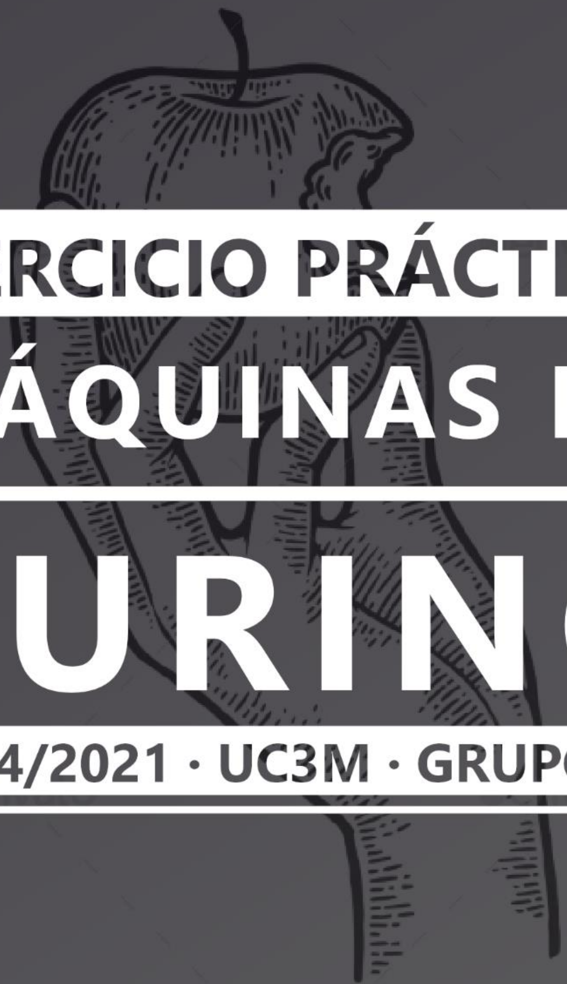


# EJERCICIO PRÁCTICO MÁQUINAS DE TURING

27/04/2021 · UC3M · GRUPO 50


Hugo Romero
100 386 322

Irene Fernández
100 386 294

Maribel Medrano
100 386 149

# CONTENIDO DEL DOCUMENTO

# 0. PALÍNDROMOS

## 0.0. MT Determinista de 1 cinta

### Implementación

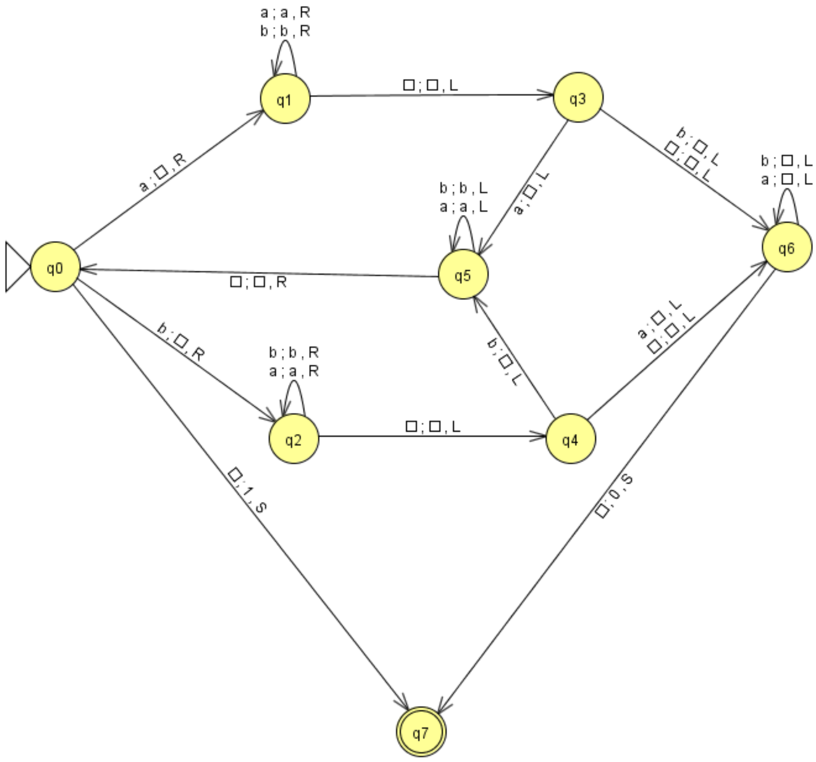



*Ilustración 1*

### Funcionamiento y peor caso

El funcionamiento de la máquina sigue la siguiente lógica:

1. Con el cabezal en la letra más a la izquierda de la palabra verifica sobre qué letra está colocado y la borra.

2. Se posiciona a la derecha del todo de la palabra tratando de buscar una correspondencia con la letra del paso 1.

3.1. Si encuentra la correspondencia la borra y vuelve a empezar el algoritmo.

3.2. Si no encuentra la correspondencia borra toda la palabra y coloca un 0 en la cinta y termina.

4. Si termina de borrar la palabra sin detectar anomalías coloca un 1 en la cinta y termina.

Dado que si se da el paso 3.2. se borra toda la palabra sin analizar la parte restante, este es un caso en el que se reduce el coste. Por tanto, el peor caso es aquel que no reduce pasos. En este caso los tamaños impares no son considerados palíndromos nunca por la máquina, y arrojan valor 0, por lo que el análisis se hará a partir de tamaños par de palabra.

## Simulación y diferencias

A continuación, se hace un análisis de la complejidad del algoritmo en función del tamaño de la palabra introducida, siempre recurriendo que la palabra pertenezca al "peor caso" para el tamaño dado.

| **Entrada** | $\lambda$ | aa | abba | abbbba | ababbaba | ababaababa | aabbaaaabbaa |
|---|---|---|---|---|---|---|---|
| **Tamaño** | 0 | 2 | 4 | 6 | 8 | 10 | 12 |
| **Pasos** | 1 | 6 | 15 | 28 | 45 | 66 | 91 |

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| **Dif 1** | | 5 | 9 | 13 | 17 | 21 | 25 |
| **Dif 2** | | 4 | 4 | 4 | 4 | 4 | |
| **Dif 3** | | 0 | 0 | 0 | 0 | | |

## Cálculo de T(N)

Tras analizar los resultados de la tabla de diferencias finitas se observa que en la diferencia 2 se obtiene valores constantes, esto es un indicador de que el coste es de orden cuadrático. Se procederá a calcular los coeficientes de esta expresión.

$$T(N) = aN^2 + bN + c$$

$$a * 0^2 + b * 0 + c = 1$$

$$a * 2^2 + b * 2 + c = 6$$

$$a * 4^2 + b * 4 + c = 15$$

Resolviendo el sistema se obtiene que a = 1/2, b = 3/2 y c = 1.

## Cota superior asintótica ($n_0$ = 10)

$$g(N) = kN^2$$

$$kN^2 > \frac{1}{2}N^2 + \frac{3}{2}N + 1$$

$$k > \frac{1}{2} + \frac{3}{2N} + \frac{1}{N^2}$$

$$Para\ N = 10 \ \ \rightarrow k > \frac{1}{2} + \frac{3}{20} + \frac{1}{100}$$

$$k > \frac{66}{100}$$

$$g(N) = 0{,}66 * N^2$$

## 0.1. MT Determinista de 2 cintas

### Implementación

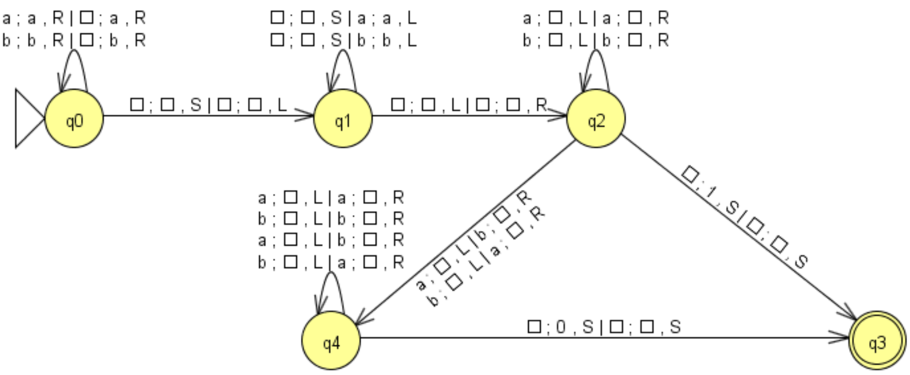



*Ilustración 2*

### Funcionamiento y peor caso

El funcionamiento de la máquina sigue la siguiente lógica:

1. Con el cabezal en la letra más a la izquierda de la palabra recorre toda la cinta 1 copiando la palabra en la cinta 2.

2. Cuando se llegue al final de la cinta 1 se procede a desplazar la cinta 2 hasta el inicio (izquierda)

3. Se recorre ambas cintas, la cinta 1 desde la derecha y la 2 desde la izquierda, comparando las letras de ambas palabras.

