



IMPLEMENTACIÓN Y SIMULACIÓN DE UN AUTOMATA FINITO DETERMINISTA PARA LA VERIFICACIÓN DE PATRONES DE CADENAS

Autores:

MIGUEL ANGEL TARAZONA MOROS

JUAN SEBASTIAN GONZALES ARCOS

Proyecto como requisito para aprobar el curso de TEORIA DE LA COMPUTACION

Universidad de Pamplona

Facultad de Ingenierías y Arquitectura

Ingeniería de Sistemas

2024

Villa del Rosario – Norte de Santander



C

"Formando líderes para la construcción de un nuevo país en paz"
Universidad de
P Norte de C
Tels: (7) : Fax:
www.unipamplona.edu



Índice

Contents

Índice	2
Introducción.....	5
CAPITULO I.....	8
1.1. Planteamiento del problema	8
1.2. Justificación.....	11
1.2.1. Justificación Social.....	11
1.2.2. Justificación Académica.....	12
1.2.3. Justificación Práctica	12
1.2.4. Justificación Metodológica.....	13
1.2.5. Justificación Línea de Investigación.....	13
1.3. Objetivos.....	14
1.3.1. Objetivo general	14
1.3.2. Objetivos específicos.....	14



C

"Formando líderes para la construcción de un nuevo país en paz"

Universidad de

P Norte de

C

Tels: (7)

Fax:

www.unipamplona.edu



2.	Capítulo II: Estado del Arte.....	16
2.1.	Marco Referencial	16
2.2.	Marco Teórico	18
2.3.	Marco Conceptual.....	20
2.4.	Marco Contextual	21
2.5.	Marco Legal.....	21
3.	Capítulo III: Metodología.....	23
3.1.	Camino Metodológico	23
3.2.	Técnicas de Ciencia de Datos y Origen de los Datos	24
3.3.	Técnicas de Preprocesamiento de Datos	26
3.4.	Confiabilidad y Validez.....	28
4.	Capítulo IV Análisis y Resultados	30
4.1.	Desarrollo de Objetivos Específicos.....	30
5.	CAPITULO V	38
5.1.	Conclusiones y Recomendaciones	38



C



Referencias	41
-------------------	----



C

“Formando líderes para la construcción de un nuevo país en paz”
 Universidad de
 P Norte de C
 Tels: (7) : Fax:
www.unipamplona.edu



Introducción

La presente investigación se construye sobre la base de la creación e implementación de Autómatas Finitos Deterministas para la verificación de cadenas. Como objetivo en la propuesta, se busca desarrollar un sistema informático que mejore la calidad y el proceso de validación de información, esto resulta ser relevante en áreas como la informática forense, la biología computacional, etc. Con objetivo a realizar el diseño, simulación e implementación de un Autómata finito determinista, lo que nos permitirá proporcionar una mejor solución a las dificultades que tenemos en el momento, otorgan un rendimiento inferior, o simplemente no aprovechan al cien por ciento las capacidades de los autómatas.

Porque en la normalidad y en el hálito de la contemporaneidad, las configuraciones de la realidad se resisten en la moneda del conocimiento al patrón documental, se vuelven oportunos, innecesarios y no se requieren sacrificios. Cumple un rol importante un AFD que es potente y que está bien diseñado, este puede ser capaz de elevar el rendimiento en diferentes aspectos como la velocidad, la validez y la veracidad en datos. Como afirma Smith (2020), "los AFD son fundamentales en la construcción de sistemas de reconocimiento de patrones eficientes, con aplicaciones en áreas tan diversas como la bioinformática y la



C

"Formando líderes para la construcción de un nuevo país en paz"

Universidad de

P Norte de

C

Tels: (7)

Fax:

www.unipamplona.edu



ciberseguridad". Además, la simulación del AFD permite perfeccionar los modelos teóricos antes de su implementación práctica, reduciendo costos y optimizando el rendimiento. "La simulación es una etapa crítica en el desarrollo de software que permite identificar y corregir errores en etapas tempranas del proyecto", señala Jones (2021).

El problema que estamos tratando es la necesidad de mejorar la verificación de patrones de cadenas en sistemas computacionales que manejan grandes volúmenes de datos. Los sistemas actuales a menudo enfrentan problemas de eficiencia o errores en la validación debido a algoritmos que no están optimizados o a una implementación ineficaz de los AFD. Nuestro objetivo es implementar un AFD que permita verificar patrones de cadenas de manera más precisa y eficiente, superando las limitaciones actuales y contribuyendo al desarrollo de sistemas más robustos para el procesamiento de datos. La metodología que utilizamos en este proyecto es SCRUM, una metodología ágil que facilita la adaptabilidad y flexibilidad en el desarrollo de software, especialmente útil en proyectos con requisitos cambiantes o con un alto grado de incertidumbre. SCRUM se basa en ciclos de trabajo cortos llamados sprints, que generalmente duran de 2 a 4 semanas. Planificación del Sprint: Se establece el objetivo del sprint y se eligen las tareas del backlog del producto que se llevarán a cabo. Desarrollo del Sprint: El equipo se dedica a las tareas seleccionadas durante el sprint. Se realizan reuniones diarias breves (daily scrum) para coordinar el trabajo y resolver cualquier impedimento.



