

PRÁCTICA 2

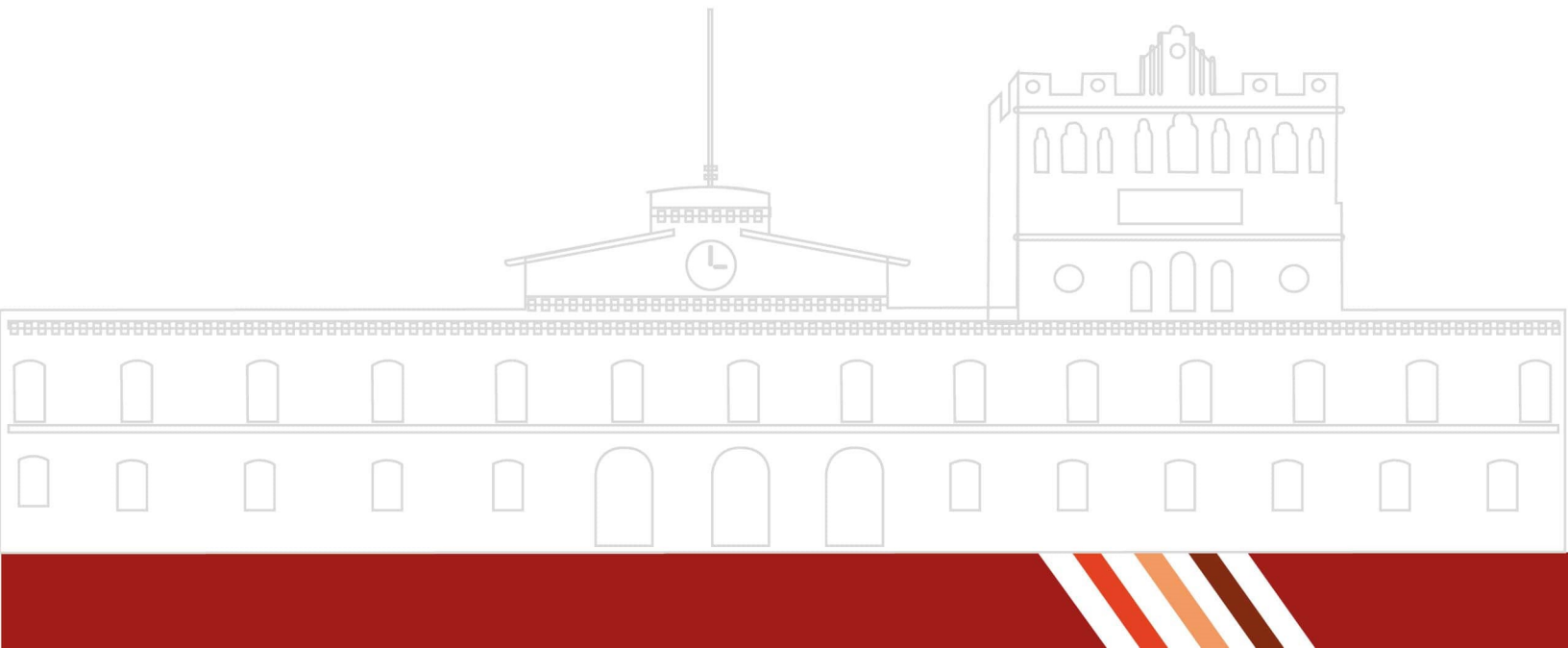
Licenciatura en Ciencias Computacionales

ALUMNO:

Sánchez Lara José de Jesús

SEMESTRE/GRUPO

6 ° 3



Introducción

Los autómatas finitos deterministas (AFD) y no deterministas (AFND) son modelos matemáticos fundamentales en la teoría de la computación y la automatización de procesos. Estas estructuras permiten representar y analizar el comportamiento de sistemas. En esta práctica se busca representar y crear autómatas finitos deterministas y no deterministas en base a enunciados propuestos.

Objetivo General

Comprender y aplicar los conceptos de autómatas finitos deterministas y no deterministas a través del diseño, implementación y simulación de modelos mediante un software.

Marco Teórico

Un autómata finito es una máquina abstracta. Existen dos tipos principales:

2.1.1 Autómata Finito Determinista (AFD)

Es un modelo en el que para cada estado y símbolo de entrada existe una única transición.

2.1.2 Autómata Finito No Determinista (AFND)

En este modelo, un estado puede tener múltiples transiciones para un mismo símbolo de entrada, de ahí que no este determinado que transición debería de seguir.

2.1.3 Cadena vacía

Representada por ϵ o por λ es una representación de la cadena vacía dentro de los autómatas y representa la cadena de longitud cero, que es distinto a un lenguaje vacío.

Herramientas

Para el desarrollo de esta práctica se emplearán las siguiente herramientas:

2.2.1 Software

Automaton Simulator: Herramienta para el diseño y simulación de autómatas finitos.

2.2.2 Videos:

Lista de reproducción en youtube sobre autómatas del canal Codemath.

Ejercicios

Ejercicio 1. Obtenga un Autómata Finito Determinista (AFD) dado el lenguaje definido en el alfabeto $\Sigma = \{0, 1\}$, que acepte el conjunto de palabras que inician en “0”.

Tupla del AFD

- Conjunto de estados: $Q = \{a, b\}$
- Alfabeto: $\Sigma = \{0, 1\}$
- Funciones de transición: δ
 - $\delta(a, 0) = b$
 - $\delta(a, 1) = a$
 - $\delta(b, 0) = b$
 - $\delta(b, 1) = b$
- Estado inicial: a
- Estados finales: $F = \{b\}$

Diagrama de transiciones

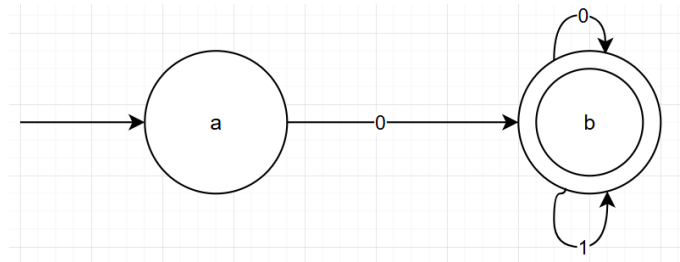


Figure 1: Diagrama de transiciones del AFD propuesto.