4.1. Si encuentra la correspondencia la borra y sigue comparando.

4.2. Si no encuentra la correspondencia borra toda la palabra y coloca un 0 en la cinta y termina.

5. Si termina de borrar la palabra sin detectar anomalías coloca un 1 en la cinta y termina.

Tras analizar los resultados vemos que esta máquina presenta un fallo en la validación de palabras impares, debido a que se tratan como válidas palabras como: aba y aaa, arrojando el valor 1. Por otro lado, tanto las palabras que no son palíndromos como las que lo son tienen el mismo número de pasos hasta el final ya que, aunque exista un fallo se recorrerá la línea entera borrando toda la secuencia. Por esto teniendo en cuenta el fallo antes mencionado tenemos que el peor caso son los palíndromos pares válidos.

### Simulación y diferencias

A continuación, se hace un análisis de la complejidad del algoritmo en función del tamaño de la palabra introducida, siempre recurriendo que la palabra pertenezca al "peor caso" para el tamaño dado.

| Entrada | $\lambda$ | aa | abba | abbbba | ababbaba | ababaababa | aabbaaaabbaa |
|---|---|---|---|---|---|---|---|
| **Tamaño** | 0 | 2 | 4 | 6 | 8 | 10 | 12 |
| **Pasos** | 3 | 9 | 15 | 21 | 27 | 33 | 39 |

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| **Dif 1** | | 6 | 6 | 6 | 6 | 6 | 6 |
| **Dif 2** | | | 0 | 0 | 0 | 0 | 0 |

### Cálculo de T(N)

Tras analizar los resultados de la tabla de diferencias finitas se observa que en la diferencia 1 se obtiene valores constantes, esto es un indicador de que el coste es de orden lineal. Se procederá a calcular los coeficientes de esta expresión.

$$T(N) = aN + b$$

$$a * 0 + b = 3$$

$$a * 2 + b = 9$$

Resolviendo el sistema se obtiene que a = 3 y b = 3. Concluyendo que, para este algoritmo:

$$T(N) = 3N + 3$$

### Cota superior asintótica ($n_0$ = 10)

$$g(N) = kN$$

$$kN > 3N + 3$$

$$k > 3 + \frac{3}{N}$$

$$Para\ N = 10 \ \rightarrow k > 3 + \frac{3}{10} \ \rightarrow \ k > \frac{330}{100}$$

$$g(N) = 3{,}3 * N$$

## 0.2. Evaluación de la mejora obtenida

Finalmente procederemos a realizar una comparación de las dos Máquinas de Turing anteriores, con una cinta y con 2 cintas. Como ya se ha calculado previamente el coste computacional para la MT Determinista de 1 cinta es cuadrático ya que recorrerá la cinta de extremo a extremo borrando los caracteres iguales.

Se tienen dos bucles que realizan dicha operación, un bucle interno que recorre la palabra de un extremo a otro y vuelta (siendo el doble que el tamaño de la palabra) y un bucle externo que mientras haya símbolos llamará al bucle interno.

Sin embargo; como se ha visto antes el coste de la MT de 2 cintas es lineal. La teoría nos indica que al utilizar MT con más cintas puede ocurrir que encontremos un algoritmo más eficiente que teniendo 1 cinta, ya que al añadir una cinta podríamos reducir el coste computacional. Teniendo una estructura de datos más elaborada se podrá simplificar el tiempo de procesamiento en comparación con estructuras más básicas con procesamiento muy complejo. Esto es lo que sucede en este caso, ya que encontramos un algoritmo que teniendo una MT de 2 cintas resuelve este problema en tiempo lineal.

## 0.3. MT No Determinista de 2 cintas

### Implementación

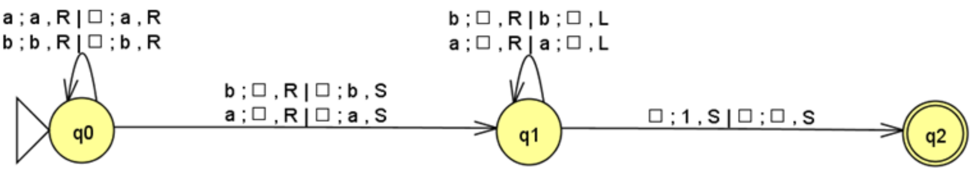



*Ilustración 3*

### Funcionamiento y peor caso

El funcionamiento de la máquina sigue la siguiente lógica:

6. Con el cabezal en la letra más a la izquierda de la palabra recorre toda la cinta 1 copiando la palabra en la cinta 2. A la vez puede pasar que el símbolo actual se borre ya que se ha llegado a la mitad de la cinta.

7. Se tiene entonces una tentativa de ND donde se crean instancias para los diferentes casos del paso 1, donde se copia la letra actual en la cinta 2 o se pasa a la comparación ya que se asume que es la mitad de la palabra.

8. A partir de la mitad de la palabra se comparará lo que queda de esta con lo copiado en la segunda cinta.

9. Si alguna de las instancias llega al estado final entonces se confirmará que la palabra es un palíndromo.

En este caso no se aceptarán las palabras con tamaño 0 ya que solo debe haber instancias que reconozcan un palíndromo. Como las instancias se irán cancelado cuando se encuentre un fallo en la comparación, entonces el peor caso es el de palíndromos pares que se consideren válidos para cada n.

### Simulación y diferencias

A continuación, se hace un análisis de la complejidad del algoritmo en función del tamaño de la palabra introducida, siempre recurriendo que la palabra pertenezca al "peor caso" para el tamaño dado.

| Entrada | aa | abba | abbbba | ababbaba | ababaababa | aabbaaaabbaa | aaabbaaaabbaaa |
|---|---|---|---|---|---|---|---|
| **Tamaño** | 2 | 4 | 6 | 8 | 10 | 12 | 14 |
| **Pasos** | 3 | 5 | 7 | 9 | 11 | 13 | 15 |

| | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| **Dif 1** | | 2 | 2 | 2 | 2 | 2 | 2 | 2 |
| **Dif 2** | | | 0 | 0 | 0 | 0 | 0 | 0 |

## Cálculo de T(N)

Tras analizar los resultados de la tabla de diferencias finitas se observa que en la diferencia 1 se obtiene valores constantes, esto es un indicador de que el coste es de orden lineal. Se procederá a calcular los coeficientes de esta expresión.

$$T(N) = aN + b$$

$$a * 2 + b = 3$$

$$a * 4 + b = 5$$

Resolviendo el sistema se obtiene que a =1 y b =1. Concluyendo que, para este algoritmo:

$$T(N) = N + 1$$

## Cota superior asintótica ($n_0$ = 10)

$$g(N) = kN$$

$$kN > N + 1$$

$$k > 1 + \frac{1}{N}$$

$$Para\ N = 10 \ \rightarrow k > 1 + \frac{1}{10} \rightarrow \ k > \frac{11}{10}$$

$$g(N) = 1{,}1 * N$$

# 1. SUMA DE ENTEROS EN BASE UNO

## 1.0. MT determinista con una cinta

### Implementación

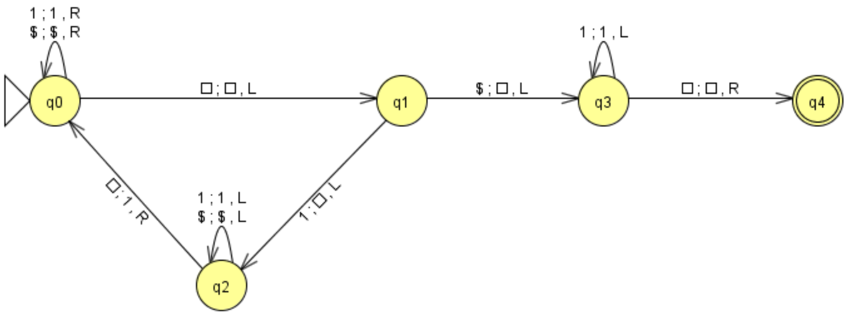



*Ilustración 4*

### Funcionamiento y peor caso

El funcionamiento de la máquina sigue la siguiente lógica:

1. Se recorre toda la cinta hasta el final.
2. Siempre que haya un 1 a la derecha del símbolo $ se borra.
3. Se vuelve a ir al inicio de la cinta y se escribe un uno, realizando así la suma.
4. Cuando no haya ningún uno más a la derecha del símbolo $ se borra y se vuelve al inicio de la cinta.

Dado que la iteración se repite tantas veces como es el tamaño del número a la derecha del $, para un cierto tamaño N, el peor de los casos se presenta cuando dedicamos una unidad de tamaño al $ y todas las demás N-1 a poner bits 1 en el número de la derecha. ya que se tendrán que copiar los 1 de esta parte a la izquierda del símbolo $.