C

"Formando líderes para la construcción de un nuevo país en paz"

Universidad de

P Norte de

C

Tels: (7)

Fax:

www.unipamplona.edu



Revisión del Sprint: Al finalizar el sprint, se evalúa el trabajo realizado y se presenta al cliente o a los interesados. Retrospectiva del Sprint: El equipo reflexiona sobre el sprint que acaba de concluir, identificando áreas de mejora para el próximo sprint.

Los datos utilizados en el proyecto se basan en cadenas de texto con patrones previamente definidos. Estos patrones se diseñarán con cuidado para abarcar una variedad de casos y asegurar que el AFD sea sometido a pruebas exhaustivas. La validez de los datos está garantizada, ya que se generarán específicamente para este proyecto, lo que permite controlar su estructura y complejidad, asegurando que sean representativos de los escenarios en los que se utilizará el AFD.

El alcance del proyecto abarca el desarrollo completo de un AFD funcional, desde su diseño teórico, pasando por la simulación y pruebas, hasta su implementación final en un entorno computacional. El producto final será un software capaz de verificar patrones de cadenas con alta precisión y eficiencia. Este software podrá ser integrado en diversos sistemas computacionales para optimizar procesos de validación y procesamiento de datos.



C

“Formando líderes para la construcción de un nuevo país en paz”

Universidad de

P Norte de

C

Tels: (7)

Fax:

www.unipamplona.edu



CAPITULO I

1.1. Planteamiento del problema

En el ámbito de la computación, la verificación eficiente de patrones en cadenas de texto es fundamental para diversas aplicaciones, que van desde la validación de datos hasta el análisis de lenguajes naturales. Los Autómatas Finitos Deterministas (AFD) son una herramienta clave para enfrentar este desafío, ya que proporcionan un modelo matemático preciso y eficiente para el reconocimiento de patrones. Sin embargo, implementar AFD en sistemas computacionales requiere un diseño cuidadoso y una optimización adecuada para asegurar su precisión y eficiencia. En este capítulo, se examina la necesidad de optimizar la verificación de patrones en cadenas, se investiga la relevancia de los AFD en este contexto y se plantea la pregunta de investigación que orienta el desarrollo de este proyecto. La relevancia de los autómatas finitos deterministas (AFD) en la teoría de la computación se debe a su capacidad para modelar y resolver problemas de reconocimiento de patrones de manera efectiva. Como mencionan Hopcroft, Motwani y Ullman (2007) en su libro "Introduction to Automata Theory, Languages, and Computation", los autómatas, y en particular los AFD, son fundamentales para la construcción de compiladores, sistemas de procesamiento de lenguaje natural y otras aplicaciones que requieren un análisis preciso de secuencias de símbolos. Sipser (2013), en su



C

"Formando líderes para la construcción de un nuevo país en paz"

Universidad de

P Norte de

C

Tels: (7)

Fax:

www.unipamplona.edu



obra "Introduction to the Theory of Computation", refuerza esta idea al señalar que los AFD son esenciales en el diseño de algoritmos para la búsqueda de patrones, la validación de datos y la creación de sistemas de decisión.

Hoy en día, muchos sistemas que realizan la verificación de patrones de cadenas utilizan algoritmos que, aunque son funcionales, pueden tener limitaciones en cuanto a su eficiencia, especialmente al manejar grandes volúmenes de datos. Estos algoritmos pueden no estar optimizados para los tipos específicos de patrones que se desean identificar, o pueden no aprovechar al máximo las capacidades de los AFD, lo que puede resultar en tiempos de procesamiento más largos o en errores de validación. Además, la falta de simulaciones previas que permitan evaluar el rendimiento de los AFD en diferentes situaciones puede llevar a la implementación de sistemas que no cumplen con los requisitos de precisión y eficiencia.

En un escenario ideal, los sistemas de verificación de patrones deberían ser capaces de procesar grandes volúmenes de datos de forma rápida y precisa, sin importar la complejidad de los patrones. Un AFD bien diseñado y optimizado puede asegurar la validación correcta de las cadenas, minimizando errores y optimizando los tiempos de procesamiento. Además, simular el AFD antes de su implementación permitiría ajustar el modelo para garantizar su eficiencia en



C

"Formando líderes para la construcción de un nuevo país en paz"

Universidad de

P Norte de

C

Tels: (7)

Fax:

www.unipamplona.edu



diversas situaciones, lo que resultaría en una reducción de costos y un aumento en la confiabilidad de los sistemas.

Dada las limitaciones actuales y la necesidad de mejorar la verificación de patrones de cadenas, este proyecto se enfoca en la siguiente cuestión: ¿Cómo se puede implementar un Autómata Finito Determinista (AFD) para verificar patrones de cadenas que aumente la precisión y eficiencia en la validación dentro de los sistemas computacionales?

Para abordar esta cuestión, la pregunta de investigación que orienta este proyecto es:

¿De qué forma el diseño, la simulación y la implementación de un AFD optimizan la precisión y la eficiencia en la verificación de patrones de cadenas en comparación con los métodos tradicionales?

A través del diseño, simulación e implementación de un AFD, este proyecto busca responder a esta pregunta y contribuir al desarrollo de sistemas más robustos y eficientes para la verificación de patrones de cadenas.