Tabla de transiciones

Estado	0	1
a	b	a
b	b	b

Simulación

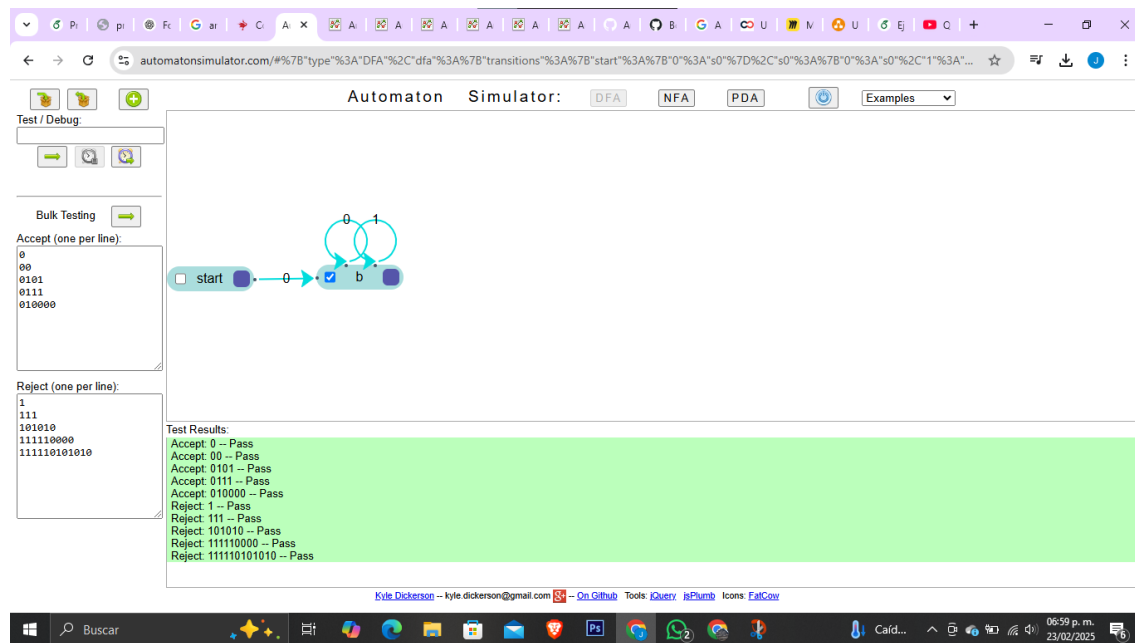


Figure 2: Simulación.

Palabras aceptadas y rechazadas

Test Results:

```
Accept: 0 -- Pass
Accept: 00 -- Pass
Accept: 0101 -- Pass
Accept: 0111 -- Pass
Accept: 010000 -- Pass
Reject: 1 -- Pass
Reject: 111 -- Pass
Reject: 101010 -- Pass
Reject: 111110000 -- Pass
Reject: 111110101010 -- Pass
```

Figure 3: Palabras rechazadas y aceptadas.

Ejercicio 2. Obtenga un Autómata Finito Determinista (AFD) dado el lenguaje definido en el alfabeto $\Sigma = 0, 1$, que acepte el conjunto de palabras que terminan en “1”.

Tupla del AFD

- Conjunto de estados: $Q = \{a, b\}$
- Alfabeto: $\Sigma = \{0, 1\}$
- Funciones de transición: δ
 - $\delta(a, 0) = a$
 - $\delta(a, 1) = b$
 - $\delta(b, 0) = a$
 - $\delta(b, 1) = b$
- Estado inicial: a
- Estados finales: $F = \{b\}$

Diagrama de transiciones

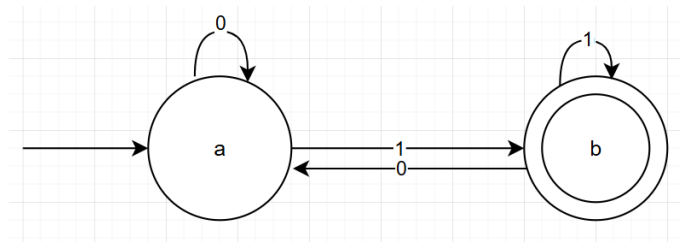


Figure 4: Diagrama de transiciones del AFD propuesto.

Tabla de transiciones

Estado	0	1
a	a	b
b	a	b

Simulación

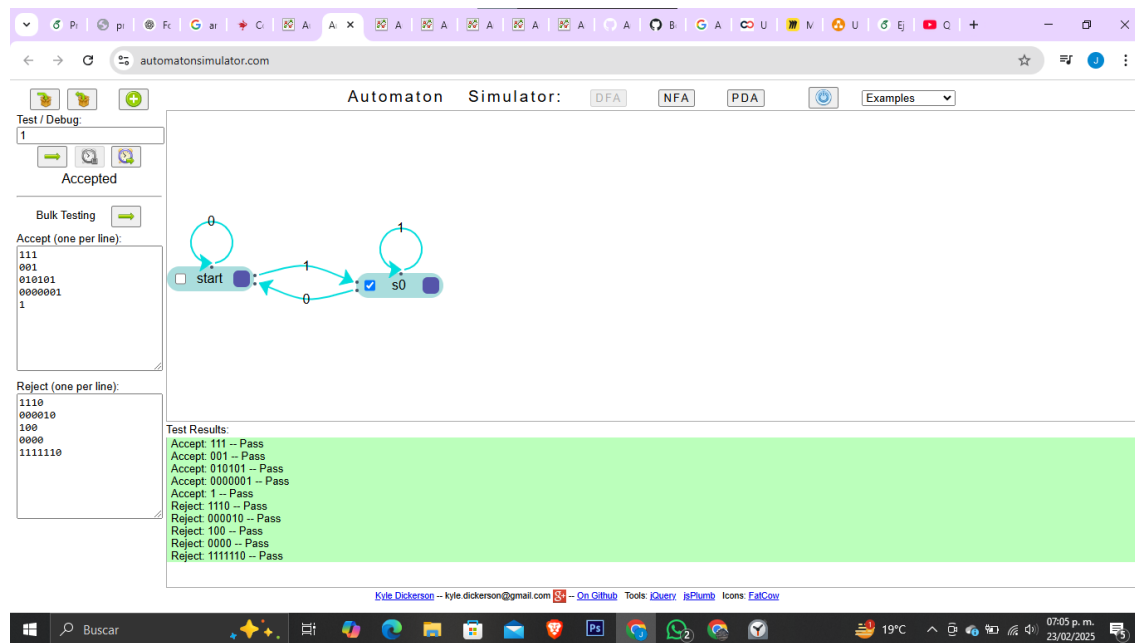


Figure 5: Simulación.

