

# Evaluación del desempeño de un Autómata Finito Determinista Bidireccional con memoria Lifo/Fifo

Juan Giró<sup>1,2</sup>, Juan Vázquez<sup>2</sup>, Brenda Meloni<sup>2</sup> y Leticia Constable<sup>2</sup>

<sup>1</sup> *Departamento de Estructuras, Facultad de Ciencias Exactas, Físicas y Naturales, Universidad Nacional de Córdoba, Ciudad de Córdoba, Argentina*

<sup>2</sup> *Departamento de Ingeniería en Sistemas de Información, Facultad Regional Córdoba, Universidad Tecnológica Nacional, Ciudad de Córdoba, Argentina*

Fecha de recepción del manuscrito: 22/03/2016

Fecha de aceptación del manuscrito: 16/08/2016

Fecha de publicación: 15/03/2017

**Resumen**— En el campo de las máquinas abstractas hay una franja muy interesante que normalmente recibe poca atención, que está integrada por aquellas de capacidad inferior a la Máquina de Turing y mayor a la del Autómata Finito. Las evidencias mostraron que estas máquinas disponen de una gran potencialidad y pueden tener desempeños muy interesantes ante problemas específicos, lo que llevó a tratarlas como objetos de estudio en este trabajo. Con este fin se reconocieron y evaluaron las principales máquinas disponibles, se propuso una nueva máquina con memoria Lifo/Fifo, se seleccionó un caso de estudio y se analizaron los resultados obtenidos mediante el uso de un simulador implementado a tal fin. Las pruebas se orientaron a evaluar la complejidad temporal y la sensibilidad de este indicador ante variantes en las cadenas de datos, contrastando los resultados con los obtenidos con dos Máquinas de Turing. El trabajo ofreció la oportunidad de reconocer otras máquinas a ser estudiadas en el futuro y también confirmaron el enorme valor técnico y pedagógico de los procesos de simulación.

**Palabras clave**— Máquinas abstractas, complejidad computacional, simulación.

**Abstract**— The field of abstract machines includes a very interesting band that usually receives little attention. It is composed of machines of lower capacity than Turing Machine and greater than Finite Automaton. Evidences showed that these machines have a great potential and have very interesting performance when solving specific problems. So, main machines available were recognized and a new machine with LIFO / FIFO memory was proposed. After that, a case study was selected and its results were analyzed with a specific simulator that had to be implemented in order to study the behavior of the new machine. The tests were oriented to the evaluation of time complexity and to study the sensitivity of this indicator to different data strings. The results were compared with those obtained with Turing machines. This work offered the opportunity to recognize other machines to be studied in the future and also confirmed the enormous technical and educational value of simulation processes.

**Keywords**— Abstract machines, computational complexity, simulation.

## INTRODUCCIÓN

El campo de la *Ciencia de la Computación* se estableció y consolidó a partir de las máquinas abstractas. En efecto, estas máquinas anticiparon conceptualmente muchos de los progresos de la computación práctica que hoy son realidades, y hay otras que esperan hacerlo en algún futuro, acompañando los avances de la tecnología.

Al tratarlas resulta conveniente comenzar por la Máquina de Turing, que representa el límite de lo computable. Luego, solo restringiendo con diferentes recursos el acceso a la

memoria, que está representada como una cinta infinita de acceso secuencial, se obtiene una enorme y variada diversidad de máquinas de capacidades progresivamente inferiores, algunas muy originales y poco conocidas.

Cabría entonces preguntarse el sentido que tienen tales restricciones. La respuesta es simple: con máquinas bien pensadas, de menor capacidad y más específicas, se puede ganar en eficiencia, lo que significa menor complejidad operativa. Es decir, se obtiene una reducción del tiempo de proceso y/o espacio demandado para resolver un mismo problema, lo que en muchos casos hace posible su viabilidad práctica y justifica el esfuerzo que se realiza en este amplio campo de investigación.

En este contexto se inscribe este trabajo, que está organizado de la siguiente manera: se comienza por proponer y describir una máquina abstracta “no convencional”, se selecciona un caso testigo y se muestra el

Nombre y Dirección de autor de contacto:

Juan Francisco Giró, FCEFyN-UNC, Avenida Vélez Sarsfield 1611, Ciudad Universitaria, 5016 Ciudad de Córdoba, Argentina, Tel: 351 5353800 interno 723, juanfgiro@gmail.com.

desempeño de la nueva máquina confrontándola con dos Máquinas de Turing. Por último, se discuten los resultados obtenidos y se presentan las conclusiones.

## DESCRIPCIÓN DEL AUTÓMATA PROPUESTO

En la búsqueda de una máquina interesante y novedosa, que justifique su estudio, se revisaron las diversas opciones propuestas y descriptas en la literatura. En la selección se dio preferencia a autómatas sin capacidad para alterar la información de la cinta de entrada, orientando el trabajo hacia máquinas reconocedoras de lenguajes. Luego, se consideró la conveniencia de disponer de memoria auxiliar con un tipo de acceso específico y control del movimiento del cabezal de entrada en dos sentidos, todo lo cual brinda recursos para poder trascender el reconocimiento de los lenguajes regulares o de tipo 3.

En resumen, la máquina buscada se obtiene tomando como base un Autómata Finito Determinista Bidireccional (AFDB), al que se lo dota de una memoria dual de tipo Lifo/Fifo. La primera condición tiene numerosos antecedentes, al igual que las máquinas con memorias Lifo o Fifo, no siendo habitual la coexistencia de ambos tipos de accesos a la memoria. A partir de sus componentes se adopta la designación *Autómata Finito Determinista Bidireccional Lifo/Fifo (AFDB-LF)* para el nuevo autómata.