Por ejemplo, para n = 3 el peor caso es: $11

### Simulación y diferencias

A continuación, se hace un análisis de la complejidad del algoritmo en función del tamaño de la palabra introducida, siempre recurriendo que la palabra pertenezca al “peor caso” para el tamaño dado.

| Entrada | $ | $1 | $11 | $111 | $1111 | $11111 |
|---|---|---|---|---|---|---|
| **Tamaño** | 1 | 2 | 3 | 4 | 5 | 6 |
| **Pasos** | 4 | 11 | 22 | 37 | 56 | 79 |

| | | | | | | |
|---|---|---|---|---|---|---|
| **Dif 1** | | 7 | 11 | 15 | 19 | 23 |
| **Dif 2** | | | 4 | 4 | 4 | 4 |
| **Dif 3** | | | | 0 | 0 | 0 |

## Cálculo de T(N)

Tras analizar los resultados de la tabla de diferencias finitas se observa que en la diferencia 2 se obtiene valores constantes, esto es un indicador de que el coste es de orden cuadrático. Se procederá a calcular los coeficientes de esta expresión.

$$T(N) = aN^2 + bN + c$$

$$a * 1^2 + b * 1 + c = 4$$

$$a * 2^2 + b * 2 + c = 11$$

$$a * 3^2 + b * 3 + c = 22$$

Resolviendo el sistema se obtiene que a = 2, b = 1 y c = 1.

$$T(N) = 2N^2 + N + 1$$

## Cota superior asintótica ($n_0$ = 10)

$$g(N) = kN^2$$

$$kN^2 > 2N^2 + N + 1$$

$$k > 2 + \frac{1}{N} + \frac{1}{N^2}$$

$$Para\ N = 10 \ \ \rightarrow k > 2 + \frac{1}{10} + \frac{1}{100}$$

$$k > \frac{211}{100}$$

$$g(N) = 2{,}11 * N^2$$

## 1.1. MT determinista de dos cintas

### Implementación

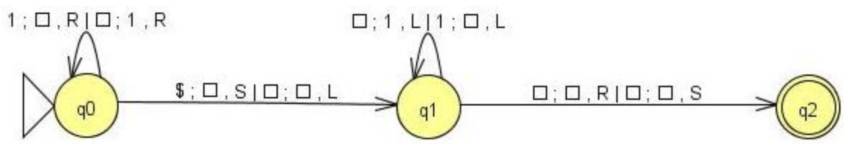



*Ilustración 5*

### Funcionamiento y peor caso

El funcionamiento de la máquina sigue la siguiente lógica:

1. Se escribe un 1 en la segunda cinta por cada 1 que haya en la cinta principal a la izquierda del símbolo $.

2. Cuando llega al símbolo $, este se borra y se añaden a la izquierda tantos unos como los que haya en la cinta secundaria.

3. Cuando no haya más símbolos a la izquierda de la cinta secundaria, el cabezal de la cinta principal se sitúa al inicio de esta, en el primer 1.

El peor caso para cada n será cuando todos los 1 se encuentran a la izquierda del símbolo $, ya que se tendrá que recorrer dos veces el número (n-1), una vez para copiarlo en la cinta secundaria y otra para volver a copiarlo en la cinta principal.

### Simulación y diferencias

A continuación, se hace un análisis de la complejidad del algoritmo en función del tamaño de la palabra introducida, siempre recurriendo que la palabra pertenezca al “peor caso” para el tamaño dado.

| **Entrada** | **$** | **1$** | **11$** | **111$** | **1111$** | **11111$** |
|---|---|---|---|---|---|---|
| **Tamaño** | 1 | 2 | 3 | 4 | 5 | 6 |
| **Pasos** | 2 | 4 | 6 | 8 | 10 | 12 |

| | | | | | | |
|---|---|---|---|---|---|---|
| **Dif 1** | | 2 | 2 | 2 | 2 | 2 |
| **Dif 2** | | | 0 | 0 | 0 | 0 |

### Cálculo de T(N)

Tras analizar los resultados de la tabla de diferencias finitas se observa que en la diferencia 1 se obtiene valores constantes, esto es un indicador de que el coste es de orden lineal. Se procederá a calcular los coeficientes de esta expresión.

$$T(N) = aN + b$$

$$a * 1 + b = 2$$

$$a * 2 + b = 4$$

Resolviendo el sistema se obtienen que a = 2 y b = 0.

$$T(N) = 2N$$

Cota superior asintótica ($n_0$ = 10)

$$g(N) = cN$$

$$cN > 2N$$

$$c > 2$$

$$Para\ N = 10 \ \ \rightarrow c > 2$$

$$g(N) = 2N$$

## 1.2. Evaluación de la mejora obtenida

Finalmente procederemos a realizar una comparación de las dos Máquinas de Turing anteriores, con 1 cinta y con 2 cintas.

La MT Determinista de 1 cinta tiene coste cuadrático, ya que recorre la cinta de extremo a extremo borrando y agregando un 1 a la izquierda cada vez. Esta MT tiene dos bucles, un bucle interno que copiará los 1 de la izquierda a la derecha, y un bucle externo que mientras halla 1 por sumar llamará al bucle interno.

Como se puede comprobar con una MT de 2 cintas se simplifica el coste a una complejidad lineal, ya que se puede copiar los 1 de la izquierda en un recorrido gracias a la utilización de una cinta adicional.

Esto es posible debido a que como indica la teoría al utilizar MT con más cintas puede encontrar un algoritmo más eficiente que teniendo 1 cinta, debido a que al añadir una cinta podríamos reducir el coste computacional.

Podemos concluir que una estructura simple de datos con un procesamiento más complejo puede ser mejorada por una estructura de datos elaborada que simplifica el tiempo de procesamiento.

# 2. SUMA DE ENTEROS EN BASE DOS

## 2.0. MT determinista con una cinta

### Implementación

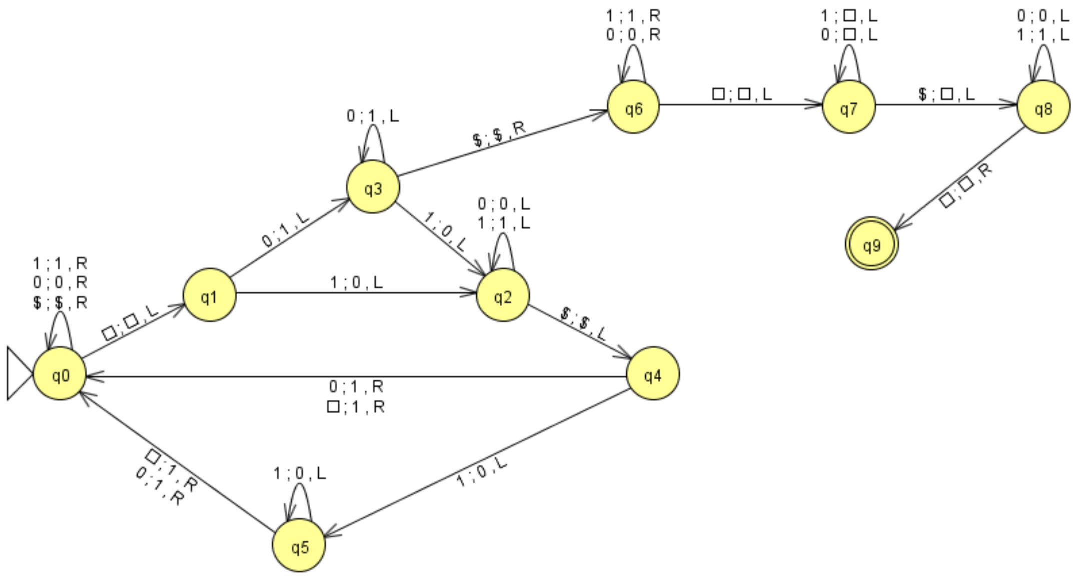



*Ilustración 6*

### Funcionamiento y peor caso

El funcionamiento de esta máquina se basa en sustracciones y adiciones unitarias. Más concretamente, la máquina resta una unidad al número de la derecha del $ y añade una unidad al número de la izquierda sucesivamente. Este proceso finaliza en el momento que el número de la derecha ha sido reducido a 0, en ese momento pasa a la etapa de limpiar la cinta, dejando solo el resultado de la izquierda del $.

Dado que la iteración se repite tantas veces como es el tamaño del número a la derecha del $, el peor caso de la cinta será aquel en el que maximicemos el número de iteraciones, o lo que es equivalente, el caso en el que maximicemos el número a la derecha del $.

Para un cierto tamaño N, la maximización presentada ocurre cuando dedicamos una unidad de tamaño al $ y todas las demás N-1 a poner bits 1 en el número de la derecha. Poniendo un ejemplo, si N = 3; conseguiremos el mayor número de pasos introduciendo en la cinta $11.

Al dejar en blanco el brazo izquierdo de la entrada, la máquina procesa como si se tratase de un 0, pero ahorramos el espacio al no poner explícitamente el 0, que podemos dedicar en aumentar el brazo derecho. Entendemos por brazo cada uno de los lados de la cinta separados por el símbolo $.

La máquina no acepta entradas de tamaño 1. El carácter introducido en la cinta debería ser el $, con lo que trataríamos de simbolizar la operación 0+0, pero la cinta no tolera este caso. De hecho, requiere un brazo derecho estrictamente no nulo.

## Simulación y diferencias

A continuación, se hace un análisis de la complejidad del algoritmo en función del tamaño de la palabra introducida, siempre recurriendo que la palabra pertenezca al "peor caso" para el tamaño dado.

| Entrada | $1 | $11 | $111 | $1111 | $11111 | $111111 | $1111111 | $11111111 |
|---|---|---|---|---|---|---|---|---|
| **Tamaño** | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 |
| **Pasos** | 17 | 42 | 99 | 228 | 517 | 1158 | 2567 | 5640 |

| | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| **Dif 1** | | 25 | 57 | 129 | 289 | 641 | 1409 | 3073 |
| **Dif 2** | | | 32 | 72 | 160 | 352 | 768 | 1664 |
| **Dif 3** | | | | 40 | 88 | 192 | 416 | 896 |
| **Dif 4** | | | | | 48 | 104 | 224 | 480 |
| **Dif 5** | | | | | | 56 | 120 | 256 |

No llegamos a diferencias constantes, por lo que la ecuación que rige esta sucesión debe ser algo más compleja que una polinómica.