Palabras aceptadas y rechazadas

Test Results:

```
Accept: 111 -- Pass
Accept: 001 -- Pass
Accept: 010101 -- Pass
Accept: 0000001 -- Pass
Accept: 1 -- Pass
Reject: 1110 -- Pass
Reject: 000010 -- Pass
Reject: 100 -- Pass
Reject: 0000 -- Pass
Reject: 1111110 -- Pass
```

Figure 6: Palabras rechazadas y aceptadas.

Ejercicio 3. . Obtenga un Autómata Finito Determinista (AFD) dado el lenguaje definido en el alfabeto $\Sigma = \{0, 1\}$, que acepte el conjunto de palabras que contienen la subcadena “01”.

Tupla del AFD

- Conjunto de estados: $Q = \{a, b, c\}$
- Alfabeto: $\Sigma = \{0, 1\}$
- Funciones de transición: δ
 - $\delta(a, 0) = b$
 - $\delta(a, 1) = a$
 - $\delta(b, 0) = b$
 - $\delta(b, 1) = c$
 - $\delta(c, 0) = c$
 - $\delta(c, 1) = c$
- Estado inicial: a
- Estados finales: $F = \{c\}$

Diagrama de transiciones

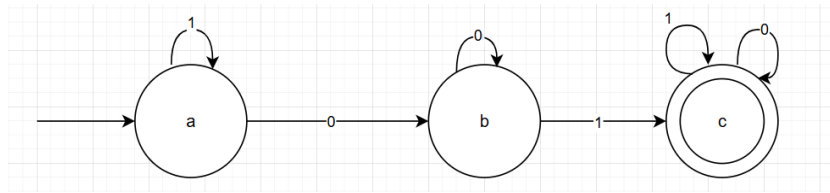


Figure 7: Diagrama de transiciones del AFD propuesto.

Tabla de transiciones

Estado	0	1
a	b	a
b	b	c
c	c	c

Simulación

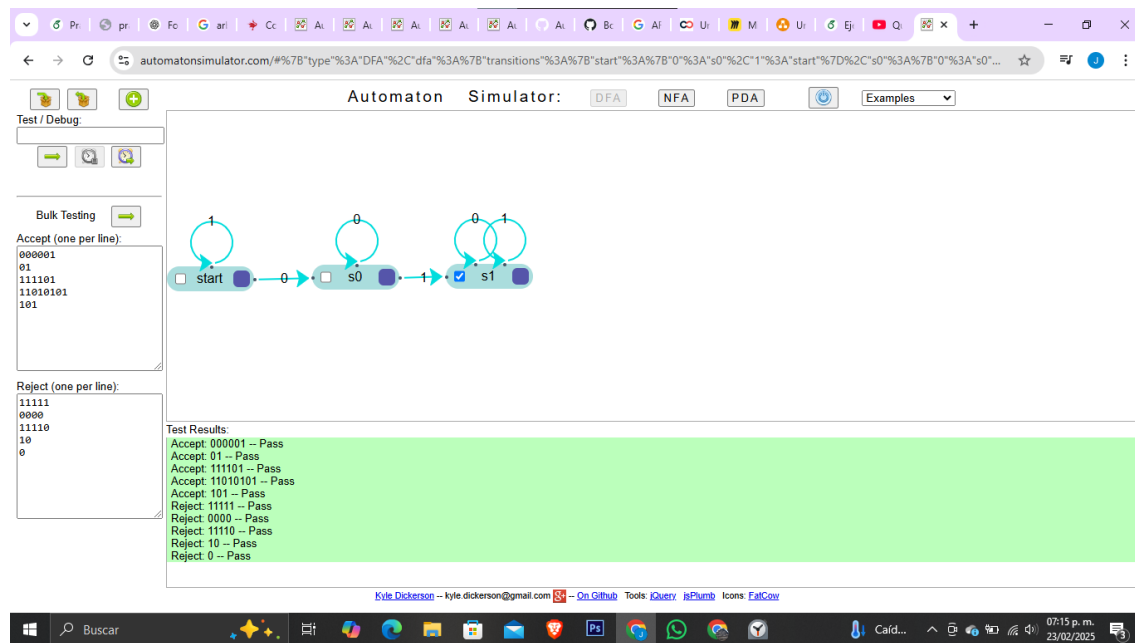


Figure 8: Simulación.

Palabras aceptadas y rechazadas

Test Results:

```
Accept: 000001 -- Pass
Accept: 01 -- Pass
Accept: 111101 -- Pass
Accept: 11010101 -- Pass
Accept: 101 -- Pass
Reject: 11111 -- Pass
Reject: 0000 -- Pass
Reject: 11110 -- Pass
Reject: 10 -- Pass
Reject: 0 -- Pass
```

Figure 9: Palabras rechazadas y aceptadas.

Ejercicio 4. Obtenga un Autómata Finito Determinista (AFD) dado el lenguaje definido en el alfabeto $\Sigma = \{0, 1\}$, que acepte el conjunto de palabras que no contienen la subcadena “01”.

Tupla del AFD

- Conjunto de estados: $Q = \{a, b, c\}$
- Alfabeto: $\Sigma = \{0, 1\}$
- Funciones de transición: δ
 - $\delta(a, 0) = b$
 - $\delta(a, 1) = a$
 - $\delta(b, 0) = b$
 - $\delta(b, 1) = c$
 - $\delta(c, 0) = c$
 - $\delta(c, 1) = c$
- Estado inicial: a
- Estados finales: $F = \{a, b\}$

Diagrama de transiciones

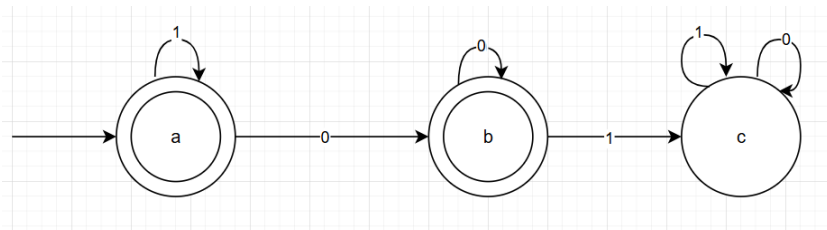


Figure 10: Diagrama de transiciones del AFD propuesto.