Como resultado de la búsqueda de antecedentes se citan a continuación algunos de los principales y más reconocidos: el AFDB fue estudiado por Rabin y Scott [1] y Sheperdson [2], el autómata con pila o de memoria Lifo por Oettinger [3] y Schutzenberger [4], finalmente el autómata con cola o de memoria Fifo fue estudiado por Cherubini et Al [5]. Caben aquí agregar los aportes de Rosenberg [7] y de Petersen [6], que demostraron que un autómata con memoria Fifo podía emular una Máquina de Turing, al igual que lo puede hacer el autómata con dos pilas de Koslowski [8]. También deben citarse los trabajos de Aho, Hopcroft y Ullman [9], Hopcroft y Ullman [10], Alonso [12] y Gluck [11] referidos al autómata con pila bidireccional y los de Bhattacharjee [13] y Asha latha [14] que estudiaron un autómata con memoria dual Lifo/Fifo. Este último es denominado en la literatura *Deque Automata*.

Las características del *AFDB-LF* se definieron a partir del análisis exhaustivo de estas referencias y de otras similares, adoptándose para su descripción la forma clásica de definir este tipo de máquinas.

El autómata propuesto queda establecido como:

$$AFDB-LF = (\Sigma_E, \Sigma_C, \Gamma, Q, q_0, \#, A, f)$$

donde:

$\Sigma_E$  : Alfabeto de entrada

$\Sigma_C$  : Alfabeto de cinta,  $\Sigma_C = \Sigma_E \cup \{ >, < \}$

$\Gamma$  : Alfabeto de la pila/cola

$Q$  : Conjunto de estados

$q_0$  : Estado inicial,  $q_0 \in Q$

$\#$  : Referencia de pila / cola vacía,  $\# \in \Gamma$

$A$  : Estados de aceptación,  $A \subset Q$

$f$  : Función de transición:  $f(\Omega \times \Sigma_C \times \Gamma) \rightarrow Q \times \Gamma^* \times m$

Siendo:

$\Omega$  : Conjunto de estados con identificadores del tipo de

acceso que es operado por cada uno. Lifo (pila) o Fifo (cola):  $\Omega = \{ \omega / \omega = \mu q, \mu \in \{ +, - \}, q \in Q \}$ , donde “+” = acceso Lifo y “-” = acceso Fifo .

$m$  : Sentidos del próximo movimiento del cabezal sobre la cinta de entrada,  $m = \{ I, N, P, D \} = \{ \text{Izquierda, Neutro, Parada y Derecha} \}$ .

Se representa a la descripción instantánea o configuración del autómata por la cuaterna  $(q, \langle a \rangle, k, \delta)$  que define: *i)* el estado actual, *ii)* el contenido de la cinta, *iii)* la posición del cabezal y *iv)* el contenido de la pila. A partir de esta definición se expresan los posibles movimientos del autómata, que son los siguientes:

a) **acceso LIFO**,  $f(+q, a, c) = (s, ca, D)$

$$(+q, \rangle a \beta \langle, 1, \#bc) \vdash (s, \rangle a \beta \langle, 2, \#bca)$$

El símbolo de tope de pila “c” es leído y repuesto. A continuación se apila el símbolo “a” leído de la cinta de entrada.

b) **acceso FIFO**,  $f(-q, a, b) = (s, ba, D)$

$$(-q, \rangle a \beta \langle, 1, \#bc) \vdash (s, \rangle a \beta \langle, 2, \#cba)$$

El símbolo de fondo de pila “b” es leído y apilado. Luego, a continuación se apila el símbolo “a” leído de la cinta de entrada.

En los citados ejemplos los significados de los símbolos empleados son:  $D \in m$ ;  $c, ca, b, ba, \#bc, \#bca, \#cba \in \Gamma^*$ ;  $+q, -q \in \Omega$ ;  $a \in \Sigma_E$ ;  $\beta \in \Sigma_E^*$  y  $s \in Q$ .

Aquí debe observarse que la condición de acceso a la memoria está asociado a cada estado, es decir que hay estados desde los cuales el acceso es Lifo y desde otros es Fifo, lo que distingue este autómata de los presentados en los trabajos de Bhattacharjee [13] y Asha latha [14] ya citados. Merece destacarse que en caso de admitirse ambos accesos en un mismo estado, quedaría planteada una nueva forma de no determinismo, que no es aquí objeto de estudio.

Nótese que los dos tipos de acceso disponibles permiten leer desde el tope (Lifo) o desde el fondo (Fifo). En este último caso el símbolo leído es el penúltimo, salvo que la cola esté vacía, en cuyo caso se lee el último (#). Es decir, al estar vacía la pila o cola la máquina recibe como entrada la marca específica que identifica esta condición (#). Por su parte las grabaciones ( $\Gamma^*$ ) tienen siempre como destino el tope de la pila, pudiendo tratarse de una cadena vacía ( $\lambda$ ).

Como consecuencia de lo expuesto, es importante observar que, con un solo acceso Fifo y salvo que esté vacía, la memoria de cola nunca puede restablecerse a su condición anterior a un movimiento, ya que se descarga por abajo y carga por arriba. Por el contrario, esto si es posible con un acceso Lifo, que descarga y carga por el mismo extremo. Las ventajas y limitaciones de esta diferencia es uno de los temas a comprobar a través del simulador.