## Cálculo de T(N)

Ecuación recurrente (la obtención de esta ecuación recurrente se presenta en el anexo):

$$a_{n+1} = 2a_n + 2^{n+1} - (n+1) + 3$$

Haciendo uso de la herramienta de recurrencia de WolframAlpha (link en bibliografía), se obtiene que la expresión general es la siguiente:

$$T(n) = n + 2^n (n + 2) - 1$$

## Demostración

La demostración se basa en la inducción matemática.

Caso base: $T(2) = 2 + 2^2 (2 + 2) - 1 = 17$

Lo asumimos cierto para todo n, y a partir de ello tratamos de llegar a la forma de T(n+1).

Es decir, trataremos de llegar a la fórmula:

$$T(n+1) = (n+1) + 2^{(n+1)} \big((n+1) + 2\big) - 1$$

Combinando la ecuación recurrente con la general:

$$T(n+1) = 2T(n) + 2^{n+1} - (n+1) + 3 = 2(n + 2^n (n + 2) - 1) + 2^{n+1} - (n+1) + 3$$
$$= 2n + 2^{n+1}(n+2) - 2 + 2^{n+1} - (n+1) + 3 = n + 2^{n+1}(n+3)$$

Aplicando transformaciones básicas y separando coeficientes:

$$n + 2^{n+1}(n+3) = n + 1 - 1 + 2^{n+1}(n+1+2)$$

Por último, reordenamos, y obtenemos la siguiente fórmula (como queríamos demostrar):

$$n + 1 - 1 + 2^{n+1}(n+1+2) = (n+1) + 2^{(n+1)} \big((n+1) + 2\big) - 1$$

### Cota superior asintótica ($n_0$ = 10)

$$g(n) = k * n * 2^n$$

$$k * n * 2^n > n + 2^n * n + 2 * 2^n - 1$$

$$k > \frac{1}{2^n} + 1 + \frac{2}{n} - \frac{1}{n * 2^n}$$

$$Para\ n_0 = 10 \ \rightarrow k > \frac{1}{2^{10}} + 1 + \frac{2}{10} - \frac{1}{10 * 2^{10}}$$

$$k > \frac{12297}{10240}$$

$$g(n) = \frac{12297}{10240} * n * 2^n$$

## 2.1. MT determinista de dos cintas

### Implementación

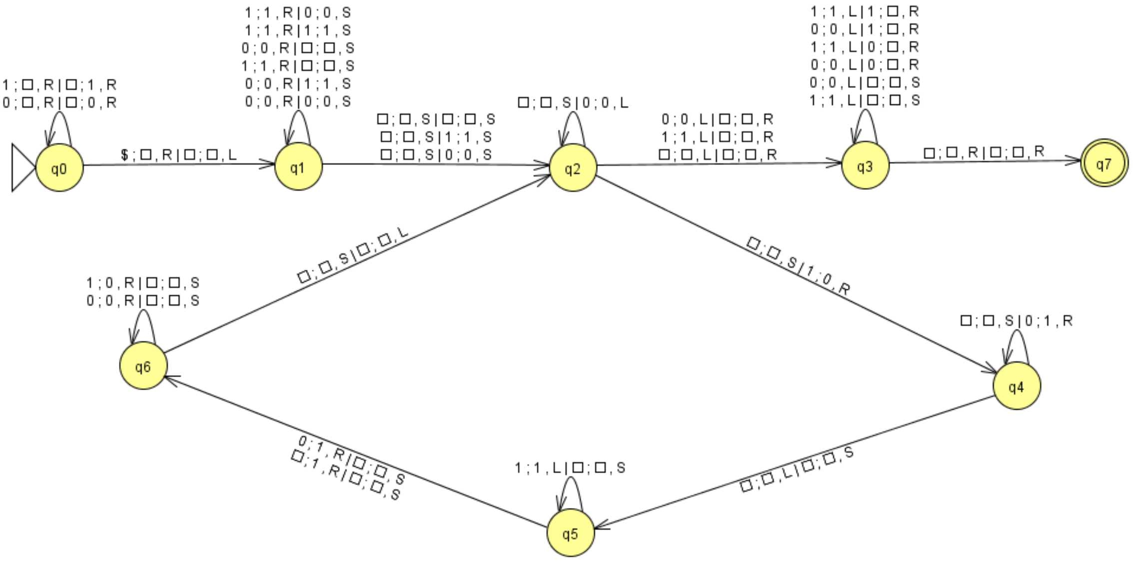


*Ilustración 7*

### Funcionamiento y peor caso

En primer lugar, se pasa uno de los números a la otra cinta, en este caso será el número situado a la izquierda del $ para poder comenzar a realizar la operación aplicando el método de Peano, que consiste en restar una unidad a uno de los dígitos, en este caso al que se sitúa en la segunda cinta para, a continuación, sumar uno al número de la cinta principal. Realizando esta operación tantas veces hasta que el segundo número sea cero, se obtendrá el resultado de la suma de ambos números. En este caso, se resta uno y se suma en el mismo bucle ya que, al contar con dos cintas esto es posible. Por último, se expresa el resultado de la forma correcta, es decir, la segunda cinta está vacía y la primera contiene el resultado.

Teniendo en cuenta el funcionamiento de esta máquina de Turing, el peor caso para un determinado tamaño se da cuando dedicamos una unidad de tamaño al $ y todas las demás N-1 a poner bits 1 en el número de la izquierda.

### Simulación y diferencias

A continuación, se hace un análisis de la complejidad del algoritmo en función del tamaño de la palabra introducida, siempre recurriendo que la palabra pertenezca al “peor caso” para el tamaño dado.

| Entrada | $ | 1$ | 11$ | 111$ | 1111$ | 11111$ | 111111$ |
|---|---|---|---|---|---|---|---|
| **Tamaño** | 1 | 2 | 3 | 4 | 5 | 6 | 7 |
| **Pasos** | 4 | 11 | 26 | 57 | 120 | 247 | 502 |

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| **Dif 1** | | 7 | 15 | 31 | 63 | 127 | 255 |
| **Dif 2** | | | 8 | 16 | 32 | 64 | 128 |
| **Dif 3** | | | | 8 | 16 | 32 | 64 |
| **Dif 4** | | | | | 8 | 16 | 32 |
| **Dif 5** | | | | | | 8 | 16 |

## Cálculo de T(N)

Observando el patrón que se repite a partir de Dif 2 se llega a la siguiente expresión:

$$T(N) = a2^N + bN + c$$

Resolviendo el siguiente sistema de ecuaciones:

$$a2^1 + 1b + c = 4$$

$$a2^2 + 2b + c = 11$$

$$a2^3 + 3b + c = 26$$

Se obtiene que a=4, b=-1 y c=-3 por lo que la expresión resultante es:

$$T(N) = 4 * 2^N - N - 3$$

## Cota superior asintótica ($n_0$ = 10)

$$g(N) = k * 2^N$$

$$k * 2^N > 4 * 2^N - N - 3$$

$$k > 4 - \frac{N}{2^N} - \frac{3}{2^N}$$

$$k > 4 - \frac{10}{2^{10}} - \frac{3}{2^{10}}$$

$$g(n) = 3.987 * 2^n$$

# 3. COMPARATIVA DE EJERCICIOS 1 Y 2

Puede resultar de interés hacer un análisis de las variaciones de complejidad entre los ejercicios 1 y 2 debido a que el objetivo de las máquinas es el mismo: sumar dos números. Por tanto, el principal factor que modifica las complejidades, asumiendo que los algoritmos son eficientes en ambos casos, es el cambio de base a la hora de expresar la entrada del número.

Por ello veremos cómo ha afectado dicho cambio de base en cada caso, MT Deterministas de una cinta y multicinta.

| Máquina | Base 1 | Base 2 |
|---|---|---|
| **MT Determinista 1 cinta** | $2N^2 + N + 1$ | $N + 2^N (N + 2) - 1$ |
| **MT Determinista multicinta** | $2N$ | $4 * 2^N - N - 3$ |

## Determinar la eficiencia de cada algoritmo

En primer lugar, comparando la eficiencia de las máquinas de Turing deterministas de una cinta, para una entrada del mismo tamaño se obtiene que en el caso de la suma en base uno la eficiencia obtenida es $O(N^2)$, mientras que, en el caso de la suma binaria, la eficiencia resultante es $O(N * 2^N)$, lo que supone una gran diferencia entre ambos algoritmos.

Por otro lado, al comparar las eficiencias para las máquinas deterministas multicinta, se obtiene también una gran diferencia, dando como resultado para la suma en base uno una eficiencia de $O(N)$, mientras que para la suma binaria la eficiencia es de $O(2^N)$.