Tabla de transiciones

Estado	0	1
a	b	a
b	b	c
c	c	c

Simulación

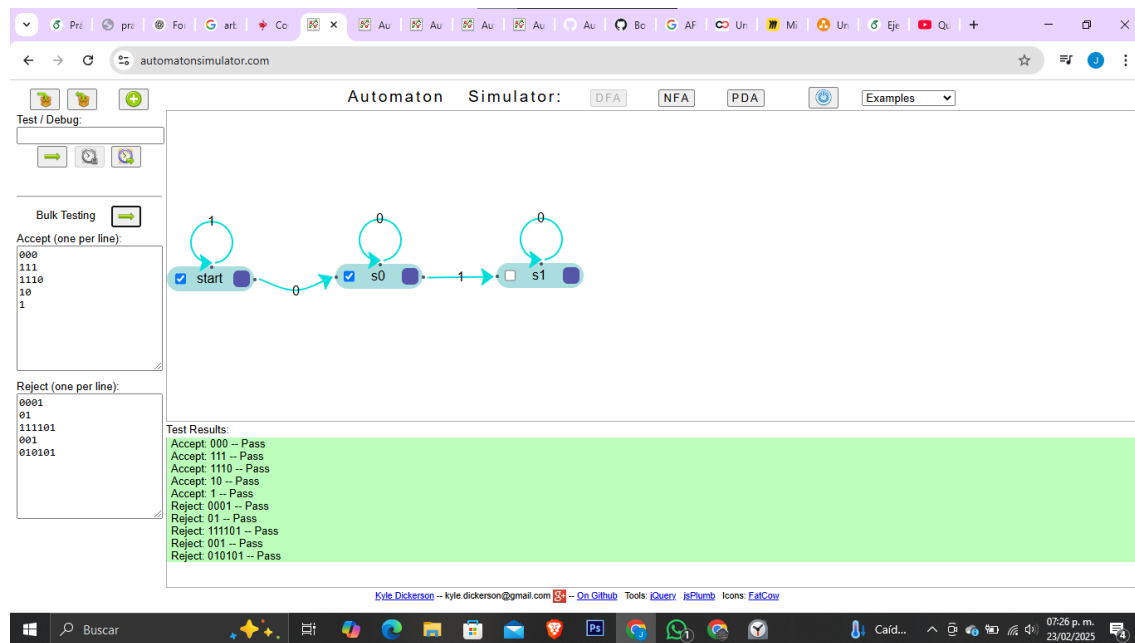


Figure 11: Simulación.

Palabras aceptadas y rechazadas

Test Results:

Accept: 000 -- Pass
Accept: 111 -- Pass
Accept: 1110 -- Pass
Accept: 10 -- Pass
Accept: 1 -- Pass
Reject: 0001 -- Pass
Reject: 01 -- Pass
Reject: 111101 -- Pass
Reject: 001 -- Pass
Reject: 010101 -- Pass

Figure 12: Palabras rechazadas y aceptadas.

Ejercicio 5. Obtenga un Autómata Finito Determinista (AFD) dado el lenguaje definido en el alfabeto $\Sigma = \{a, b, c\}$, que acepte el conjunto de palabras que inician con la subcadena “ac” o terminan con la subcadena “ab”.

Tupla del AFD

- Conjunto de estados: $Q = \{1, 2, 3, 4, 5, 6\}$
- Alfabeto: $\Sigma = \{a, b, c\}$
- Funciones de transición: δ
 - $\delta(1, a) = 2$
 - $\delta(1, b) = 4$
 - $\delta(1, c) = 4$
 - $\delta(2, a) = 5$
 - $\delta(2, b) = 6$
 - $\delta(2, c) = 3$
 - $\delta(3, a) = 3$
 - $\delta(3, b) = 3$
 - $\delta(3, c) = 3$
 - $\delta(4, a) = 5$
 - $\delta(4, b) = 4$
 - $\delta(4, c) = 4$
 - $\delta(5, a) = 5$
 - $\delta(5, b) = 6$
 - $\delta(5, c) = 4$
 - $\delta(6, a) = 5$
 - $\delta(6, b) = 4$
 - $\delta(6, c) = 4$
- Estado inicial: 1
- Estados finales: $F = \{3, 6\}$

Diagrama de transiciones

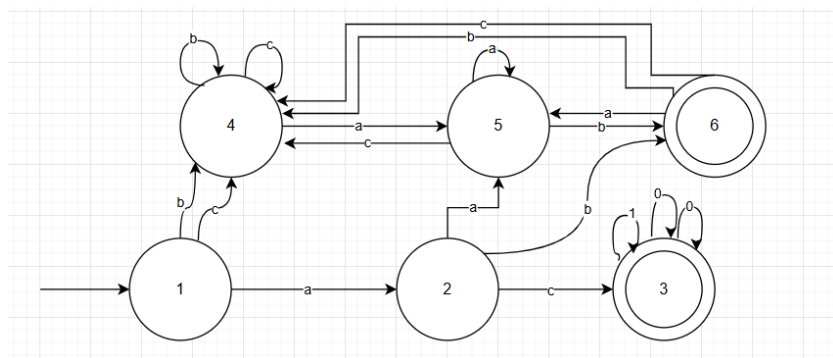


Figure 13: Diagrama de transiciones del AFD propuesto.

Tabla de transiciones

Estado	a	b	c
1	2	4	4
2	5	6	3
3	3	3	3
4	5	4	4
5	5	6	4
6	5	4	4

Simulación

Automaton Simulator: DFA NFA PDA Examples

Test / Debug

Bulk Testing

Accept (one per line):

```
ac
ab
acab
ccccccabbbab
cccccc
```

Reject (one per line):

```
aa
bb
aaaaab
bbbbba
aabb
```

Test Results:

```
Accept: ac -- Pass
Accept: ab -- Pass
Accept: acab -- Pass
Accept: cccccabbbab -- Pass
Accept: ccccc -- Pass
Reject: aa -- Pass
Reject: bb -- Pass
Reject: aaaaab -- Pass
Reject: bbbba -- Pass
Reject: aabb -- Pass
```

Figure 14: Simulación.