Como ocurre con los autómatas con pila convencionales, para poder representar la función de transición con una tabla se adopta el criterio de que las columnas corresponden a los elementos de  $\Sigma_C$  y las filas a pares  $\Omega \times \Gamma$  (estados con sus formas de acceso y símbolos leídos de la memoria Lifo/Fifo). Esta idea de mostrar funciones tridimensionales en tablas combinando dos dimensiones es atribuida a

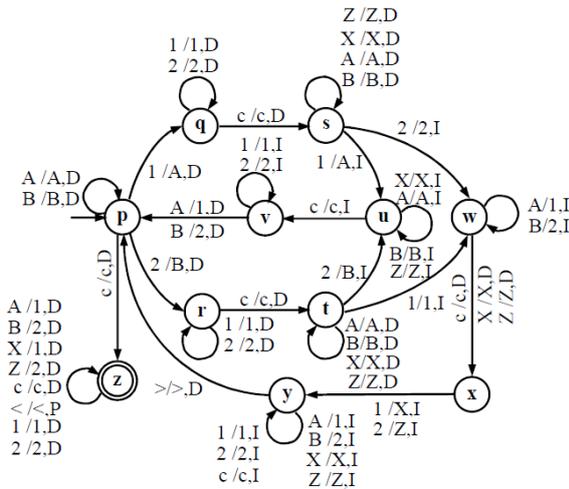
Alfonseca et Al. [15] y posibilita una representación muy compacta y conveniente de este tipo de autómatas. Asimismo, como ocurre con todas las máquinas abstractas, los AFDB-LF también pueden ser representados con grafos.

**CASO DE ESTUDIO**

El objetivo planteado es comparar las complejidades temporales obtenidas a partir de la solución de un caso de estudio con dos máquinas: el Autómata Finito Determinista Bidireccional Lifo/Fifo (AFDB-LF) antes descrito y una Máquina de Turing Determinista (MTD). Para ello se utilizaron dos simuladores, ambos desarrollados especialmente para este estudio y acordes a las características de las máquinas involucradas.

Como caso de estudio se optó por un lenguaje que no es independiente de contexto, que fue propuesto por Rytter [16] (caso 49), tal que:  $L = \{\alpha / \alpha = \delta c \beta \delta \gamma, \delta \in \{1,2\}^+, \beta, \gamma \in \{1,2\}^* \text{ y } \Sigma_E = \{1,2,c\}\}$ . Es decir que cada sentencia del lenguaje contiene un prefijo  $\delta$  seguido de un separador “c”, luego del cual se repite la cadena  $\delta$  en un contexto  $\beta \delta \gamma$ .

Tal como fue anticipado, el problema es resuelto mediante la MTD, de la cual se presentan dos variantes, y a las que se suma el AFDB-LF presentado en este trabajo. Los grafos de los tres autómatas son representados a continuación.

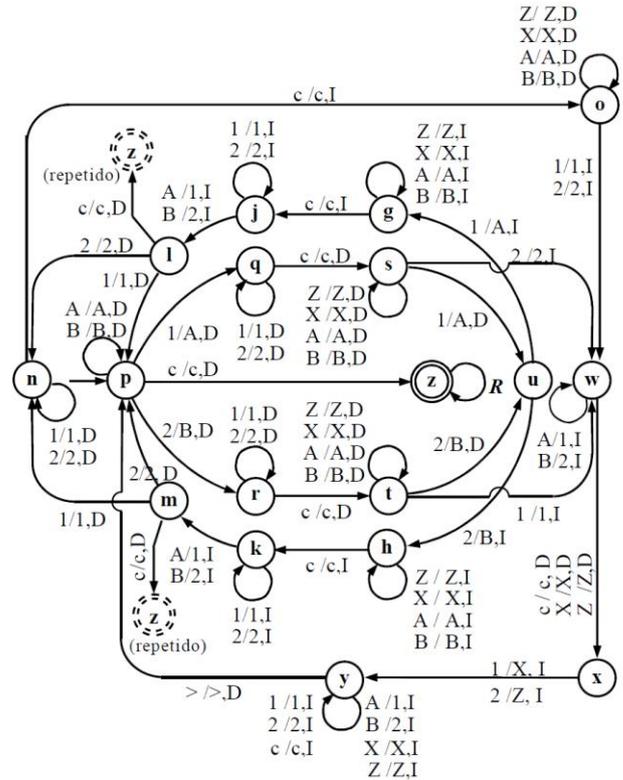


**Fig.1.** Grafo de la Máquina de Turing “básica” (MTD-B)

En la Figura 1 se representa el grafo de la primer MTD, denominada “básica” e identificada en lo sucesivo por MTD-B. Como puede comprobarse, por cada símbolo del prefijo que es leído se busca encontrar en el sufijo el mismo símbolo en la secuencia correcta. En caso negativo la máquina vuelve su cabezal a la izquierda al inicio de la cadena y repite el mismo ciclo anterior hasta encontrar la secuencia buscada o alcanzar el fin de la cadena. Para ello debe desplazarse progresivamente hacia la derecha, sobre el sufijo, el punto a partir del cual comienza la búsqueda de ésta secuencia.

Nótese que la MTD-B controla un solo símbolo del prefijo por cada avance y retroceso del cabezal de lectura. Buscando una reducción del tiempo demandado para la verificación de las cadenas, se propone controlar dos símbolos, uno a la ida y otro a la vuelta, lo que da lugar a una optimización de la máquina anterior, que en lo sucesivo es denominada MTD-O.

