0 y δ(q0, 0) = q1	*La primera indica que estando en el estado q0, se recibe el símbolo 1 como entrada, entonces permanece en el estado q0. *Pero si se recibe un 0 como entrada, entonces pasa al estado q1
El estado q1 tiene otras dos funciones posibles: δ(q1, 0) = q1 y δ(q1, 1) = q2	*La primera indica que estando en el estado q1, se recibe el símbolo 0 como entrada, por lo que permanece en el estado q1. *Pero si se recibe un 1 como entrada, entonces pasa al estado de aceptación q2.
En el estado q2, finalmente tenemos las funciones δ(q2, 0) = q2 y δ(q2, 1) = q2	*Indican que, tanto si se recibe un 0 como un 1, el autómata permanecerá en el estado de aceptación q0, pero si se recibe un 0 como entrada, entonces pasa al estado q1. *Si el autómata, al responder a los símbolos de entrada, logra alcanzar el estado de aceptación, entonces el conjunto de símbolos o cadenas recibidas constituye una fórmula válida para el autómata. En este ejemplo, las cadenas válidas son 100101, 10010, 10011, 1001, 101 y 01.

Mediante la figura 8, se describen las funciones de transición del autómata finito no determinista.

Figura 8

Funciones de transcripción del autómata finitos no determinista



En q0	*Es un estado de espera o búsqueda. El bucle con {0, 1} significa que el autómata puede procesar cualquier cantidad de caracteres iniciales sin moverse necesariamente de ahí
Tránsito q0 → q1	*Solo ocurre con un 0. Es el inicio de la secuencia final
Tránsito q1 → q2	*Solo ocurre con un 1. Es el cierre de la secuencia
En q2	*Una vez que llega aquí, el autómata ha cumplido su misión

Lenguaje Formal

Es una abstracción de las características generales de los lenguajes de programación que son procesados por una computadora. Está constituido por un conjunto de símbolos y algunas reglas de formación o combinación en cadenas denominadas entidades llamadas variables de programa, y estas entidades pueden agruparse en lo que se denomina oraciones o sentencias de programación. Por lo tanto, un lenguaje formal es un conjunto de cadenas aceptables según las reglas de formación, por ello, cualquier lenguaje de programación desarrollado debería tener las mismas características esenciales que un lenguaje formal.

Lenguajes formales y autómatas finitos

La teoría del lenguaje formal tiene diversas aplicaciones en el campo de la informática, para establecer una gramática formal, los matemáticos buscaron proporcionar una descripción rigurosa de las leyes gramaticales. Argumentaron que una descripción detallada de los lenguajes naturales (como el inglés, el hindi, etc.) facilitaría la traducción automática. Cabe considerar que, en 1956, Noam Chomsky presentó por primera vez un modelo formalizado de gramática, a pesar de su aparente falta de utilidad para expresar lenguajes naturales como el inglés, resultó eficaz para caracterizar lenguajes de programación.

Discusión

El análisis de la computación finita permite ratificar que los autómatas finitos constituyen el nivel más elemental dentro de la Jerarquía de Chomsky, siendo estrictamente equivalentes a las gramáticas y expresiones regulares. Según Allamanis et al. (2019), esta posición no es meramente taxonómica; pues la formalización de estas leyes gramaticales buscaba originalmente una descripción rigurosa de los lenguajes naturales, aunque su mayor eficacia terminó hallándose en la caracterización de lenguajes de programación. Por su parte Dhayalkar (2025) subraya que, esta carencia impide el reconocimiento de estructuras de conteo o dependencias anidadas como el caso paradigmático de $a^n b^n$, los resultados expuestos demuestran que es precisamente esta limitación la que otorga a los autómatas finitos una decidibilidad completa.

Un punto central de debate, es la equivalencia computacional entre el modelo AFD y el AFN, en este sentido, la investigación coincide con los postulados de Nicol y Frohme (2025), al verificar que el no determinismo, no incrementa el poder expresivo, sino que funciona como una herramienta

de abstracción para diseños más compactos que luego se transforman en un AFD funcional. En este caso, Cummins et al. (2023) argumenta que, esta relación es vital en el análisis del lenguaje, donde la capacidad de un autómata para ser minimizado en un único modelo óptimo, garantiza una eficiencia algorítmica superior en el diseño de compiladores.

Por otra parte, la robustez de los lenguajes formales se sustenta en sus propiedades de cerradura, pues, el hecho de que la clase de lenguajes definidos bajo este modelo sea cerrada ante operaciones como la unión, intersección, complemento y la estrella de Kleene, permite que los autómatas finitos superen el ámbito teórico para integrarse en aplicaciones críticas de ingeniería. Dicho argumento coincide con las perspectivas de (Seshia et al., 2018) quienes mencionan que, el modelado sistémico que posiciona a estos autómatas como piezas fundamentales en; verificación formal de sistemas (aprovechando que problemas como la vacuidad o la equivalencia son decidibles), y diseño de circuitos digitales (donde la finitud de los estados es una restricción física necesaria para los sistemas secuenciales), procesamiento de patrones (permitiendo el reconocimiento de cadenas de manera simple y rápida).