## ¿Por qué la diferencia de complejidades?

Las diferencias en las complejidades y por lo tanto en la eficiencia de los algoritmos estudiadas en el apartado anterior se obtienen debido a que para la suma en base uno, para sumar una unidad a un número es suficiente con añadir un 1 a un lado de dicho número, siendo además indiferente el lado en el que se añada ya que en la representación de los números en base uno, la posición no es determinante.

Por el contrario, para realizar una suma en base binaria, se debe tener en cuenta la posición de los dígitos a diferencia del método anterior y, al utilizar el método de Peano, que consiste en restar una unidad a un número y sumar esa unidad al otro número hasta que el primero sea cero, incrementa la complejidad del algoritmo ya que se recorre el bucle tantas veces como el valor que tiene el número al que se le va restando una unidad cada vez.

Adicionalmente, hay que tener en cuenta que en la representación en base uno, para un tamaño de cinco el máximo número que se puede representar es cinco excluyendo del tamaño el símbolo $ que separa ambos números, mientras que en la representación binaria para un tamaño de cinco se pueden representar $2^5 = 32$ números, por lo que, al comparar la complejidad para un mismo tamaño no se tiene en cuenta que en la suma binaria para un mismo tamaño se pueden representar más números.

## ¿Cómo se interpretan las diferencias en complejidad y en eficiencia?

Recogiendo lo estudiado en los apartados anteriores, se obtienen que el algoritmo de la suma en base uno es más eficiente que el de la suma binaria para una entrada de mismo tamaño, sin embargo, hay que tener en cuenta que el primer algoritmo tienen algunas limitaciones como por ejemplo que se necesita más espacio para representar un número en base uno que en base binaria y, por lo tanto, si las limitaciones de espacio son muy grandes, puede ser mejor optar por el segundo algoritmo.

# 4. PALÍNDROMOS DE ORDEN K

## 4.0. MT multicinta determinista

### Implementación

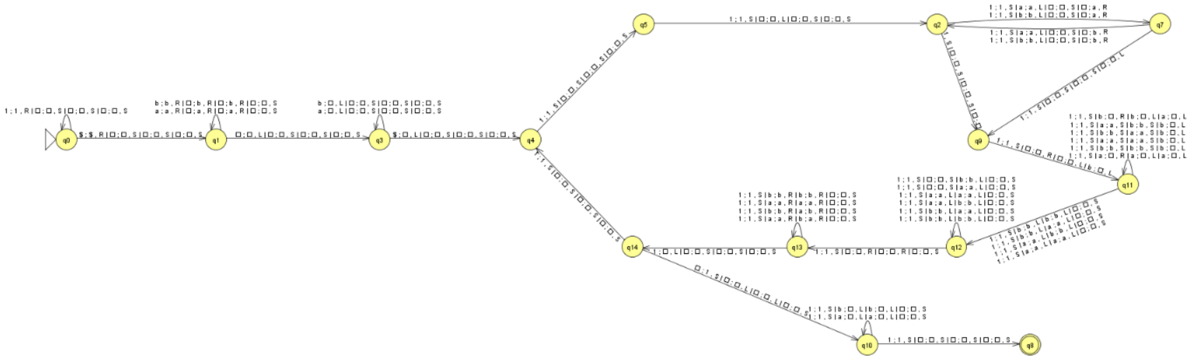


*Ilustración 8*

### Funcionamiento y peor caso

La máquina diseñada es una máquina de Turing de 4 cintas, cada una tiene una serie de funciones y entendiendo cada una de ellas lograremos entender el funcionamiento de la máquina.

Además, distinguimos dos etapas de funcionamiento, la primera etapa sirve de preparación y la segunda etapa es la etapa de iteración.

En la etapa de preparación se dejan las 4 cintas preparadas para el estado inicial que deben tener al inicio de la etapa de iteración. Esta preparación, cinta por cinta, consiste en:

- La primera cinta sirve para copiar la palabra en la segunda y la tercera cinta, la palabra se borra, junto con el símbolo de separación, y la posición del cabezal queda colocada en el bit menos significativo del número en base 1.

- En la segunda cinta se copia la palabra a partir de la primera cinta, el cabezal queda en el primer blanco a la derecha de la palabra.

- La tercera cinta es exactamente igual que la segunda.

- La cuarta cinta no se utiliza durante esta etapa.

En este contexto comienza la iteración de la máquina, en la que, por cintas y orden:

1. En la cuarta cinta se crea una palabra con el formato en expresión regular *(ab)*+a** y de longitud igual a la palabra existente en la cinta 2.

2. Si la cuarta cinta termina en *b*, (lo que indica que la palabra es de longitud par y por tanto es candidata a ser palíndroma), por cada *a* que hay en la cuarta cinta, se compara la letra más a la izquierda de la segunda cinta con la que está más a la izquierda de la tercera, si coinciden, se borran y se sigue avanzando, hasta se ha terminado (y borrado) la palabra de la cuarta cinta.

3. Finalmente replicamos la cinta 2 en la 3, y dejamos los cabezales listos para la siguiente iteración. Reducimos el contador de la primera cinta, si hemos terminado se borra todo y deja un 1 en la primera cinta, si no, volvemos al inicio de la iteración.

Dado que la máquina solo contendrá #1# en caso de aceptar. Las palabras no aceptadas no requieren tanta computación como las palabras aceptadas. Dado que para cada input tenemos dos variables sobre las que podemos hacer modificaciones de tamaño N (el tamaño de la palabra) y k (el orden del palíndromo). El análisis empírico se hará en función de las dos variables. En este caso se analizará el pero caso que son los palíndromos que tienen una combinación de a y b.

## Simulación y diferencias

A continuación, se hace un análisis de la complejidad del algoritmo en función del tamaño de la palabra introducida, siempre recurriendo que la palabra pertenezca al "peor caso" para el tamaño dado.

| **Entrada** | **1$aa** | **1$abba** | **1$abbbba** | **1$abbaabba** | **1$abbaaaabba** | **1$abbaaaaaabba** |
|---|---|---|---|---|---|---|
| **k (orden)** | 1 | 1 | 1 | 1 | 1 | 1 |
| **N** | 2 | 4 | 6 | 8 | 10 | 12 |
| **Pasos** | 22 | 33 | 44 | 55 | 66 | 77 |

| | | | | | | |
|---|---|---|---|---|---|---|
| **Dif 1** | | 11 | 11 | 11 | 11 | 11 |
| **Dif 2** | | | 0 | 0 | 0 | 0 |

| **Entrada** | **11$aaaa** | **11$abbaabba** | **11$abbbbaabbbba** | **11$abbaabbaabbaabba** |
|---|---|---|---|---|
| **k (orden)** | 2 | 2 | 2 | 2 |
| **N** | 4 | 8 | 12 | 16 |
| **Pasos** | 45 | 72 | 99 | 126 |

| | | | | |
|---|---|---|---|---|
| **Dif 1** | | 27 | 27 | 27 |
| **Dif 2** | | | 0 | 0 |

| **Entrada** | **111$aaaaaaa** | **111$abbaabbaabbaabba** | **111$abbbbaabbbbaabbbbaabbbba** | **111$abbbbbbaabbbbbbaabbbbbbaabbbbbba** |
|---|---|---|---|---|
| **k (orden)** | 3 | 3 | 3 | 3 |
| **N** | 8 | 16 | 24 | 32 |
| **Pasos** | 84 | 143 | 202 | 261 |

| | | | | |
|---|---|---|---|---|
| **Dif 1** | | 59 | 59 | 59 |
| **Dif 2** | | | 0 | 0 |

## Cálculo de T(N)

Tras analizar los resultados de las tablas de diferencias finitas de cada orden k se observa que en la diferencia 1 se obtiene valores constantes, esto es un indicador de que el coste es de orden lineal. Se procederá a calcular los coeficientes de esta expresión.

Observando el patrón que se repite a partir de Dif 1 se llega a la siguiente expresión:

$$T(N) = aN + b$$

Ahora procedemos a resolver el sistema de ecuaciones para cada valor de k estudiado:

$$k = 1$$

$$a * 2 + b = 22$$

$$a * 4 + b = 33$$

Resolviendo el sistema se obtienen que a = $\frac{11}{2}$ y b = 11

$$T(N) = \frac{11}{2}N + 11$$

$$k = 2$$

$$a * 4 + b = 45$$

$$a * 8 + b = 72$$

Resolviendo el sistema se obtienen que a = $\frac{27}{4}$ y b = 18

$$T(N) = \frac{27}{4}N + 18$$

$$k = 3$$

$$a * 8 + b = 84$$

$$a * 16 + b = 143$$

Resolviendo el sistema se obtienen que a = $\frac{59}{8}$ y b = 25

$$T(N) = \frac{59}{8}N + 25$$

A partir de las 3 complejidades estudiadas podemos deducir la ecuación general T(N, k) observando que con los aumentos de la k, el coeficiente b incrementa en 7 unidades por cada unidad de k. Por otro lado, el coeficiente a, en la primera iteración aumenta 5/4, mientras que para la segunda aumenta 5/8, o lo que es lo mismo $5/2^{k+1}$.