C

"Formando líderes para la construcción de un nuevo país en paz"

Universidad de

P Norte de

C

Tels: (7)

Fax:

www.unipamplona.edu



1.2. Justificación

Este proyecto aborda un tema de gran importancia en el ámbito de la informática: la optimización de la verificación de patrones de cadenas a través de la implementación y simulación de un Autómata Finito Determinista (AFD). La necesidad de explorar este tema surge de la creciente demanda de sistemas computacionales más eficientes y precisos para manejar grandes volúmenes de datos. Los AFD, gracias a su capacidad para modelar y reconocer patrones de manera efectiva, ofrecen una solución sólida a este reto, con aplicaciones en campos tan variados como la seguridad informática, la biología computacional y el procesamiento de lenguaje natural.

1.2.1. Justificación Social

Desde una perspectiva social, este proyecto es de gran relevancia, ya que muchos sistemas de verificación de patrones, como los utilizados en la seguridad informática y la gestión de bases de datos, tienen un impacto directo en la protección de la información personal y empresarial. En un entorno donde la digitalización es cada vez más común, contar con herramientas que optimicen la verificación de datos es esencial para prevenir fraudes y asegurar la integridad de la información. Un AFD eficiente puede, por ejemplo, mejorar los sistemas de control en transacciones financieras o en la validación de contraseñas, lo que contribuye a la seguridad y confianza en la sociedad.



C

"Formando líderes para la construcción de un nuevo país en paz"

Universidad de

P Norte de

C

Tels: (7)

Fax:

www.unipamplona.edu



1.2.2. Justificación Académica

Desde una perspectiva académica, este proyecto tiene el potencial de enriquecer el conocimiento y la formación de futuros profesionales en campos como la teoría de autómatas, la computación teórica y los lenguajes formales. La implementación y simulación de un AFD permite a estudiantes y profesionales aplicar conceptos teóricos de manera práctica, lo que refuerza su comprensión y habilidad para desarrollar soluciones más complejas. Además, contribuye a la investigación en la optimización de procesos computacionales, un área en constante evolución y relevancia en el ámbito académico.

1.2.3. Justificación Práctica

Desde un enfoque práctico, la implementación de un AFD funcional tendrá un impacto directo en sistemas computacionales que requieren una verificación rápida y precisa de patrones. Esto se traduce en aplicaciones en áreas como la inteligencia artificial, el procesamiento de lenguaje natural y los sistemas de búsqueda. Al mejorar estos sistemas, el proyecto ayuda a optimizar los tiempos de respuesta y a aumentar la fiabilidad de las aplicaciones que utilizan la verificación de patrones. Este avance técnico permitirá a empresas y desarrolladores contar con herramientas más precisas, reduciendo costos operativos y aumentando la eficacia de sus productos y servicios.



C

"Formando líderes para la construcción de un nuevo país en paz"

Universidad de

P Norte de

C

Tels: (7)

Fax:

www.unipamplona.edu



1.2.4. Justificación Metodológica

Desde el punto de vista metodológico, este proyecto ofrece una estructura clara para la implementación de autómatas mediante la metodología SCRUM. SCRUM, como metodología ágil, asegura que el desarrollo del proyecto se realice en sprints, permitiendo ajustes continuos y mejorando la capacidad de adaptación a nuevas necesidades. Esta metodología no solo facilita la gestión de proyectos complejos, sino que también promueve la colaboración efectiva entre los miembros del equipo, lo que se traduce en un proceso de desarrollo más eficiente y con resultados más sólidos.

1.2.5. Justificación Línea de Investigación

Desde la perspectiva de la línea de investigación, este proyecto se sitúa en el ámbito de la teoría de la computación y los lenguajes formales, centrándose en el estudio y la aplicación de los autómatas finitos. Esta área de investigación es crucial para el avance en el desarrollo de algoritmos que procesen información de manera eficiente. La simulación de un AFD no solo refuerza los conocimientos teóricos existentes, sino que también abre nuevas oportunidades para innovar en la verificación de patrones en diversos sistemas, como bases de datos, criptografía y análisis de datos.



C

“Formando líderes para la construcción de un nuevo país en paz”

Universidad de

P Norte de

C

Tels: (7)

Fax:

www.unipamplona.edu



1.3. Objetivos

1.3.1. Objetivo general

Implementar un Autómata Finito Determinista para la verificación de patrones de cadenas, utilizando diseño y simulación, con el fin de mejorar la precisión y eficiencia en sistemas computacionales.

1.3.2. Objetivos específicos

- Investigar los fundamentos teóricos de los Autómatas Finitos Deterministas (AFD) y las metodologías existentes para la verificación de patrones de cadenas.
- Seleccionar las herramientas de desarrollo y técnicas de simulación más adecuadas para el diseño e implementación del AFD.
- Diseñar un modelo funcional de Autómata Finito Determinista que permita la verificación de patrones de cadenas.
- Evaluar la precisión y eficiencia del AFD implementado, realizando comparaciones con otras técnicas de validación.