Finalmente, resulta imperativo destacar que, aunque los autómatas finitos representan el nivel más restringido de cómputo, su eficiencia y solidez matemática los mantienen como una herramienta insustituible. Lo que es confirmado por Hou (2021) quien argumenta que, otros modelos más potentes, como los autómatas de pila, son requeridos para estructuras recursivas, mientras que, la simplicidad de los autómatas finitos garantiza que cualquier lenguaje en esta clase pueda ser procesado con recursos finitos y predecibles. En conclusión, Broy et al. (2025) destacan que, la teoría de lenguajes formales no solo proporciona una abstracción de las características generales de la programación, sino que establece las reglas de formación indispensables para la informática moderna.

Conclusiones

Los autómatas finitos se consolidan como el nivel más elemental de la jerarquía de Chomsky y las gramáticas regulares, aunque nacieron para describir lenguajes naturales, su mayor eficacia se halló en la estructuración de la programación, esta base formal no es solo una clasificación, sino un pilar que define la esencia del procesamiento de datos, la misma que representa el punto de partida necesario para entender la comunicación técnica entre el hombre y la computadora.

Por otra parte, la ausencia de memoria estructurada, limita el reconocimiento de estructuras recursivas de largo alcance, sin embargo, esta restricción física es la que permite que problemas como la pertenencia sean decidibles; por ello, al ser modelos finitos, garantizan que cualquier lenguaje formal sea procesado con recursos técnicos predecibles, esto los hace preferibles sobre los autómatas de pila cuando la simplicidad algorítmica es la máxima prioridad.

En el caso de la equivalencia entre modelos deterministas y no deterministas, se demuestra que el diseño compacto no sacrifica el poder expresivo, pues, esta propiedad facilita la creación de autómatas mínimos únicos, optimizando el rendimiento en el análisis léxico, dicha eficiencia es fundamental en el diseño de compiladores, donde se requiere una velocidad de respuesta superior.

Finalmente, las propiedades de cerradura algebraica permiten integrar estos modelos en aplicaciones críticas como la verificación formal, su capacidad para el procesamiento de patrones y el diseño de circuitos digitales los vuelve piezas de ingeniería fundamentales, al ofrecer un reconocimiento de cadenas simple y rápido, facilitan el desarrollo de sistemas computacionales robustos. Por ello, en investigaciones posteriores, se recomienda priorizar el uso de autómatas finitos deterministas minimizados en el desarrollo de analizadores léxicos para compiladores y motores de búsqueda de patrones, pues al explotar su robustez matemática y eficiencia algorítmica, se logra un procesamiento de datos veloz que opera bajo parámetros de memoria estrictamente controlados.

Referencias Bibliográficas

Allamanis, M., Barr, E. T., Devanbu, P., & Sutton, C. (2019). A Survey of Machine Learning for Big Code and Naturalness. *ACM Computing Surveys*, *51*(4), 1-37. https://doi.org/10.1145/3212695

Broy, M., Brucker, A. D., Fantechi, A., Gleirscher, M., Havelund, K., Kuppe, M. A., Mendes, A., Platzer, A., Ringert, J. O., & Sullivan, A. (2025). Does Every Computer Scientist Need to Know Formal Methods? *Formal Aspects of Computing*, *37*(1), 1-17. https://doi.org/10.1145/3670795

Cummins, C., Seeker, V., Grubisic, D., Elhoushi, M., Liang, Y., Roziere, B., Gehring, J., Gloeckle, F., Hazelwood, K., Synnaeve, G., & Leather, H. (2023). *Large Language Models for Compiler Optimization* (Versión 1). arXiv. https://doi.org/10.48550/ARXIV.2309.07062

Dhayalkar, S. R. (2025). *Neural Networks as Universal Finite-State Machines: A Constructive Deterministic Finite Automaton Theory* (Versión 2). arXiv. https://doi.org/10.48550/ARXIV.2505.11694

Dora, J. R., Hluchý, L., & Staňo, M. (2025). In-Memory Shellcode Runner Detection in Internet of Things (IoT) Networks: A Lightweight Behavioral and Semantic Analysis Framework. *Sensors*, *25*(17), 5425. https://doi.org/10.3390/s25175425