Palabras aceptadas y rechazadas

Test Results:

```
Accept: ac -- Pass
Accept: ab -- Pass
Accept: acab -- Pass
Accept: ccccccab -- Pass
Accept: accccc -- Pass
Reject: aa -- Pass
Reject: bb -- Pass
Reject: aaaacb -- Pass
Reject: bbbbbac -- Pass
Reject: aabbc -- Pass
```

Figure 15: Palabras rechazadas y aceptadas.

Ejercicio 6. Obtenga un Autómata Finito Determinista (AFD) dado el lenguaje definido en el alfabeto $\Sigma = \{a, b, c\}$, que acepte el conjunto de palabras que inician con la subcadena “ac” y no terminan con la subcadena “ab”.

Tupla del AFD

- Conjunto de estados: $Q = \{0, 1, 2, 3, 4, 5\}$
- Alfabeto: $\Sigma = \{a, b, c\}$
- Funciones de transición: δ
 - $\delta(0, a) = 1$
 - $\delta(0, b) = 4$
 - $\delta(0, c) = 4$
 - $\delta(1, a) = 4$
 - $\delta(1, b) = 4$
 - $\delta(1, c) = 2$
 - $\delta(2, a) = 3$
 - $\delta(2, b) = 2$
 - $\delta(2, c) = 2$
 - $\delta(3, a) = 3$
 - $\delta(3, b) = 5$
 - $\delta(3, c) = 2$
 - $\delta(4, a) = 4$
 - $\delta(4, b) = 4$
 - $\delta(4, c) = 4$
 - $\delta(5, a) = 3$
 - $\delta(5, b) = 3$
 - $\delta(5, c) = 3$
- Estado inicial: 0
- Estados finales: $F = \{2, 3\}$

Diagrama de transiciones

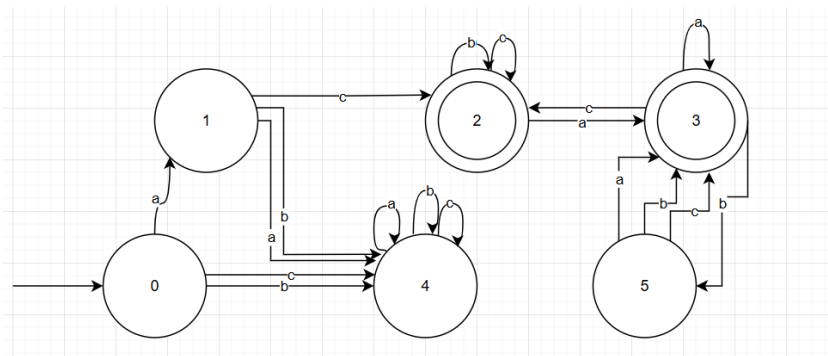


Figure 16: Diagrama de transiciones del AFD propuesto.

Tabla de transiciones

Estado	a	b	c
0	1	4	4
1	4	4	2
2	3	2	2
3	3	5	2
4	4	4	4
5	3	3	3

Simulación

Automaton Simulator: DFA NFA PDA Examples

Test / Debug:

Bulk Testing

Accept (one per line):

```

acb
acac
acba
acbb
acbc

```

Reject (one per line):

```

abc
bac
acab
acbab
aaa
bbb
cccc

```

Test Results:

```

Accept: acb -- Pass
Accept: acac -- Pass
Accept: acba -- Pass
Accept: acbb -- Pass
Accept: acbc -- Pass
Reject: abc -- Pass
Reject: bac -- Pass
Reject: acab -- Pass
Reject: acbab -- Pass
Reject: aaa -- Pass
Reject: bbb -- Pass
Reject: cccc -- Pass

```

Figure 17: Simulación.

Palabras aceptadas y rechazadas

Test Results:

```
Accept: acb -- Pass
Accept: acac -- Pass
Accept: acba -- Pass
Accept: accb -- Pass
Accept: acabc -- Pass
Reject: abc -- Pass
Reject: bac -- Pass
Reject: acab -- Pass
Reject: acbab -- Pass
Reject: aaa -- Pass
Reject: bbb -- Pass
Reject: cccc -- Pass
```

Figure 18: Palabras rechazadas y aceptadas.

Ejercicio 7. Obtenga un Autómata Finito Determinista (AFD) dado el lenguaje definido en el alfabeto $\Sigma = \{a, b, c\}$, que acepte el conjunto de palabras que inician con la subcadena “ac” o no terminan con la subcadena “ab”.

Tupla del AFD

- Conjunto de estados: $Q = \{0, 1, 2, 3, 4, 5\}$
- Alfabeto: $\Sigma = \{a, b, c\}$
- Funciones de transición: δ
 - $\delta(0, a) = 1$
 - $\delta(0, b) = 3$
 - $\delta(0, c) = 3$
 - $\delta(1, a) = 4$
 - $\delta(1, b) = 5$
 - $\delta(1, c) = 2$
 - $\delta(2, a) = 2$
 - $\delta(2, b) = 2$
 - $\delta(2, c) = 2$
 - $\delta(3, a) = 4$
 - $\delta(3, b) = 3$
 - $\delta(3, c) = 3$
 - $\delta(4, a) = 4$
 - $\delta(4, b) = 5$
 - $\delta(4, c) = 3$
 - $\delta(5, a) = 4$
 - $\delta(5, b) = 3$
 - $\delta(5, c) = 3$
- Estado inicial: 0
- Estados finales: $F = \{1, 2, 3, 4\}$

Diagrama de transiciones

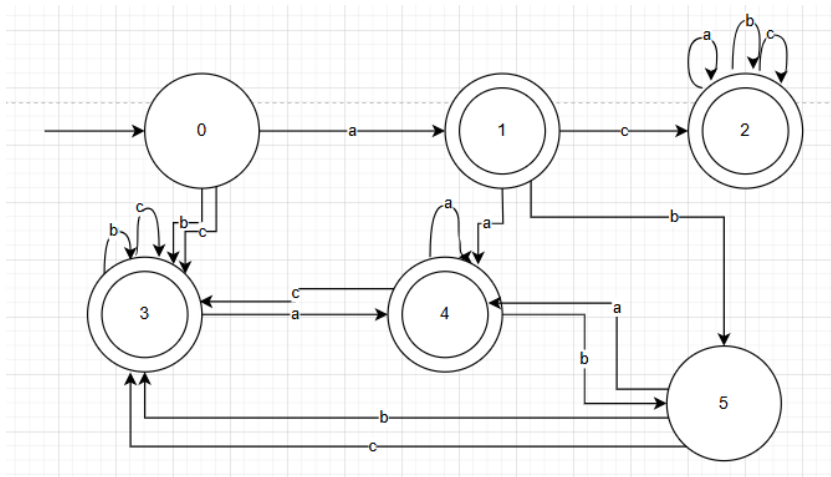


Figure 19: Diagrama de transiciones del AFD propuesto.