Podemos por tanto hacer una forma general extendida de la siguiente forma:

$$T(N,k) = \left(3 + \left(\frac{5}{2^1} + \frac{5}{2^2} + \cdots + \frac{5}{2^k}\right)\right)N + 4 + 7k$$

Operando:

$$T(N,k) = \left(3 + 5\left(\frac{2^{k-1} + 2^{k-2} + \cdots + 2^0}{2^k}\right)\right)N + 4 + 7k$$

O lo que es lo mismo:

$$T(N,k) = \left(3 + 5\left(\frac{2^k - 1}{2^k}\right)\right)N + 4 + 7k$$

Cota superior asintótica ($n_0$ = 10)

$$g(N) = cN$$

$$k = 1 \rightarrow cN > \frac{11}{2}N \rightarrow c > \frac{11}{2}$$

$$Para\ N = 10 \rightarrow c > \frac{11}{2}$$

$$g(N) = 5.5N$$

$$k = 2 \rightarrow cN > \frac{27}{4}N \rightarrow c > \frac{27}{4}$$

$$Para\ N = 10 \rightarrow c > \frac{27}{4}$$

$$g(N) = 6.75\,N$$

$$k = 3 \rightarrow cN > \frac{59}{8}N \rightarrow c > \frac{59}{8}$$

$$Para\ N = 10 \rightarrow c > \frac{59}{8}$$

$$g(N) = 7.375\,N$$

## 4.1 MT multicinta No determinista

### Implementación

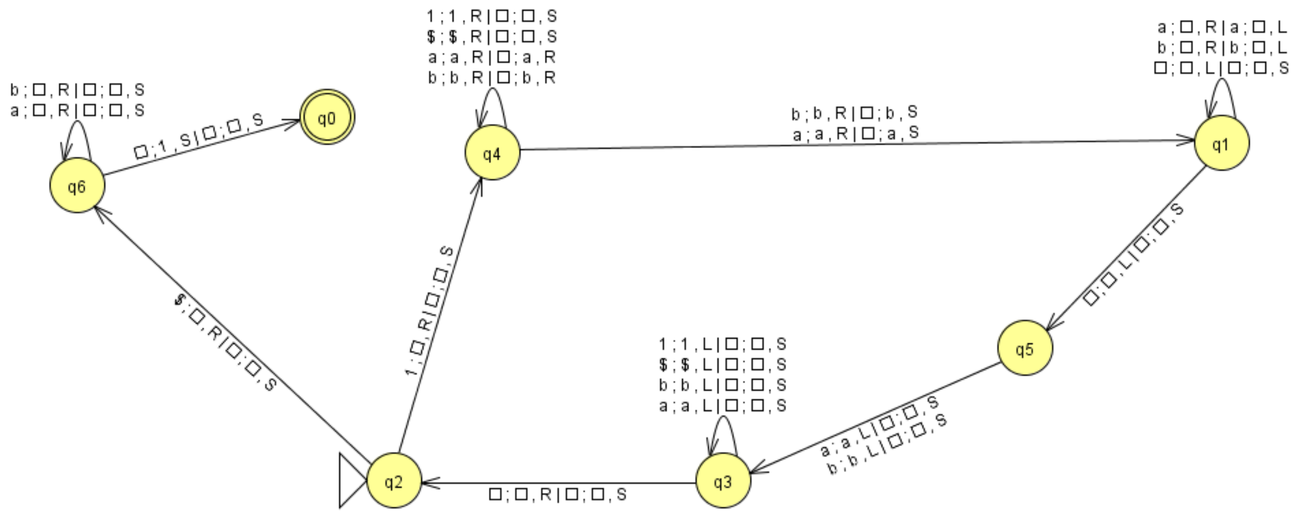



*Ilustración 9*

### Funcionamiento y peor caso

Esta MT se basa en la implementada en el apartado 0, agregando estados que permitan la iteración analizar si un orden k y una palabra x es palíndroma de al menos orden k. Para esto a la izquierda del $ se tiene un número en base 1 que representa el orden del palíndromo, mientras a la derecha del $ se tiene la palabra a la cual se desea comprobar.

El funcionamiento de la MT empieza restando 1 al número k para ir reduciendo el número de veces que se comprueba que la palabra es palíndroma, si en esta posición tendríamos k = 0 (es decir ningún valor a la izquierda de $), el palíndromo se daría por válido y se terminaría la ejecución. Luego el cabezal recorre la cinta pasando por el número k y el símbolo $. En el caso de los caracteres a y b se crean diferentes instancias dependiendo de si el siguiente valor en la cinta se copia en la cinta secundaria o se señala que se ha llegado a la mitad, esto se hace ya que como no se sabe que valor es la mitad solo una de las instancias creadas dividirá la palabra correctamente. En todas las instancias se comparará lo que queda de la cinta principal con lo copiado en la secundaria, borrando a la vez ambos caracteres de las dos cintas. En caso de éxito en la comparación se recorre la cinta hasta el inicio para comprobar que el k todavía no es 0. En caso de k>0 se realizarán los pasos anteriores nuevamente, mientras si debido a las diferentes comprobaciones la k = 0 se borrarán los datos de la cinta y se escribirá un 1 indicando que es un palíndromo válido de al menos orden k.

Como las instancias se irán cancelado cuando se encuentre un fallo en la comparación, el peor caso es el de palíndromos pares que se consideren válidos para cada k>0. Además, serán los palíndromos que tienen una combinación de a y b.

### Simulación y diferencias

A continuación, se hace un análisis de la complejidad del algoritmo en función del tamaño de la palabra introducida, siempre recurriendo que la palabra pertenezca al “peor caso” para el tamaño dado. En este caso se separará el análisis dependiendo del orden k que se quiere comparar.

| Entrada | 1$aa | 1$abba | 1$abbbba | 1$abbaabba | 1$abbaaaabba | 1$abbaaaaaabba |
|---|---|---|---|---|---|---|
| **k (orden)** | 1 | 1 | 1 | 1 | 1 | 1 |
| **N** | 2 | 4 | 6 | 8 | 10 | 12 |
| **Pasos** | 12 | 17 | 22 | 27 | 32 | 37 |

| | | | | | | |
|---|---|---|---|---|---|---|
| **Dif 1** | | 5 | 5 | 5 | 5 | 5 |
| **Dif 2** | | | 0 | 0 | 0 | 0 |

| Entrada | 11$aaaa | 11$abbaabba | 11$abbbbaabbbba | 11$abbaabbaabbaabba |
|---|---|---|---|---|
| **k (orden)** | 2 | 2 | 2 | 2 |
| **N** | 4 | 8 | 12 | 14 |
| **Pasos** | 27 | 40 | 53 | 66 |

| | | | | |
|---|---|---|---|---|
| **Dif 1** | | 13 | 13 | 13 |
| **Dif 2** | | | 0 | 0 |

| Entrada | 111$aaaaaaaa | 111$abbaabbaabbaabba | 111$abbbbaabbbbaabbbbaabbbba | 111$abbbbbbaabbbbbbaabbbbbbaabbbbbba |
|---|---|---|---|---|
| **k (orden)** | 3 | 3 | 3 | 3 |
| **N** | 8 | 16 | 24 | 32 |
| **Pasos** | 52 | 81 | 110 | 139 |

| | | | | |
|---|---|---|---|---|
| **Dif 1** | | 29 | 29 | 29 |
| **Dif 2** | | | 0 | 0 |

## Cálculo de T(N)

Tras analizar los resultados de las tablas de diferencias finitas de cada orden k se observa que en la diferencia 1 se obtiene valores constantes, esto es un indicador de que el coste es de orden lineal. Se procederá a calcular los coeficientes de esta expresión.

Observando el patrón que se repite a partir de Dif 1 se llega a la siguiente expresión:

$$T(N) = aN + b$$

Ahora procedemos a resolver el sistema de ecuaciones para cada valor de k estudiado:

$$k = 1$$

$$a * 2 + b = 12$$

$$a * 4 + b = 17$$

Resolviendo el sistema se obtienen que a = $\frac{5}{2}$ y b = 7

$$T(N) = \frac{5}{2}N + 7$$

$$k = 2$$

$$a * 4 + b = 27$$

$$a * 8 + b = 40$$

Resolviendo el sistema se obtienen que a = $\frac{13}{4}$ y b = 14

$$T(N) = \frac{13}{4}N + 14$$

$$k = 3$$

$$a * 8 + b = 52$$

$$a * 16 + b = 81$$

Resolviendo el sistema se obtienen que a = $\frac{29}{8}$ y b = 23

$$T(N) = \frac{29}{8}N + 23$$

En base a estas ecuaciones y alguna prueba suelta adicional hemos podido deducir la expresión en función de las dos variables T(N, k).

El término independiente ha aumentado +7 en la primera subida de k y +9 en la segunda, si aplicamos diferencias finitas a esa progresión 7,14,23 vemos que tiene la forma $ak^2 + bk + c$. Resolvemos y obtenemos a = 1, b = 4, c = 2. Por lo que ya tenemos el valor de lo que hasta ahora era término independiente en función de k.