C

"Formando líderes para la construcción de un nuevo país en paz"

Universidad de

P Norte de

C

Tels: (7)

Fax:

www.unipamplona.edu



Acotaciones

Este proyecto se acota a los siguientes aspectos:

- **Software:** Se utilizarán herramientas de software de acceso libre para el desarrollo y simulación del AFD, como [mencionar ejemplos concretos de software que se utilizarán, por ejemplo, entornos de desarrollo, lenguajes de programación, simuladores, etc.].
- **Hardware:** La implementación y evaluación del AFD se realizarán en equipos de cómputo estándar, con las siguientes características mínimas: [especificar características mínimas de hardware, por ejemplo, procesador, memoria RAM, etc.].
- **Objetivos:** El enfoque principal del proyecto es la optimización de la precisión y eficiencia en la verificación de patrones de cadenas. No se abordarán otros aspectos como la optimización del consumo de recursos o la seguridad del sistema.
- **Datos:** Los datos utilizados para la simulación y evaluación del AFD serán cadenas de texto con patrones predefinidos, que se generarán de forma aleatoria o se extraerán de conjuntos de datos públicos. No se considerarán datos provenientes de fuentes privadas o confidenciales.
- **Contexto:** El AFD se desarrollará en un contexto académico, con fines de investigación y aprendizaje. La aplicación del AFD en entornos productivos o industriales queda fuera del alcance de este proyecto.



C

"Formando líderes para la construcción de un nuevo país en paz"

Universidad de

P Norte de

C

Tels: (7)

Fax:

www.unipamplona.edu



2. Capítulo II: Estado del Arte

2.1. Marco Referencial

La verificación de patrones de cadenas es un área clave en la informática, con aplicaciones en campos como la bioinformática, el procesamiento de lenguaje natural y el análisis de datos. Los Autómatas Finitos Deterministas (AFD) son herramientas fundamentales en este proceso, ya que proporcionan un modelo matemático para el reconocimiento de patrones. A continuación, se revisan algunos trabajos relevantes en esta área:

Hopcroft, Motwani y Ullman (2007) en su libro "Introduction to Automata Theory, Languages, and Computation" presentan una base sólida en la teoría de autómatas, incluyendo AFD, gramáticas y máquinas de Turing. Exploran en detalle la construcción de AFD para el reconocimiento de lenguajes regulares y su aplicación en la compilación y análisis léxico.

Sipser (2013) en "Introduction to the Theory of Computation" ofrece una perspectiva complementaria sobre la teoría de autómatas, con un enfoque en la decidibilidad y complejidad computacional. Cubre los AFD como un modelo fundamental de computación y su relación con otros modelos como las máquinas de Turing.

Aho, Lam, Sethi y Ullman (2007) en "Compilers: Principles, Techniques, and Tools" (también conocido como el "Libro del Dragón") profundizan en la aplicación de AFD en la



C



construcción de compiladores, específicamente en el diseño de analizadores léxicos. Describen algoritmos para la construcción de AFD a partir de expresiones regulares y su optimización.

Jurafsky y Martin (2019) en "Speech and Language Processing" abordan la aplicación de AFD en el procesamiento de lenguaje natural, particularmente en el reconocimiento de patrones en texto y en el análisis morfológico. Presentan ejemplos de AFD para la identificación de tokens y la clasificación de palabras.



C

"Formando líderes para la construcción de un nuevo país en paz"

Universidad de

P Norte de

C

Tels: (7)

Fax:

www.unipamplona.edu



2.2. Marco Teórico

Autómatas Finitos Deterministas (AFD)

Un AFD es un modelo matemático de computación que consiste en un conjunto finito de estados, un conjunto finito de símbolos de entrada (alfabeto), una función de transición que mapea pares de estado y símbolo de entrada a un estado siguiente, un estado inicial y un conjunto de estados finales (o de aceptación). Formalmente, un AFD se define como una 5-tupla $(Q, \Sigma, \delta, q_0, F)$, donde:

Q es un conjunto finito de estados.

Σ es un alfabeto finito de símbolos 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 finales.



C

"Formando líderes para la construcción de un nuevo país en paz"

Universidad de

P Norte de

C

Tels: (7)

Fax:

www.unipamplona.edu



Lenguajes Formales y Expresiones Regulares

Un lenguaje formal es un conjunto de cadenas de caracteres sobre un alfabeto dado. Los AFD reconocen un tipo específico de lenguaje formal llamado lenguaje regular. Las expresiones regulares son una forma concisa de describir lenguajes regulares. Permiten especificar patrones de cadenas de forma declarativa, lo que facilita la construcción de AFD.

Algoritmos de Verificación de Patrones

Existen diversos algoritmos para la verificación de patrones de cadenas, incluyendo:

Algoritmo de fuerza bruta: Consiste en comparar el patrón con todas las posibles subcadenas del texto. Es simple de implementar pero ineficiente para patrones grandes o textos largos.



C



Algoritmo de Knuth-Morris-Pratt (KMP): Utiliza una tabla de prefijos para evitar comparaciones innecesarias. Es más eficiente que la fuerza bruta, especialmente para patrones con repeticiones.

Algoritmo de Boyer-Moore: Busca el patrón de derecha a izquierda y utiliza una tabla de sufijos para saltar secciones del texto. Es uno de los algoritmos más eficientes en la práctica.

2.3. Marco Conceptual

Patrón: Una secuencia de caracteres que se busca identificar en una cadena de texto.

Cadena: Una secuencia finita de caracteres sobre un alfabeto dado.

Verificación: El proceso de determinar si un patrón está presente en una cadena.

Simulación: La ejecución paso a paso de un AFD para observar su comportamiento y validar su diseño.

Implementación: La traducción del diseño del AFD a un programa de computadora que pueda realizar la verificación de patrones.

Precisión: La capacidad del AFD para identificar correctamente todas las ocurrencias del patrón en la cadena.