Tabla de transiciones

Estado	a	b	c
0	1	3	3
1	4	5	2
2	2	2	2
3	4	3	3
4	4	5	3
5	4	3	3

Simulación

Automaton Simulator: DFA NFA PDA Examples

Test / Debug

Bulk Testing

Accept (one per line):

```
ac
aa
aaaaac
accbbbb
acbb
```

Reject (one per line):

```
aaaaab
ab
abbab
bbbab
bbcab
```

Test Results:

```
Accept: ac -- Pass
Accept: aa -- Pass
Accept: aaaaac -- Pass
Accept: accbbbb -- Pass
Accept: acbb -- Pass
Accept: [Empty String] -- Pass
Reject: aaaaab -- Pass
Reject: ab -- Pass
Reject: abbab -- Pass
Reject: bbbab -- Pass
Reject: bbbcab -- Pass
```

Figure 20: Simulación.

Palabras aceptadas y rechazadas

```
Test Results:  
Accept: ac -- Pass  
Accept: aa -- Pass  
Accept: aaaaac -- Pass  
Accept: acccbbb -- Pass  
Accept: accb -- Pass  
Accept: [Empty String] -- Pass  
Reject: aaaaab -- Pass  
Reject: ab -- Pass  
Reject: abbab -- Pass  
Reject: bbbbab -- Pass  
Reject: bbbcab -- Pass
```

Figure 21: Palabras rechazadas y aceptadas.

Ejercicio 8. Obtenga un Autómata Finito Determinista (AFD) dado el lenguaje definido en el alfabeto $\Sigma = a, b, c$, que acepte el conjunto de palabras que inician con la subcadena “ac” o no terminan con la subcadena “ab”.

Tupla del AFD

- Conjunto de estados: $Q = \{0, 1, 2, 3, 4, 5, 6\}$
- Alfabeto: $\Sigma = \{a, b, c\}$
- Funciones de transición: δ
 - $\delta(0, a) = 1$
 - $\delta(0, b) = 5$
 - $\delta(0, c) = 5$
 - $\delta(1, a) = 3$
 - $\delta(1, b) = 4$
 - $\delta(1, c) = 2$
 - $\delta(2, a) = 2$
 - $\delta(2, b) = 2$
 - $\delta(2, c) = 2$
 - $\delta(3, a) = 3$
 - $\delta(3, b) = 4$
 - $\delta(3, c) = 5$
 - $\delta(4, a) = 6$
 - $\delta(4, b) = 5$
 - $\delta(4, c) = 6$
 - $\delta(5, a) = 6$
 - $\delta(5, b) = 5$
 - $\delta(5, c) = 5$
 - $\delta(6, a) = 6$
 - $\delta(6, b) = 4$
 - $\delta(6, c) = 5$
- Estado inicial: 0
- Estados finales: $F = \{1, 3, 5, 6\}$

Diagrama de transiciones

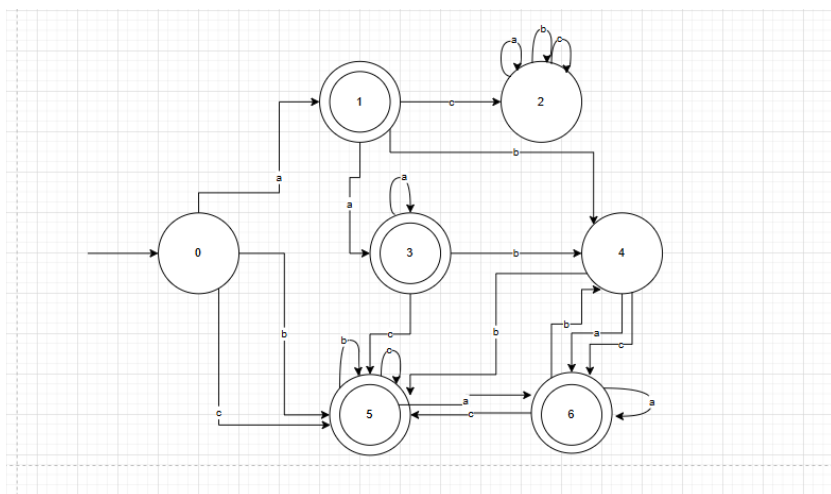


Figure 22: Diagrama de transiciones del AFD propuesto.

Tabla de transiciones

Estado	a	b	c
0	1	5	5
1	3	4	2
2	2	2	2
3	3	4	5
4	6	5	6
5	6	5	5
6	6	4	5

Simulación

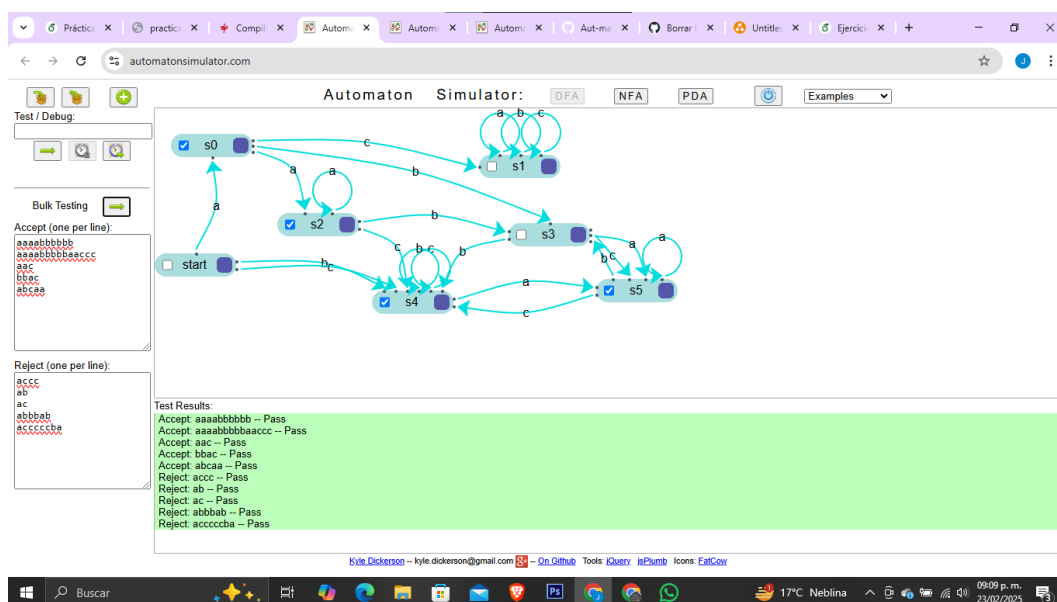


Figure 23: Simulación.