Esta mejora lleva a un incremento significativo en la cantidad de estados de la máquina, que con una lógica similar pasa de once a dieciocho, tal como se muestra a continuación en el Grafo de la Figura 2.

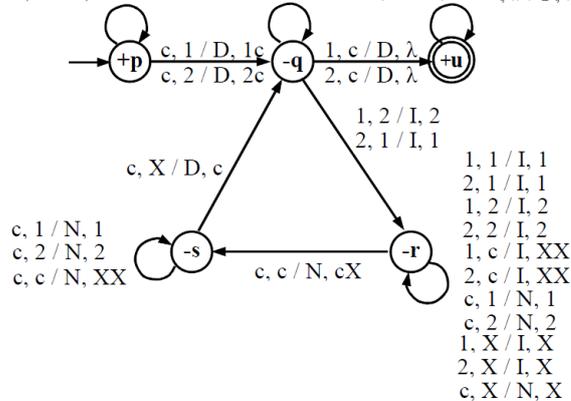


$$R = \{A/1,D, B/2,D, X/1,D, Y/2,D, c/c,D, 1/1,D, 2/2,D, </<,P\}$$

**Fig.2.** Grafo de la Máquina de Turing “optimizada” (MTD-O)

Finalmente, en la Figura 3 se representa el grafo del AFDB-LF que es la nueva máquina propuesta en este trabajo.

- 1, # / D, #1
- 2, # / D, #2
- 1, 1 / D, 11
- 1, 2 / D, 21
- 2, 1 / D, 12
- 2, 2 / D, 22
- 2, c / D, λ
- 1, c / D, λ
- 1, X / D, λ
- 1, 1 / D, λ
- 1, 2 / D, λ
- 2, 1 / D, λ
- 2, 2 / D, λ
- 2, 2 / D, λ
- 1, # / D, #
- 2, # / D, #
- <, 1 / N, λ
- <, 2 / N, λ
- <, X / N, λ
- <, # / P, #



**Fig.3.** Grafo del Autómata Finito Bidireccional Lifo/Fifo (AFDB-LF)

Como puede comprobarse, resuelve el problema con solo cinco estados, donde los denominados “p” y “u” tienen acceso Lifo a la memoria auxiliar, mientras que el de los estados “q”, “r” y “s” es Fifo. Esto le permite cargar una sola vez el prefijo y hacerlo rotar en la memoria (Fifo) a

medida que recorre el sufijo. Cada vez que falla, vuelve al separador y sincroniza el contenido de la memoria de manera de saltar un símbolo adicional, idea equivalente a la utilizada por las MTD.

**EVALUACIÓN DE LAS MÁQUINAS**

La complejidad temporal es un indicador indirecto, cuyo valor depende de tres factores: *i*) la complejidad intrínseca del propio problema, *ii*) las características del procedimiento propuesto para resolverlo (en este caso las máquinas adoptadas y sus implementaciones) y *iii*) las particularidades de los propios datos.

En cuanto a la dependencia de los datos, en el caso del problema estudiado, se puede presumir que la complejidad temporal no solo depende del largo de la cadena  $n = |\alpha|$ , sino que también influyen: *i*) la posición relativa de  $\delta$  en el contexto  $\beta\delta\gamma$  y *ii*) la composición de las cadenas  $\beta$ ,  $\delta$ , y  $\gamma$ . Por este motivo se adoptó como criterio común que el prefijo buscado  $\delta$  está en el centro del sufijo  $\beta\delta\gamma$ , lo que equivale a decir que  $|\beta| = |\gamma|$  y además se les asignó un largo tres ( $|\beta| = |\gamma| = 3$ ). Es decir,  $n = 2|\delta| + 7$ .

Finalmente, para tener una apreciación de la importancia de la composición de las cadenas y facilitar la interpretación de los resultados se seleccionaron los siguientes tres casos simples:

Caso A:  $\alpha_1 = \underline{12121}c222\underline{12121}1111$

Caso B:  $\alpha_2 = \underline{21111}c111\underline{21111}222$

Caso C:  $\alpha_3 = \underbrace{12222}_\delta c \underbrace{111}_\beta \underbrace{12222}_\delta \underbrace{222}_\gamma$

Los tres casos fueron resueltos con la Máquina de Turing Básica (MTD-B, Figura 1), Máquina de Turing Optimizada (MTD-O, Figura 2) y Autómata Finito Determinista Bidireccional Lifo/Fifo (AFDB-LF, Figura 3). Luego, para obtener las expresiones de complejidad se planteó la hipótesis de que se trata de polinomios que son de 3<sup>er</sup> grado como máximo, lo que fue confirmado haciendo distintas determinaciones con diferentes largos de cadenas.

Finalmente, para obtener los datos necesarios sobre el desempeño de los tres autómatas en los tres casos se utilizaron los simuladores ya citados y cadenas de largo  $n = 13, 17, 21$  y  $25$ , que corresponden a  $|\delta| = 3, 5, 7$  y  $9$ .