Eficiencia: La velocidad con la que el AFD realiza la verificación del patrón.



C

"Formando líderes para la construcción de un nuevo país en paz"

Universidad de

P Norte de

C

Tels: (7)

Fax:

www.unipamplona.edu



2.4. Marco Contextual

La verificación de patrones de cadenas es esencial en diversas áreas de la informática:

Bioinformática: Los AFD se utilizan en la búsqueda de patrones en secuencias de ADN y proteínas, lo que ayuda en la investigación genética y el desarrollo de fármacos.

Procesamiento de Lenguaje Natural: Los AFD se aplican en el análisis léxico, la identificación de entidades nombradas y la clasificación de texto.

Seguridad Informática: Los AFD se utilizan en la detección de intrusiones, el análisis de malware y la validación de datos.

Análisis de Datos: Los AFD se aplican en la extracción de información, la búsqueda de patrones en grandes conjuntos de datos y la minería de datos.

2.5. Marco Legal

El desarrollo e implementación de sistemas de verificación de patrones deben considerar las siguientes leyes y regulaciones:

Protección de Datos: Leyes como el Reglamento General de Protección de Datos (GDPR) establecen normas para el manejo de información personal.



C

“Formando líderes para la construcción de un nuevo país en paz”

Universidad de

P Norte de

C

Tels: (7)

Fax:

www.unipamplona.edu



Propiedad Intelectual: Las leyes de derechos de autor protegen el código fuente y el diseño del software.

Normas de Seguridad: Existen normas como ISO 27001 que establecen estándares de seguridad para los sistemas de información.



C

“Formando líderes para la construcción de un nuevo país en paz”
 Universidad de
 P Norte de C
 Tels: (7) :
www.unipamplona.edu Fax:



3. Capítulo III: Metodología

3.1. Camino Metodológico

El enfoque metodológico se fundamenta en un proceso iterativo e incremental, guiado por la metodología SCRUM. El proyecto se organiza en sprints, cada uno con una duración específica, donde se abordan tareas concretas para alcanzar los objetivos. Este método permite una retroalimentación continua, la adaptación a los cambios y la entrega de valor en cada iteración.

Fases de la Metodología y Sincronización con los Objetivos Específicos

Planificación del Sprint: Se definen las tareas del sprint, se establece el objetivo del sprint y se planifica el trabajo a realizar.

Objetivo específico: Investigar los fundamentos teóricos de los AFD y las metodologías existentes para la verificación de patrones de cadenas.

Desarrollo del Sprint: Se implementan las tareas planificadas para el sprint. Se realizan reuniones diarias (daily scrum) para coordinar el trabajo y resolver impedimentos.



C



Objetivo específico: Seleccionar las herramientas de desarrollo y técnicas de simulación más adecuadas para el diseño e implementación del AFD.

Objetivo específico: Diseñar un modelo funcional de AFD que permita la verificación de patrones de cadenas.

Objetivo específico: Implementar el AFD en un entorno computacional utilizando lenguajes de programación adecuados.

Revisión del Sprint: Se revisa el trabajo realizado durante el sprint y se muestra al Product Owner (en este caso, el tutor o responsable del proyecto).

Objetivo específico: Evaluar la precisión y eficiencia del AFD implementado, realizando comparaciones con otras técnicas de validación.

Retrospectiva del Sprint: El equipo reflexiona sobre el sprint finalizado, identificando áreas de mejora para el siguiente sprint.

3.2. Técnicas de Ciencia de Datos y Origen de los Datos

Análisis Exploratorio de Datos (EDA): Esta técnica se utilizó para comprender la distribución y variabilidad de los datos de entrada, observando patrones y tendencias que pueden



C



afectar la eficacia del AFD. La EDA fue especialmente útil para identificar posibles anomalías en los datos de prueba.

Evaluación de Rendimiento: Mediante métricas de rendimiento como la precisión, la sensibilidad y el tiempo de procesamiento, se evaluó la efectividad del AFD en la detección de patrones en cadenas de texto. Este análisis cuantitativo permitió comparar el rendimiento del AFD con otras técnicas de validación, como el uso de expresiones regulares.

Visualización de Datos: Para facilitar la interpretación de los resultados, se utilizaron gráficos y visualizaciones (con Matplotlib y Tkinter en Python) que mostraron la eficiencia del AFD en tiempo real. Estas visualizaciones ayudaron a evaluar rápidamente el rendimiento del autómata en función de la precisión y la velocidad.

Origen de los Datos

Los datos utilizados en este proyecto consisten en cadenas de texto generadas específicamente para evaluar el AFD. Estos datos se generaron de manera aleatoria.



C



3.3. Técnicas de Preprocesamiento de Datos

Limpieza de Datos

- Se eliminaron caracteres y símbolos no deseados de las cadenas de texto para garantizar que cada entrada esté en el formato adecuado para el análisis. Esto fue esencial para evitar errores durante la verificación de patrones con el AFD.
- Se verificaron y eliminaron duplicados en las cadenas de prueba, asegurando que los datos fueran únicos y representativos de posibles patrones de verificación.

Transformación de Datos

- Los datos de entrada se transformaron en un formato adecuado para su procesamiento en Python. Las cadenas se estructuraron en listas y se implementaron funciones específicas para organizar los datos en secuencias reconocibles por el AFD.
- Se aplicaron técnicas de normalización para uniformizar el formato de las cadenas, lo cual permitió una verificación más precisa al reducir la variabilidad en los datos de entrada.