Palabras aceptadas y rechazadas

Test Results:
Accept: aaaabbbbb -- Pass
Accept: aaaabbbbbaacc -- Pass
Accept: aac -- Pass
Accept: bbac -- Pass
Accept: abcaa -- Pass
Reject: accc -- Pass
Reject: ab -- Pass
Reject: ac -- Pass
Reject: abbbab -- Pass
Reject: accccba -- Pass

Figure 24: Palabras rechazadas y aceptadas.

Ejercicio 9. Obtenga un Autómata Finito No Determinista (AFND) dado el lenguaje definido en el alfabeto $\Sigma = 0, 1$, que acepte el conjunto de palabras que no contienen a la subcadena “01”.

Tupla del AFND

- Conjunto de estados: $Q = \{a, b, c, d\}$
- Alfabeto: $\Sigma = \{0, 1\}$
- Funciones de transición: δ
 - $\delta(a, 0) = \{c\}$
 - $\delta(a, 1) = \{b, c\}$
 - $\delta(b, 0) = \{c\}$
 - $\delta(b, 1) = \{b\}$
 - $\delta(c, 0) = \{c\}$
 - $\delta(c, 1) = \{d\}$
 - $\delta(d, 0) = \{d\}$
 - $\delta(d, 1) = \{d\}$
- Estado inicial: a
- Estados finales: $F = \{b, c\}$

Diagrama de transiciones

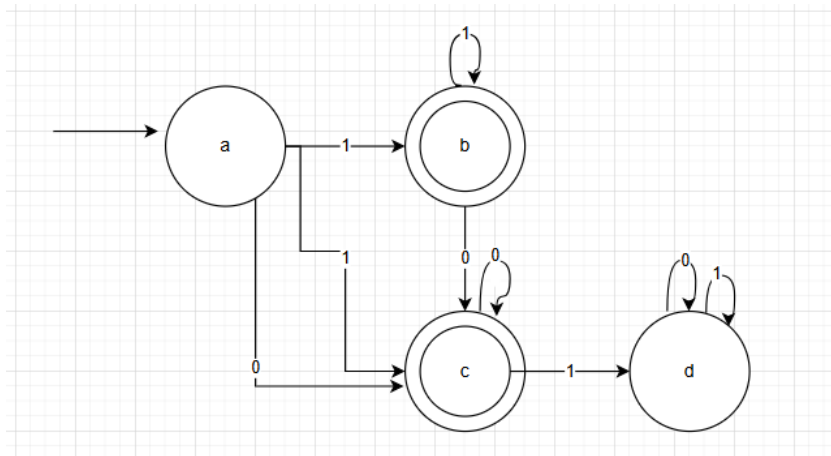


Figure 25: Diagrama de transiciones del AFND propuesto.

Tabla de transiciones

Estado	0	1
a	{c}	{b,c}
b	{c}	{b}
c	{c}	{d}
d	{d}	{d}

Simulación

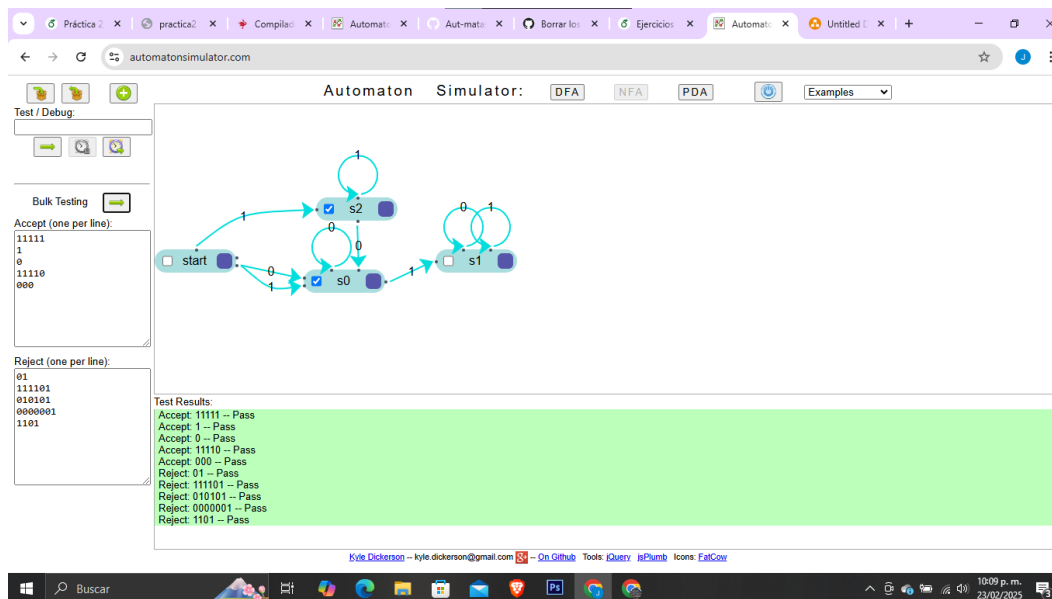


Figure 26: Simulación del AFND.

Palabras aceptadas y rechazadas

Test Results:

```

Accept: 11111 -- Pass
Accept: 1 -- Pass
Accept: 0 -- Pass
Accept: 11110 -- Pass
Accept: 000 -- Pass
Reject: 01 -- Pass
Reject: 111101 -- Pass
Reject: 010101 -- Pass
Reject: 0000001 -- Pass
Reject: 1101 -- Pass
  
```

Figure 27: Palabras rechazadas y aceptadas en el AFND.