Para el cálculo del término que acompaña a la N en función de k aplicamos una lógica similar al caso de la máquina determinista. De K = 1 a K = 2 el término aumenta $3/2^k$, y sucesivamente se repite ese incremento, por lo que podemos llegar a:

$$T(N,k) = \left(1 + \left(\frac{3}{2^1} + \frac{3}{2^2} + \cdots + \frac{3}{2^k}\right)\right)N + k^2 + 4k + 2$$

También de forma análoga al caso determinista llegamos a la expresión:

$$T(N,k) = \left(1 + 3\left(\frac{2^k - 1}{2^k}\right)\right)N + k^2 + 4k + 2$$

### Cota superior asintótica ($n_0$ = 10)

$$g(N) = cN$$

$$k = 1 \rightarrow cN > \frac{5}{2}N \rightarrow c > \frac{5}{2}$$

$$Para\ N = 10 \rightarrow c > \frac{5}{2}$$

$$g(N) = 2.5N$$

$$k = 2 \;\rightarrow\; cN > \frac{13}{4}\,N \rightarrow c > \frac{13}{4}$$

$$Para\ N = 10 \quad \rightarrow c > \frac{13}{4}$$

$$g(N) = 3.25\,N$$

$$k = 3 \;\rightarrow\; cN > \frac{29}{8}\,N \rightarrow c > \frac{29}{8}$$

$$Para\ N = 10 \quad \rightarrow c > \frac{29}{8}$$

$$g(N) = 3.625\,N$$

## 4.2 MT Determinista con una cinta (Opcional)

### Implementación

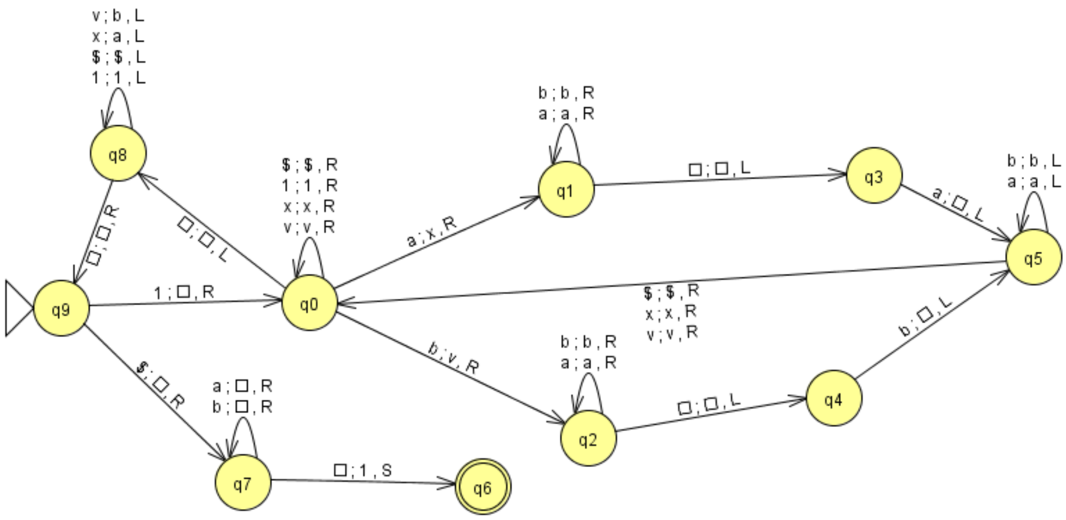



*Ilustración 10*

### Funcionamiento y peor caso

Esta MT se basa en la implementada en el apartado 0, agregando estados que permitan la recursividad al analizar si un orden k y una palabra x es palíndroma de al menos orden k. Para esto a la izquierda del $ se tiene un número en base 1 que representa el orden del palíndromo, mientras a la derecha del $ se tiene la palabra a la cual se desea comprobar. Lo primero que se hace es restar 1 del número k para ir reduciendo el número de veces que se comprueba que la palabra es palíndroma, si en esta posición tendríamos k = 0 (es decir ningún valor a la izquierda de $), el palíndromo se daría por válido y se terminaría la ejecución. Lo siguiente será evaluar la palabra eliminando caracteres del final y sustituyendo a y b por x y v respectivamente en el inicio de la secuencia. Si encontramos fallo en la secuencia se terminará la ejecución ya que solo se contempla terminar cuando sea válido. Por otro lado, al realizar la comparación y ver que es correcta se transformarán todos los símbolos adicionales x y v en sus valores iniciales. Entonces se comprobará que el k todavía no es 0, si no lo es se realiza la secuencia anterior, mientras que si lo es se borrará toda la cinta y se pone 1, llegando al estado final y dando como válido este palíndromo de al menos orden k.

Como se ha explicado cuando una palabra presenta error no continuará con la comparación y terminará por lo que el peor caso serán los palíndromos pares válidos para cada k>0. Además, serán los palíndromos que tienen una combinación de a y b.

## Simulación y diferencias

A continuación, se hace un análisis de la complejidad del algoritmo en función del tamaño de la palabra introducida, siempre recurriendo que la palabra pertenezca al "peor caso" para el tamaño dado. En este caso se separará el análisis dependiendo del orden k que se quiere comparar.

| Entrada | 1$aa | 1$abba | 1$abbbba | 1$abbaabba | 1$abbaaaabba | 1$abbaaaaaabba |
|---|---|---|---|---|---|---|
| **k (orden)** | 1 | 1 | 1 | 1 | 1 | 1 |
| **N** | 2 | 4 | 6 | 8 | 10 | 12 |
| **Pasos** | 14 | 25 | 40 | 59 | 82 | 109 |

| | | | | | | |
|---|---|---|---|---|---|---|
| **Dif 1** | | 11 | 15 | 19 | 23 | 27 |
| **Dif 2** | | | 4 | 4 | 4 | 4 |
| **Dif 3** | | | | 0 | 0 | 0 |

| Entrada | 11$aaaa | 11$abbaabba | 11$abbbbaabbbba | 11$abbaabbaabbaabba |
|---|---|---|---|---|
| **k (orden)** | 2 | 2 | 2 | 2 |
| **N** | 4 | 8 | 12 | 14 |
| **Pasos** | 37 | 80 | 143 | 226 |

| | | | | |
|---|---|---|---|---|
| **Dif 1** | | 43 | 63 | 83 |
| **Dif 2** | | | 20 | 20 |
| **Dif 3** | | | | 0 |

| Entrada | 111$aaaaaaaa | 111$abbaabbaabbaabba | 111$abbbbaabbbbaabbbbaabbbba | 111$abbbbbbaabbbbbbaabbbbbbaabbbbbba |
|---|---|---|---|---|
| **k (orden)** | 3 | 3 | 3 | 3 |
| **N** | 8 | 16 | 24 | 32 |
| **Pasos** | 94 | 249 | 488 | 811 |

| | | | | |
|---|---|---|---|---|
| **Dif 1** | | 155 | 239 | 323 |
| **Dif 2** | | | 84 | 84 |
| **Dif 3** | | | | 0 |

## Cálculo de T(N)

Tras analizar los resultados de las tablas de diferencias finitas de cada orden k se observa que en la diferencia 2 se obtiene valores constantes, esto es un indicador de que el coste es de orden cuadrático. Se procederá a calcular los coeficientes de esta expresión.

Observando el patrón que se repite a partir de Dif 2 se llega a la siguiente expresión:

$$T(N) = aN^2 + bN + c$$

Ahora procedemos a resolver el sistema de ecuaciones para cada valor de k estudiado:

$$k = 1$$

$$a * 2^2 + b * 2 + c = 14$$

$$a * 4^2 + b * 4 + c = 25$$

$$a * 6^2 + b * 6 + c = 40$$

Resolviendo el sistema se obtiene que a = $\frac{1}{2}$, b = $\frac{5}{2}$, y c = 7.

$$T(N) = \frac{1}{2}N^2 + \frac{5}{2}N + 7$$

$$k = 2$$

$$a * 4^2 + b * 4 + c = 37$$

$$a * 8^2 + b * 8 + c = 80$$

$$a * 12^2 + b * 12 + c = 143$$

Resolviendo el sistema se obtiene que a = $\frac{5}{8}$, b = $\frac{13}{4}$, y c = 14.

$$T(N) = \frac{5}{8}N^2 + \frac{13}{4}N + 14$$

$$k = 3$$

$$a * 8^2 + b * 8 + c = 94$$

$$a * 16^2 + b * 16 + c = 249$$

$$a * 24^2 + b * 24 + c = 488$$

Resolviendo el sistema se obtiene que a = $\frac{21}{32}$, b = $\frac{29}{8}$, y c = 23.

$$T(N) = \frac{21}{32}N^2 + \frac{29}{8}N + 23$$

La progresión del término independiente resulta muy familiar, del cálculo de la máquina no determinista, por ello sabemos que lo podemos expresar como $k^2 + 4k + 2$.