Una vez conocidos los correspondientes valores de  $T(n)$ , se plantearon los sistemas de ecuaciones y obtuvieron los coeficientes de los polinomios:

$$An^3 + Bn^2 + Cn + D = T(n) \tag{1}$$

$$\begin{bmatrix} 13^3 & 13^2 & 13 & 1 \\ 17^3 & 17^2 & 17 & 1 \\ 21^3 & 21^2 & 21 & 1 \\ 25^3 & 25^2 & 25 & 1 \end{bmatrix} \begin{bmatrix} A \\ B \\ C \\ D \end{bmatrix} = \begin{bmatrix} T(13) \\ T(17) \\ T(21) \\ T(25) \end{bmatrix} \tag{2}$$

En todos los casos estudiados fue nulo el coeficiente del término de mayor grado, lo que confirmó que la complejidad queda representada por polinomios de 2° grado como máximo.

En la Tabla 1 se muestran los resultados correspondientes al caso “A”, donde los valores de complejidad para  $n = 13, 17, 21$  y  $25$  fueron determinados con los simuladores y con los polinomios obtenidos fueron calculados los de  $n = 29$  y  $33$ .

**TABLA 1.** VALORES OBTENIDOS EN EL CASO “A” CON LOS SIMULADORES PARA LAS TRES MÁQUINAS Y POLINOMIOS T(n)

Caso A: Análisis de complejidad temporal			
Cadena: $\alpha_1 = \underline{12121}c222\underline{12121}1111$			
n	MTD-B	MTD-O	AFDB-LF
13	97	135	47
17	161	191	65
21	241	255	83
25	337	327	101
29	449	407	119
33	577	495	137
Máquina	Expresiones de T(n)		O
MTD-B	$T(n) = 0,50n^2 + n - 0,5$		$n^2$
MTD-O	$T(n) = 0,25n^2 + 6,5n + 8,25$		$n^2$
AFDB-LF	$T(n) = 4,50n - 11,5$		$n$

**TABLA 2.** VALORES OBTENIDOS EN EL CASO “B” CON LOS SIMULADORES PARA LAS TRES MÁQUINAS Y POLINOMIOS T(n)

Caso B: Análisis de complejidad temporal			
Cadena: $\alpha_2 = \underline{21111}c111\underline{21111}222$			
n	MTD-B	MTD-O	AFDB-LF
13	97	87	47
17	161	131	65
21	241	183	83
25	337	243	101
29	449	311	119
33	577	387	137
Máquina	Expresiones de T(n)		O
MTD-B	$T(n) = 0,50n^2 + n - 0,5$		$n^2$
MTD-O	$T(n) = 0,25n^2 + 3,5n - 0,75$		$n^2$
AFDB-LF	$T(n) = 4,50n - 11,5$		$n$

**TABLA 3.** VALORES OBTENIDOS EN EL CASO “C” CON LOS SIMULADORES PARA LAS TRES MÁQUINAS Y POLINOMIOS T(n)

Caso C: Análisis de complejidad temporal			
Cadena: $\alpha_3 = \underline{12222}c111\underline{12222}222$			
n	MTD-B	MTD-O	AFDB-LF
13	133	135	47
17	209	191	65
21	301	255	83
25	409	327	101
29	533	407	119
33	676	495	137
Máquina	Expresiones de T(n)		O
MTD-B	$T(n) = 0,50n^2 + 4n - 0,5$		$n^2$
MTD-O	$T(n) = 0,25n^2 + 6,5n + 8,25$		$n^2$
AFDB-LF	$T(n) = 4,50n - 11,5$		$n$

que están representados en la parte inferior de las tablas. De la misma forma se muestran en la tabla 2 los resultados y polinomios del Caso “B” y en la Tabla 3 los del Caso “C”.

Los valores presentados en las tablas anteriores son representados en los gráficos de la Figura 4 (Caso “A”), Figura 5 (Caso “B”) y Figura 6 (Caso “C”), donde las ordenadas corresponden a la complejidad temporal  $T(n)$  y las abscisas a los largos de cadenas. En los tres casos los rombos (verde) corresponden a la MTD-B, los cuadrados (azul) a la MTD-O y los triángulos (rojo) al AFDB-LF.