Ejercicio 10. Obtenga un Autómata Finito No Determinista (AFND) dado el lenguaje definido en el alfabeto $\Sigma = a, b, c$, que acepte el conjunto de palabras que inician en la subcadena “ac” y terminan en la subcadena “ab”.

Tupla del AFD

- Conjunto de estados: $Q = \{0, 1, 2, 3, 4, 5\}$
- Alfabeto: $\Sigma = \{a, b, c\}$
- Funciones de transición: δ
 - $\delta(0, a) = 1$
 - $\delta(1, a) = 3$
 - $\delta(1, c) = 2$
 - $\delta(1, b) = 2$
 - $\delta(2, c) = 2$
 - $\delta(2, b) = 2$
 - $\delta(2, a) = \{5, 3\}$
 - $\delta(3, b) = 4$
 - $\delta(5, a) = 5$
 - $\delta(5, b) = \{4, 2\}$
 - $\delta(5, c) = 2$
- Estado inicial: $q_0 = 0$
- Estados finales: $F = \{4\}$

Diagrama de transiciones

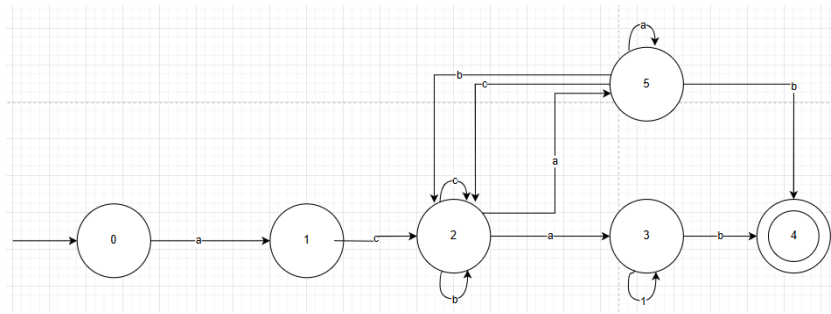


Figure 28: Diagrama de transiciones del AFD propuesto.

Tabla de transiciones

Estado	a	b	c
0	1		
1			2
2	{5,3}	2	2
3	3	4	
4			
5	5	{2,4}	2

Simulación

Práctica 2 José Sánchez - O - x Compiladores: 1.8 Práctica 2 x Aut-matas-y-compiladores-x Automaton Simulator x Untitled Diagram.drawio - d x +

← → ↺ 🌐 automatonssimulator.com ☆ 📄 🌐

Automaton Simulator: [DFA] [NFA] [PDA] [🔍] [Examples ▾]

Test / Debug: [👉] [🔄] [🛑]

Bulk Testing [➡]

Accept (one per line):

```
acab
acaaaab
accocaaacacacccacbab
acbbbaababababacab
accccbababababab
```

Reject (one per line):

```
ac
ab
acaaa
abbab
abbabac
```

Test Results:

```
Accept acab -- Pass
Accept acaaaab -- Pass
Accept accocaaacacacccacbab -- Pass
Accept acbbbaababababacab -- Pass
Accept acccbabababababab -- Pass
Reject ac -- Pass
Reject ab -- Pass
Reject acaaa -- Pass
Reject abbab -- Pass
Reject abbabac -- Pass
```

Kyle Dickerson - kyle.dickerson@gmail.com 📄 - On GitHub Tools: Query inPumb Icons: Favicon

09:54 a.m. 24/02/2025

Figure 29: Simulación.

Palabras aceptadas y rechazadas

Test Results:

```
Accept: acab -- Pass
Accept: acaaab -- Pass
Accept: accccaacacaacccacbab -- Pass
Accept: acbbbbbbaabababacab -- Pass
Accept: accccbbbbbbbab -- Pass
Reject: ac -- Pass
Reject: ab -- Pass
Reject: acaaa -- Pass
Reject: abbab -- Pass
Reject: abbbac -- Pass
```

Figure 30: Palabras rechazadas y aceptadas.

Conclusiones

En conclusión, la herramienta "Automaton Simulator" es bastante útil para poder representar autómatas de tipo determinista, no determinista y con transiciones en lambda. Además, facilita mucho la visualización del autómata de manera general y en algunos casos, particular. En adición, es importante que se tenga un buen pensamiento lógico para poder llevar a cabo las representaciones debido a que se puede entrar en bucles de los que no se puede salir, puede que el autómata solo funcione con ciertas cadenas (en algunos casos no nos daremos cuenta de esas cadenas a no ser que hagamos una evaluación general del sistema).

Por último, se pudo observar que los AFND pueden ser más eficientes en contraposición a los AFD, esto debido a que se permiten dos funciones de transición para un mismo símbolo de un estado, sin embargo, su creación puede ser más compleja si se está acostumbrado a hacer solamente autómatas finitos deterministas.

Referencias Bibliográficas

References

- [1] Kelley, D. (1995). *Teoría de autómatas y lenguajes formales*. Prentice-Hall, Madrid. ISBN 0134977777. Clasificación LC QA76 .9 .M35 K4. Traducción de Ma. Luisa Diez Platas.
- [2] Hopcroft, J. E. (2002). *Introducción a la teoría de autómatas, lenguajes y computación*. 2a ed. Pearson Education, México.
- [3] Cases Muñoz, R. (2002). *Lenguajes, gramáticas y autómatas: curso básico*. Pearson Education, México.
- [4] Codemath. (2024). *Autómatas y lenguajes formales desde cero* [Lista de reproducción]. YouTube. Recuperado de <https://www.youtube.com/>
- [5] Codemath. (2024). *Convertir un Automata NO Determinista (AFND) a Determinista (AFD)* [Video]. YouTube. Recuperado de https://www.youtube.com/watch?v=_UdVL-84rXc
- [6] Codemath. (2024). *Qué es un Autómata Finito No Determinista (AFND)* [Video]. YouTube. Recuperado de <https://www.youtube.com/watch?v=>
- [7] Codemath. (2023). *Qué es un Autómata Finito Determinista (AFD)* [Video]. YouTube. Recuperado de <https://www.youtube.com/watch?v=>
- [8] Codemath. (2023). *Operaciones con Lenguajes y Aplicaciones — Lenguajes Formales III* [Video]. YouTube. Recuperado de <https://www.youtube.com/watch?v=>