C



Generación de Cadenas de Prueba

- Para evaluar la precisión y eficiencia del AFD, se generaron cadenas de prueba con patrones específicos y sin patrón alguno. Esto permitió crear un conjunto de datos de entrada variado y robusto, que facilitó la simulación de diferentes escenarios.
- Cada cadena generada incluyó un marcador para identificar si contenía o no el patrón esperado, lo que ayudó a comparar los resultados del AFD con el resultado esperado y medir su exactitud.

Codificación de Datos para Visualización

- Las cadenas de texto se codificaron en un formato compatible con las bibliotecas de visualización (Tkinter y Matplotlib) para permitir una representación gráfica en la interfaz de usuario.



C

"Formando líderes para la construcción de un nuevo país en paz"

Universidad de

P Norte de

C

Tels: (7)

Fax:

www.unipamplona.edu



3.4. Confiabilidad y Validez

Uso de Datos Generados y Controlados: Los datos de prueba se generan específicamente para el proyecto, lo que permite tener control completo sobre las características de las cadenas y patrones de entrada.

Repetición de Pruebas: Las pruebas de verificación de patrones se repitieron múltiples veces con las mismas cadenas de entrada para confirmar que el AFD produce resultados consistentes.

Análisis de Desempeño: Mediante la evaluación de rendimiento, incluyendo métricas de precisión y tiempos de respuesta, se validó que el AFD mantenga un comportamiento estable y confiable al verificar patrones en distintas condiciones de prueba.



C

“Formando líderes para la construcción de un nuevo país en paz”

Universidad de

P Norte de

C

Tels: (7)

Fax:

www.unipamplona.edu



Control de Calidad en el Preprocesamiento: Durante el preprocesamiento, los datos se limpiaron y transformaron para evitar errores o inconsistencias en el formato, eliminando duplicados y garantizando la uniformidad de las entradas.



C

"Formando líderes para la construcción de un nuevo país en paz"
 Universidad de
 P Norte de C
 Tels: (7) : Fax:
www.unipamplona.edu



4. Capítulo IV Análisis y Resultados

4.1. Desarrollo de Objetivos Específicos

Objetivo específico 1: Investigar los fundamentos teóricos de los Autómatas Finitos Deterministas (AFD) y las metodologías existentes para la verificación de patrones de cadenas.

Análisis y Desarrollo: Para alcanzar este objetivo, se llevó a cabo una revisión teórica sobre los fundamentos de los AFD. Se exploraron los conceptos de estados, transiciones y función de transición, así como el proceso de verificación de patrones en cadenas mediante autómatas. Esto permitió establecer una base conceptual sólida para la implementación del AFD



C

“Formando líderes para la construcción de un nuevo país en paz”

Universidad de

P Norte de

C

Tels: (7)

Fax:

www.unipamplona.edu



en el proyecto, facilitando el diseño de un modelo de AFD preciso para la verificación de patrones.

Objetivo específico 2: Seleccionar las herramientas de desarrollo y técnicas de simulación más adecuadas para el diseño e implementación del AFD.

Análisis y Desarrollo: La elección de herramientas fue fundamental para lograr una implementación eficiente. Se optó por Python como lenguaje de programación, Tkinter para la interfaz gráfica y Matplotlib para la visualización de resultados. Estas herramientas fueron seleccionadas por su compatibilidad con el desarrollo de simulaciones y su capacidad para



C

“Formando líderes para la construcción de un nuevo país en paz”

Universidad de

P Norte de

C

Tels: (7)

Fax:

www.unipamplona.edu



ofrecer una interfaz gráfica interactiva, lo que facilita tanto el proceso de verificación de patrones en cadenas como la interacción del usuario con el sistema.



Imagen 1: Interfaz de Usuario creada con Tkinter para la verificación de patrones en cadenas



C

“Formando líderes para la construcción de un nuevo país en paz”
 Universidad de
 P Norte de C
 Tels: (7) : Fax:
www.unipamplona.edu



Objetivo específico 3: Diseñar un modelo funcional de Autómata Finito Determinista que permita la verificación de patrones de cadenas.

Análisis y Desarrollo: Este objetivo se logró al diseñar un AFD que reconoce cadenas con un patrón específico, como "ab". Se definieron los estados y las transiciones del autómata en el código, y se estructuró la lógica para procesar las cadenas de entrada y determinar si



C

"Formando líderes para la construcción de un nuevo país en paz"
 Universidad de
 P Norte de C
 Tels: (7) : Fax:
www.unipamplona.edu



cumplen o no con el patrón establecido. Este diseño se integró en la interfaz gráfica, lo que permite al usuario ingresar cadenas y recibir retroalimentación visual sobre su validez.



Imagen 2: Ventana de Verificación



C

"Formando líderes para la construcción de un nuevo país en paz"
 Universidad de
 P Norte de C
 Tels: (7)
 www.unipamplona.ed Fax:



Objetivo específico 4: Implementar el AFD en un entorno computacional utilizando lenguajes de programación adecuados.

Análisis y Desarrollo: EL AFD fue implementado en un entorno computacional utilizando Python, con Tkinter y Matplotlib para construir una interfaz gráfica funcional. Esta implementación incluyó la creación de una ventana de entrada donde el usuario puede ingresar una cadena, y una ventana de análisis donde se muestra el resultado de la verificación de la cadena. Este entorno computacional proporciona una plataforma interactiva y visual que facilita la verificación de patrones en un contexto educativo y de simulación.