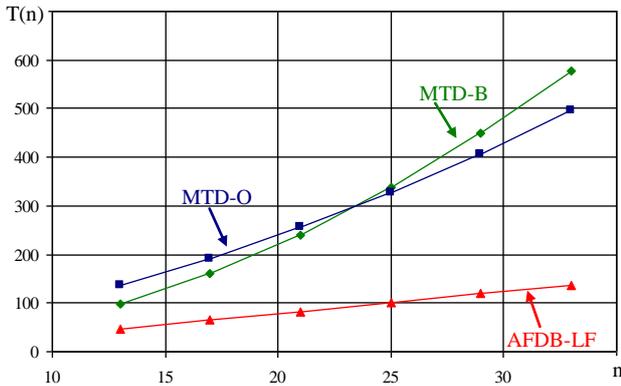


Fig. 4. Caso A: Evolución de la complejidad temporal de las tres máquinas

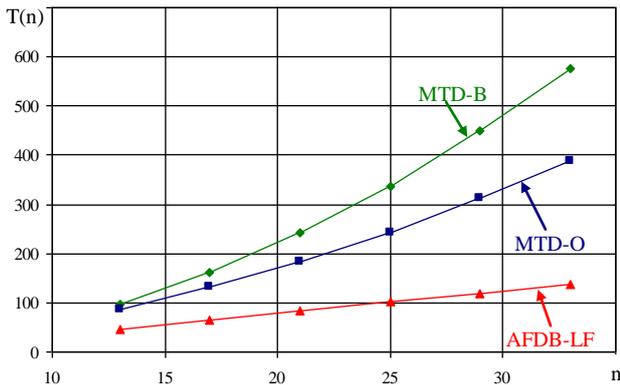


Fig. 5. Caso B: Evolución de la complejidad temporal de las tres máquinas

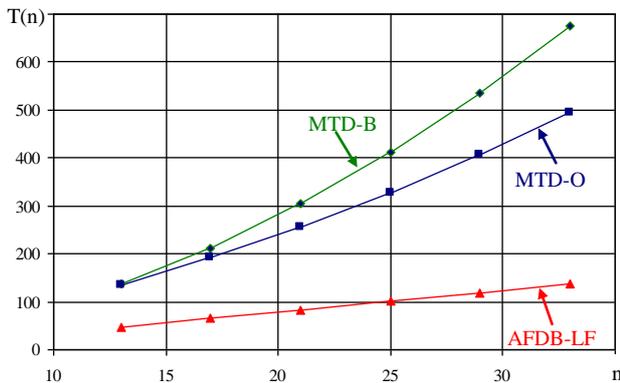


Fig. 6. Caso C: Evolución de la complejidad temporal de las tres máquinas

Una vez obtenidos los valores y expresiones de complejidad de las tres cadenas reconocidas por las tres máquinas, se determinaron las reducciones porcentuales que brindaron la Máquina de Turing optimizada y el Autómata Finito Determinista Bidireccional, que son presentados en la Tabla 4, junto a las expresiones utilizadas para su cálculo. Estos valores representan la reducción de la complejidad temporal

obtenida con: *i*) la optimización de la Máquina de Turing y *ii*) el AFDB-LF propuesto. En todos los casos se toma como referencia la complejidad de la máquina básica (MTD-B).

Tabla 4. DIFERENCIAS PORCENTUALES ENTRE LA MÁQUINA DE TURING BÁSICA, LA MÁQUINA DE TURING OPTIMIZADA Y EL AFDB-LF

Diferencias entre MTD-B, MTD-O y ADFB-LF					
n	Dif <sub>1</sub>	Dif <sub>2</sub>	Dif <sub>3</sub>	Dif <sub>4</sub>	Dif <sub>5</sub>
13	-39,18	51,55	10,31	0,74	65,44
17	-18,63	59,63	18,63	9,91	69,34
21	-5,81	65,56	24,07	16,12	72,70
25	2,97	70,03	27,89	20,63	75,49
29	9,35	73,50	30,73	24,07	77,80
33	14,21	76,26	32,93	26,78	79,73

$$Dif_1 = (T(n)_{MTD-B} - T(n)_{MTD-O}) / T(n)_{MTD-B}; \alpha_1$$

$$Dif_2 = (T(n)_{MTD-B} - T(n)_{AFDB-LF}) / T(n)_{MTD-B}; \alpha_1, \alpha_2$$

$$Dif_3 = (T(n)_{MTD-B} - T(n)_{MTD-O}) / T(n)_{MTD-B}; \alpha_2$$

$$Dif_4 = (T(n)_{MTD-B} - T(n)_{MTD-O}) / T(n)_{MTD-B}; \alpha_3$$

$$Dif_5 = (T(n)_{MTD-B} - T(n)_{AFDB-LF}) / T(n)_{MTD-B}; \alpha_3$$

Luego, en la Figura 7 se presenta la evolución de la reducción porcentual de complejidad temporal en cada caso a medida que aumentan los largos de las cadenas.

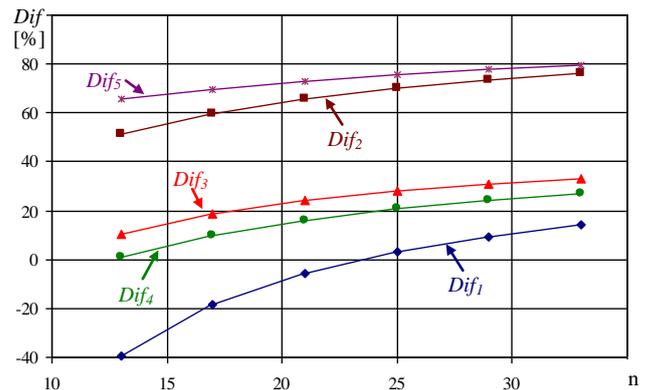


Fig. 7. Evolución de la reducción porcentual de la complejidad temporal al aumentar el largo de las cadenas en los tres casos estudiados.

Sin embargo y como puede apreciarse, la tendencia asintótica de la reducción porcentual de la complejidad al aumentar el largo “n” de las cadenas no es posible de acotar sobre intervalos de largos de cadena tan chicos.

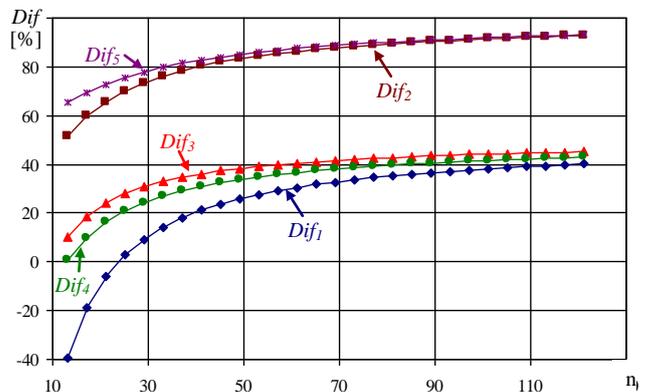


Fig. 8. Evolución de la reducción porcentual de la complejidad temporal al en los tres casos estudiados para cadenas de largos 13 a 121.