Figueiras, P., Ioannou, G., Lambri, C., Reji, S., Mossali, E., Neviani, M. I., Koussouris, S., Bountouni, N., In Cugurra, M. D. B., Hellbach, R., Bibikas, D., O'Brien, P., Agostinho, C., & Jardim-Gonçalves, R. (2026). Roadmaps and Agendas for Research and Innovation in Artificial Intelligence, Data, and Robotics. En E. Curry, P. Piatkiewicz, F. Heintz, H. Vornhagen, A. N. Belbachir, E. Girardi, M. Schoenauer, & J. Röning (Eds.), *Artificial Intelligence, Data and Robotics* (pp. 547-579). Springer Nature Switzerland. https://doi.org/10.1007/978-3-032-10561-5_19

Gambhire, S., Panhalkar, A. R., Patil, D. H., Kedar, S., Kumar, M., & Panchal, B. Y. (2026). Leveraging formal languages and automata theory in natural language processing (NLP). *Journal of Discrete Mathematical Sciences & Cryptography*, *29*(2-B), 995-1004. https://doi.org/10.47974/JDMSC-2552

Hou, Z. (2021). Automata Theory and Formal Languages. En Z. Hou, *Fundamentals of Logic and Computation* (pp. 119-162). Springer International Publishing. https://doi.org/10.1007/978-3-030-87882-5_4

Kamble, U., Nalawade, S., Mahadik, T., Salunke, Y., Dedgaonkar, S., & Shelke, P. (2024). Survey of Application of Automata Theory in Natural Language Processing. *2024 ASU International Conference in Emerging Technologies for Sustainability and Intelligent Systems (ICETSIS)*, 1-4. https://doi.org/10.1109/ICETSIS61505.2024.10459556

Karakonstantis, I., & Xylomenos, G. (2026). A Review of Two-Dimensional Cellular Automata Models for Wildfire Simulation: Methods, Capabilities, and Limitations. *Fire*, *9*(3), 108. https://doi.org/10.3390/fire9030108

Kumar, A., Awasthy, N., Jose, A. S., Ramesh, G., Gupta, M., Srinija, K., & Najm, R. (2025). The Intersection of Algorithms, Automata Theory, and Formal Languages in Computational Theory. *2025 International Conference on Intelligent Control, Computing and Communications (IC3)*, 691-697. https://doi.org/10.1109/IC363308.2025.10956289

Morazán, M. T., & Minić, T. (2024). *Finite-State Automaton To/From Regular Expression Visualization*. https://doi.org/10.48550/ARXIV.2407.08088

Nagy, B. (2022). From Finite Automata to Fractal Automata – The Power of Recursion. En J. Durand-Lose & G. Vaszil (Eds.), *Machines, Computations, and Universality* (Vol. 13419, pp. 109-125). Springer International Publishing. https://doi.org/10.1007/978-3-031-13502-6_8

Negrini, L., Arceri, V., Cortesi, A., & Ferrara, P. (2024). TARSIS: An effective automata-based abstract domain for string analysis. *Journal of Software: Evolution and Process*, *36*(8), e2647. https://doi.org/10.1002/smr.2647

Nicol, J., & Frohme, M. (2025). *Deconstructing Subset Construction—Reducing While Determinizing* (Versión 2). arXiv. https://doi.org/10.48550/ARXIV.2505.10319

Postiglione, A. (2024). Finite State Automata on Multi-Word Units for Efficient Text-Mining. *Mathematics*, *12*(4), 506. https://doi.org/10.3390/math12040506

Seshia, S. A., Sharygina, N., & Tripakis, S. (2018). Modeling for Verification. En E. M. Clarke, T. A. Henzinger, H. Veith, & R. Bloem (Eds.), *Handbook of Model Checking* (pp. 75-105). Springer International Publishing. https://doi.org/10.1007/978-3-319-10575-8_3

Urbat, H., & Schröder, L. (2020). Automata Learning: An Algebraic Approach. *Proceedings of the 35th Annual ACM/IEEE Symposium on Logic in Computer Science*, 900-914. https://doi.org/10.1145/3373718.3394775

Yadav, A., Patel, A., & Shah, M. (2021). A comprehensive review on resolving ambiguities in natural language processing. *AI Open*, *2*, 85-92. https://doi.org/10.1016/j.aiopen.2021.05.001

Yokomori, T., & Okubo, F. (2021). Theory of reaction automata: A survey. *Journal of Membrane Computing*, *3*(1), 63-85. https://doi.org/10.1007/s41965-021-00070-6

Yue, J., Yan, Y., & Chen, Z. (2019). Language acceptability of finite automata based on theory of semi-tensor product of matrices. *Asian Journal of Control*, *21*(6), 2634-2643. https://doi.org/10.1002/asjc.2190

Agradecimiento:

N/A

Nota:

El artículo no es producto de una publicación anterior.