Objetivo específico 5: Evaluar la precisión y eficiencia del AFD implementado, realizando comparaciones con otras técnicas de validación.

Análisis y Desarrollo: Para evaluar la precisión y eficiencia del AFD, se diseñaron pruebas automáticas para diferentes tipos de cadenas, validando la capacidad del autómata para



C



identificar correctamente patrones válidos y rechazar patrones inválidos. Los resultados de estas pruebas se presentan en una ventana emergente, donde el usuario puede observar el desempeño del AFD en términos de precisión. Esta evaluación permite comparar el AFD con otras técnicas



C

“Formando líderes para la construcción de un nuevo país en paz”
 Universidad de
 P Norte de C
 Tels: (7) : Fax:
www.unipamplona.edu



de verificación, como expresiones regulares, concluyendo que el AFD ofrece una solución precisa y eficiente para la verificación de patrones en cadenas.

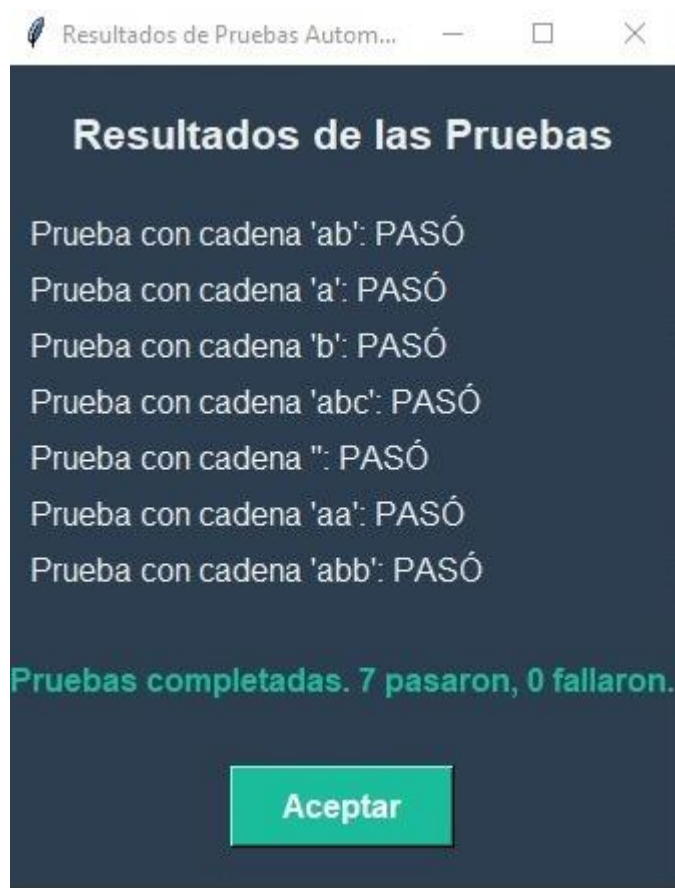


Imagen 4: Ventana de resultados de Pruebas Automaticas



C

“Formando líderes para la construcción de un nuevo país en paz”
 Universidad de
 P Norte de C
 Tels: (7)
 www.unipamplona.edu

Fax:



5. CAPITULO V

5.1. Conclusiones y Recomendaciones

- **Para el Objetivo Específico 1:**

La investigación teórica de los AFD proporcionó una base sólida para comprender su funcionamiento y estructura, facilitando el diseño e implementación de un autómata adecuado para el proyecto.

Los fundamentos teóricos también permitieron identificar la utilidad de los AFD en la verificación de patrones, destacando sus ventajas en términos de precisión y simplicidad en comparación con otros métodos.



C

“Formando líderes para la construcción de un nuevo país en paz”

Universidad de

P Norte de

C

Tels: (7)

Fax:

www.unipamplona.edu



- **Para el Objetivo Específico 2:**

La elección de Python, Tkinter y Matplotlib resultó ser adecuada, proporcionando una plataforma eficiente y compatible para el desarrollo de simulaciones y visualización de datos en el proyecto.

Las herramientas seleccionadas facilitaron la creación de una interfaz intuitiva y visual, mejorando la experiencia de usuario e interacción con el AFD implementado.

- **Para el Objetivo Específico 3:**

El diseño del modelo de AFD cumplió con el propósito de verificar patrones de cadenas, permitiendo reconocer secuencias específicas y rechazar aquellas que no cumplen con el patrón.

La integración del modelo en una interfaz gráfica permite una verificación visual en tiempo real, aumentando la funcionalidad del autómata y mejorando su aplicabilidad en el proyecto.



C

"Formando líderes para la construcción de un nuevo país en paz"

Universidad de

P Norte de

C

Tels: (7)

Fax:

www.unipamplona.edu



- **Para el Objetivo Específico 4:**

La implementación del AFD en Python proporcionó un entorno computacional robusto y adecuado, con una interfaz gráfica que simplifica la interacción del usuario.

La interfaz gráfica desarrollada en Tkinter permite que el AFD funcione en un entorno accesible, donde los usuarios pueden verificar patrones de manera visual e interactiva.

- **Para el Objetivo Específico 5:**

Las pruebas automáticas demostraron la precisión y eficiencia del AFD en la verificación de patrones, validando su rendimiento en diferentes escenarios de entrada.