Por este motivo y con la finalidad de conocer mejor las

tendencias de las curvas se amplió el intervalo estudiado con el auxilio de los polinomios. En la Figura 8 se muestran los resultados obtenidos cuando se cubre el rango  $n = 13$  a 121, donde se aprecia con más claridad la tendencia hacia una reducción del 50% con la MTD-O y una reducción del 100% con el AFDB-LF.

Finalmente, para confirmar la cota superior de la diferencia se toman límites sobre las expresiones de las diferencias porcentuales cuando el largo de las cadenas  $n$  tiende a  $\infty$ . Se obtienen entonces:

$$\begin{aligned} Dif_1 &= \lim_{n \rightarrow \infty} 100 \frac{(0,5n^2 + n - 0,5) - (0,25n^2 + 6,5n + 8,25)}{0,5n^2 + n - 0,5} \\ &= \lim_{n \rightarrow \infty} 100 \frac{0,25 - \frac{5,5}{n} - \frac{7,75}{n^2}}{0,5 + \frac{1}{n} - \frac{0,5}{n^2}} = 50,0\% \end{aligned} \quad (3)$$

$$\begin{aligned} Dif_2 &= \lim_{n \rightarrow \infty} 100 \frac{(0,5n^2 + n - 0,5) - (4,5n - 11,5)}{0,5n^2 + n - 0,5} = \\ &= \lim_{n \rightarrow \infty} 100 \frac{0,5 - \frac{3,50}{n} + \frac{11,0}{n^2}}{0,5 + \frac{1}{n} - \frac{0,5}{n^2}} = 100,0\% \end{aligned} \quad (4)$$

$$\begin{aligned} Dif_3 &= \lim_{n \rightarrow \infty} 100 \frac{(0,5n^2 + n - 0,5) - (0,25n^2 + 3,5n - 0,75)}{0,5n^2 + n - 0,5} \\ &= \lim_{n \rightarrow \infty} 100 \frac{0,25 - \frac{2,5}{n} + \frac{0,25}{n^2}}{0,5 + \frac{1}{n} - \frac{0,5}{n^2}} = 50,0\% \end{aligned} \quad (5)$$

$$\begin{aligned} Dif_4 &= \lim_{n \rightarrow \infty} 100 \frac{(0,5n^2 + 4n - 0,5) - (0,25n^2 + 6,50n + 8,25)}{0,5n^2 + 4n - 0,5} \\ &= \lim_{n \rightarrow \infty} 100 \frac{0,25 - \frac{2,50}{n} - \frac{8,75}{n^2}}{0,5 + \frac{4}{n} - \frac{0,5}{n^2}} = 50,0\% \end{aligned} \quad (6)$$

$$\begin{aligned} Dif_5 &= \lim_{n \rightarrow \infty} 100 \frac{(0,5n^2 + 4n - 0,5) - (4,5n - 11,5)}{0,5n^2 + 4n - 0,5} = \\ &= \lim_{n \rightarrow \infty} 100 \frac{0,5 - \frac{0,50}{n} + \frac{11,0}{n^2}}{0,5 + \frac{4,0}{n} - \frac{0,50}{n^2}} = 100,0\% \end{aligned} \quad (7)$$

Esto confirma las reducciones de complejidad temporal del 50% que brinda la MTD-O y la reducción del 100% del AFDB-LF ya anticipadas con los gráficos de la Figura 8.

## DISCUSIÓN DE RESULTADOS

De una inspección detallada de los resultados obtenidos surgen los siguientes comentarios:

- El AFDB-LF fue más eficiente que las MTD en todos los casos estudiados. En efecto, las expresiones de complejidad de las primeras son de 2° grado y las de la última son siempre lineales.

- Con la MTD-B se obtienen las mismas expresiones de complejidad temporal para las cadenas  $\alpha_1$  y  $\alpha_2$ , pero una diferente para  $\alpha_3$ .
- Por el contrario, con la MTD-O las que son iguales son las expresiones de  $\alpha_1$  y  $\alpha_3$ , siendo diferente la de  $\alpha_2$ .
- Estas igualdades y diferencias entre las expresiones de complejidad obtenidas con las MTD no guardan en apariencia una relación con la morfología de las cadenas procesadas. Es decir, hasta donde se pudo comprobar no parecen ser fácilmente anticipables.
- Las mismas tres cadenas dan lugar a idénticos polinomios de complejidad temporal al ser tratadas con el AFDB-LF, lo que también fue comprobado con otras diversas cadenas de variada morfología.
- En cadenas cortas la MTD-O puede ser menos eficiente, como es el caso de la cadena  $\alpha_1$  donde se alcanza una ventaja recién a partir de un largo de cadena  $n > 25$ . Por el contrario, para todos los largos de cadenas  $\alpha_2$  y  $\alpha_3$  la MTD-O redujo la complejidad temporal respecto de la MTD-B,
- Como pudo apreciarse, la reducción porcentual de la complejidad tiene una tendencia asintótica al aumentar el largo de las cadenas, que es posible acotar considerando cadenas suficientemente largas. Cuando ello ocurre el término de mayor grado es claramente dominante y las diferencias morfológicas pierden significación.

A partir de estas interpretaciones de los resultados se pueden ensayar algunas reflexiones generales, como las siguientes:

- a) Por ser la complejidad temporal un indicador indirecto que representa al problema, al procedimiento de solución y a los datos, sus valores deben ser interpretados con cuidado ya que con facilidad pueden llevar a apreciaciones erróneas.
- b) Lo anterior justifica que sea muy recomendable explorar y estudiar soluciones alternativas siempre que un problema tenga un costo elevado en términos de tiempo de proceso (complejidad temporal).
- c) En este proceso exploratorio de la solución más conveniente se debe analizar con cuidado el impacto del tercer factor ya anticipado, que son los datos. La dependencia de los datos no siempre es obvia, y en muchos casos las peores condiciones son insospechadas. En estos estudios es siempre necesario apoyarse en evidencias, y dejar en segundo plano las conjeturas provenientes del sentido común.
- d) El Automata Finito Determinista Bidireccional con memoria auxiliar de doble tipo de acceso (Lifo/Fifo) aquí presentado demostró ser una opción muy conveniente para resolver el caso propuesto, permitiendo reducir en un grado la expresión de la complejidad temporal con una disminución del 54% en la cantidad de estados.
- e) La complejidad ciclomática de las máquinas muestra también una clara ventaja a favor del AFDB-LF, lo que impacta muy favorablemente a la hora de su implementación, testing y mantenimiento.
- f) Si bien no fueron tratados aquí, es necesario reconocer la enorme utilidad de los simuladores en este tipo de estudios, convirtiéndose en la herramienta central en el análisis de la complejidad de algoritmos.

## CONCLUSIONES

Este trabajo refleja la actividad que los autores vienen realizando en el estudio del impacto de variantes no convencionales en el desempeño de autómatas finitos con memorias auxiliares de diferente tipo de acceso. En este caso poniendo el foco en la presentación de un nuevo autómata y la comparación de sus resultados con los de una MT. Para ello, el primer paso fue una indagación histórica de antecedentes, ya que un buen conocimiento de las realizaciones anteriores es siempre el mejor punto de partida. Luego se definió el autómata y un caso de estudio. Los resultados obtenidos fueron muy auspiciosos y permitieron comprobar que: *i*) la especialización de máquinas, que dispongan de ciertos recursos mínimos, puede conducir a una marcada mayor eficiencia que las de máquinas generales de mayor capacidad de cálculo y *ii*) la máquina presentada, con memoria auxiliar de acceso Lifo/Fifo, resultó una propuesta exitosa y ofrece grandes posibilidades. En efecto, la combinación de ambos tipos de accesos, relacionados a los estados de la máquina, otorga mucha flexibilidad y un potencial muy prometedor que debe continuarse estudiando, que será motivo de la actividad a realizarse en el futuro inmediato.

## REFERENCIAS

- [1] Rabin M., Scott D. (1959), "Finite automata and their decision problems", *IBM J. Res. Dev.*, 3(2), pp. 114-125.
- [2] Sheperdson J. (1959), "The reduction of two-way autómata to one-way automata", *IBM J. Res.*, 3(2), pp. 198-200.
- [3] Oettinger A. (1961), "Automatic syntactic analysis and the pushdown store", *Proc. of Symposia on Applied Mathematics*, 12, American Mathem. Soc., Rhode Island.
- [4] Schutzenberger M. (1963), "On Context-Free Languages and Push-Down Automata", *Information and Control*, 6, pp. 246-264.
- [5] Cherubini A., Citrini C., Crespi-Reghizzi S., Mandrioli D. (1991), "Qn fifo automata, breath-first grammars and their relations", *Theoret. Comput. Science*, 85(1), pp. 171-203.
- [6] Petersen H., Robson J. (2006); "Efficient simulations by queue machines", *SIAM Journal on Computing*, 35 (5), pp. 1059-1069.
- [7] Rosenberg B. (1993); "Simulating a stack by queues", *Proceedings of the XIX Latinamerican Conference on Computer Science 1*, pp. 3-13.
- [8] Koslowski J. (2013); "Deterministic single-state 2PDAs are Turing complete", Instituto de ciencias de la computación, TU Braunschweig, Alemania.
- [9] Aho A., Hopcroft J. y Ullman J. (1968), "Time and tape complexity of push-down automaton languages", *Information and Control*, 13:3, pp 186-206.
- [10] Hopcroft J. y Ullman J. (1979), *Introduction to Automata Theory, Lenguajes and Computation*, Cap. 14, Addison Wesley.
- [11] Gluck R. (2013), "Simulation of Two-Way Pushdown Automata Revisited", Dept. of Computer Science, University of Copenhagen, pp. 250-258
- [12] Alonso M., Díaz V., Vilares M. (2003), "Bidirectional Push Down Automata", *CIAA'02 Proceedings of the 7th int. conference on Implementation and application of automata*, pp 35-46.
- [13] Bhattacharjee, A., Uddin R., Debnath, B. DQA (2011), "Automata with new memory, properties and applications", *Int. Journal of Electrical, Electronics and Computer Systems (IJEECS)*.
- [14] Asha latha B., Vishnupriya T., Himabindu N. (2012), "Deque Automata for all classes of Formal languages", *Inter. Journal of Computational Engineering Research*.
- [15] Alfonseca E., Alfonseca M., Morrión R. (2007), *Teoría de Autómatas y Lenguajes Formales*, Editorial McGraw-Hill, 2007.
- [16] Rytter, W. (1987), "100 Exercises in the Theory of Automata and Formal Languages", *Research Report 99*, Ins. of Informatics, Warsaw, 1987.