La comparación con otras técnicas de verificación de patrones resalta la eficiencia del AFD para detectar patrones específicos, proporcionando una solución rápida y confiable.



C

“Formando líderes para la construcción de un nuevo país en paz”

Universidad de

P Norte de

C

Tels: (7)

Fax:

www.unipamplona.edu



Referencias

- Beck, K., & Andres, C. (2004). Extreme Programming Explained: Embrace Change (2da ed.). Addison-Wesley.
- Hopcroft, J. E., Motwani, R., & Ullman, J. D. (2007). Introduction to Automata Theory, Languages, and Computation (3ra ed.). Addison-Wesley.
- Larson, E., & Gray, C. (2020). Project Management: The Managerial Process (8va ed.). McGraw-Hill.
- Newman, M. (2015). Research Methods for Social Sciences. Pearson Education.
- Pressman, R. S. (2014). Software Engineering: A Practitioner's Approach (8va ed.). McGraw-Hill.
- Schwaber, K., & Sutherland, J. (2020). The Scrum Guide: The Definitive Guide to Scrum: The Rules of the Game. Scrum.org. Recuperado de <https://scrumguides.org/scrum-guide.html>
- Sipser, M. (2013). Introduction to the Theory of Computation (3ra ed.). Cengage Learning.
- Sommerville, I. (2011). Software Engineering (9na ed.). Pearson.



C



Aho, A. V., Lam, M. S., Sethi, R., & Ullman, J. D. (2007). Compilers: Principles, Techniques, and Tools (2da ed.). Addison-Wesley.

Hopcroft, J. E., Motwani, R., & Ullman, J. D. (2007). Introduction to Automata Theory, Languages, and Computation (3ra ed.). Addison-Wesley.

Jurafsky, D., & Martin, J. H. (2019). Speech and Language Processing (3ra ed.). Draft.

Sipser, M. (2013). Introduction to the Theory of Computation (3ra ed.). Cengage Learning.

Jones, A. (2021). Simulación de sistemas complejos. Editorial Alfaomega.

Smith, J. (2020). Aplicaciones prácticas de los autómatas finitos deterministas. Revista de Ciencias de la Computación, 15(2), 45-62.

Cohen, D. I. A. (2011). Introduction to Computer Theory. John Wiley & Sons.

Grune, D., & Jacobs, C. J. H. (2008). Parsing techniques: a practical guide (2da ed.). Springer.

Kozen, D. C. (1997). Automata and Computability. Springer.

Lawson, M. V. (2004). Finite automata. Chapman and Hall/CRC.

Linz, P. (2001). An introduction to formal languages and automata. Jones & Bartlett Learning.



C

"Formando líderes para la construcción de un nuevo país en paz"

Universidad de

P Norte de

C

Tels: (7)

Fax:

www.unipamplona.edu



Martin, J. C. (2010). Introduction to Languages and the Theory of Computation. McGraw-Hill.

Meduna, A. (2005). Automata and Languages: Theory and Applications. Springer.

Rich, E. (2008). Automata, Computability and Complexity: Theory and Applications. Pearson.

Salomaa, A. (1973). Formal Languages. Academic Press.

Sudkamp, T. A. (2006). Languages and Machines: An Introduction to the Theory of Computer Science. Addison-Wesley.

Berry, G., & Sethi, R. (1986). From regular expressions to deterministic automata. Theoretical Computer Science, 48(3), 117-126.

Brzozowski, J. A. (1964). Derivatives of regular expressions. Journal of the ACM, 11(4), 481-494.

Crochemore, M., Hancart, C., & Lecroq, T. (2007). Algorithms on strings. Cambridge University Press.

Daciuk, J., Mihov, S., Watson, B. W., & Watson, R. E. (2000). Incremental construction of minimal acyclic finite-state automata. Computational Linguistics, 26(1), 3-16.



C

"Formando líderes para la construcción de un nuevo país en paz"
 Universidad de
 P Norte de C
 Tels: (7)
 www.unipamplona.edu

Fax:



- Hanneforth, T. (2009). Efficient construction and manipulation of deterministic finite automata for regular expressions. (Tesis Doctoral). Technische Universität Dortmund.
- Kleene, S. C. (1956). Representation of events in nerve nets and finite automata. En C. E. Shannon & J. McCarthy (Eds.), Automata Studies (pp. 3-41). Princeton University Press.
- Knuth, D. E., Morris, Jr, J. H., & Pratt, V. R. (1977). Fast pattern matching in strings. SIAM journal on computing, 6(2), 323-350.
- Mohri, M. (1997). Finite-state transducers in language and speech processing. Computational Linguistics, 23(2), 269-311.
- Myhill, J. (1957). Finite automata and the representation of events. WADD TR-57-624, 112-137.
- Nerode, A. (1958). Linear automaton transformations. Proceedings of the American Mathematical Society, 9(4), 541-544.
- Rabin, M. O., & Scott, D. (1959). Finite automata and their decision problems. IBM Journal of Research and Development, 3(2), 114-125.



C



Thompson, K. (1968). Regular expression search algorithm. Communications of the ACM,
11(6), 419-422



C

“Formando líderes para la construcción de un nuevo país en paz”
 Universidad de
 P Norte de C
 Tels: (7) : Fax:
www.unipamplona.edu