Por otra parte, el término lineal en N también sufre un incremento similar a los casos anteriores, por lo que podemos deducir su expresión general como:

$$\left(1+3\left(\frac{2^k-1}{2^k}\right)\right)N$$

Por último, debemos tratar de buscar una expresión en función de k para el término cuadrático:

$$\frac{1}{2^1}+\frac{1}{2^3}+\cdots+\frac{1}{2^{2k-1}}$$

Operando:

$$\frac{2^{2k-2}+2^{2k-4}+2^{2k-6}+\cdots+2^0}{2^{2k-1}}=\frac{2^{2(k-1)}+2^{2(k-2)}+2^{2(k-3)}+\cdots+2^0}{2^{2k-1}}=\frac{2^2 2^{k-1}+2^2 2^{k-2}+2^2 2^{k-3}+\cdots+2^0}{2^{2k-1}}=$$

$$\frac{4(2^k-1)}{2^{2k-1}}$$

Juntando las tres partes, obtenemos la expresión general en función de N y k:

$$T(N,k)=\frac{4(2^k-1)}{2^{2k-1}}N^2+\left(1+3\left(\frac{2^k-1}{2^k}\right)\right)N+k^2+4k+2$$

## Cota superior asintótica ($n_0$ = 10)

$$g(N)=kN^2$$

$$k=1\rightarrow kN^2>\frac{1}{2}N^2+\frac{5}{2}N+7\rightarrow k>\frac{1}{2}+\frac{5}{2N}+\frac{7}{N^2}$$

$$Para\ N=10\ \rightarrow k>\frac{1}{2}+\frac{1}{20}+\frac{1}{100}$$

$$k>\frac{14}{25}$$

$$g(N)=0{,}56*N^2$$

$$k=2\rightarrow kN^2>\frac{5}{8}N^2+\frac{13}{4}N+14\rightarrow k>\frac{5}{8}+\frac{13}{4N}+\frac{14}{N^2}$$

$$Para\ N=10\ \rightarrow k>\frac{1}{2}+\frac{1}{40}+\frac{1}{100}$$

$$k>\frac{107}{200}$$

$$g(N)=0{,}535*N^2$$

$$k=3\rightarrow kN^2>\frac{5}{8}N^2+\frac{13}{4}N+14\rightarrow k>\frac{5}{8}+\frac{13}{4N}+\frac{14}{N^2}$$

$$Para\ N=10\ \rightarrow k>\frac{1}{2}+\frac{1}{40}+\frac{1}{100}$$

$$k>\frac{107}{200}$$

$$g(N) = 0{,}535 * N^2$$

## 4.3 Comparativa MD multicinta determinista y no determinista

Para realizar la comparativa en primer lugar vamos a presentar brevemente la complejidad T(N,k) en cada uno de los casos y la tabla para k=3 de cada una.

Caso determinista:

$$T(N,k) = \left(3 + 5\left(\frac{2^k - 1}{2^k}\right)\right)N + 4 + 7k$$

| Entrada | 111$aaaaaaaa | 111$abbaabbaabbaabba | 111$abbbbaabbbbaabbbbaabbbba | 111$abbbbbbaabbbbbbaabbbbbbaabbbbbba |
|---|---|---|---|---|
| **k (orden)** | 3 | 3 | 3 | 3 |
| **N** | 8 | 16 | 24 | 32 |
| **Pasos** | 84 | 143 | 202 | 261 |

| | | | | |
|---|---|---|---|---|
| **Dif 1** | | 59 | 59 | 59 |
| **Dif 2** | | | 0 | 0 |

Caso no determinista:

$$T(N,k) = \left(1 + 3\left(\frac{2^k - 1}{2^k}\right)\right)N + k^2 + 4k + 2$$

| Entrada | 111$aaaaaaaa | 111$abbaabbaabbaabba | 111$abbbbaabbbbaabbbbaabbbba | 111$abbbbbbaabbbbbbaabbbbbbaabbbbbba |
|---|---|---|---|---|
| **k (orden)** | 3 | 3 | 3 | 3 |
| **N** | 8 | 16 | 24 | 32 |
| **Pasos** | 52 | 81 | 110 | 139 |

| | | | | |
|---|---|---|---|---|
| **Dif 1** | | 29 | 29 | 29 |
| **Dif 2** | | | 0 | 0 |

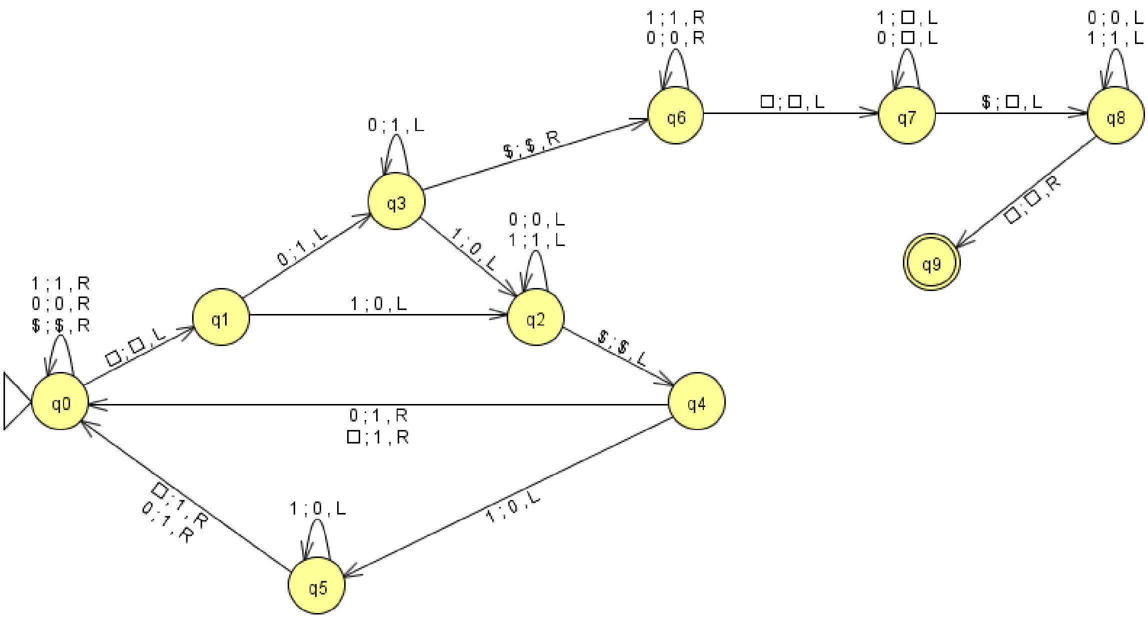
Como vemos, la tabla refleja valores inferiores para el caso no determinista. Podríamos pensar que la tendencia se va a revertir dado que el caso determinista es cuadrático en K. Sin embargo, el hecho de que el término que acompaña a N sea superior en la determinista frente a la no determinista hace que ese pensamiento no se cumpla, pues la N también guarda una cierta relación con k para que la palabra sea válida.

Por ello pese a que si tanto N como k fuesen arbitrarios la complejidad obtenida con el caso determinista sería superior, en este caso en el que N y k tienen una relación en la que N es, al menos, $2^k$, la complejidad de la máquina determinista es mayor.

# 5. ANEXO

## 5.0. Mejora en la suma binaria

### Implementación




Se ha creado el siguiente algoritmo para realizar la suma binaria que tiene una complejidad mucho menor a la que se ha obtenido anteriormente, pero este resultado se obtiene sin utilizar el método de Peano, que es lo que se pedía en la práctica.

### Simulación y diferencias

| Input | Tamaño | Pasos |
|---|---|---|
| **$** | 1 | 3 |
| **$1** | 2 | 5 |
| **$11** | 3 | 7 |
| **$111** | 4 | 9 |
| **$1111** | 5 | 11 |
| **$11111** | 6 | 13 |
| **$111111** | 7 | 15 |

| N | 1 | 2 | 3 | 4 | 5 | 6 | 7 |
|---|---|---|---|---|---|---|---|
| **Pasos** | 3 | 5 | 7 | 9 | 11 | 13 | 15 |
| **Dif 1** | | 2 | 2 | 2 | 2 | 2 | 2 |
| **Dif 2** | | | 0 | 0 | 0 | 0 | 0 |

### Cálculo de T(N)

$$T(N) = aN + b$$
$$a * 1 + b = 3$$
$$a * 2 + b = 5$$

Resolviendo el sistema se obtienen que a = 2 y b = 1.

$$T(N) = 2N + 1$$

Cota superior asintótica ($n_0$ = 10)

$$g(N) = cN$$
$$cN > 2N + 1$$
$$c > 2 + \frac{1}{N}$$
$$Para\ N = 10 \quad \rightarrow c > 2 + \frac{1}{10}$$
$$c > \frac{21}{10}$$
$$g(N) = 2{,}1N$$

## 5.1. Obtención de la ecuación recurrente en MT determinista para suma binaria

Dado que la obtención de una expresión general entraña mucha dificultad, la forma de atacar este problema se basa en buscar la relación entre dos números consecutivos en la sucesión de pasos en función de tamaño.

Es decir, en la siguiente tabla:

| Entrada | $1 | $11 | $111 | $1111 | $11111 | $111111 | $1111111 | $11111111 |
|---|---|---|---|---|---|---|---|---|
| **Tamaño** | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 |
| **Pasos** | 17 | 42 | 99 | 228 | 517 | 1158 | 2567 | 5640 |

Ver cómo podemos expresar la sucesión 17, 42, 99, 228…Para ello se han probado varias formas, pero la más útil ha sido al tratar de expresar un número en base al anterior multiplicado por 2. De esta forma:

$$\boxed{17} \xrightarrow{x2\ +\ 8} \boxed{42} \xrightarrow{x2\ +\ 15} \boxed{99} \xrightarrow{x2\ +\ 30} \boxed{228} \xrightarrow{x2\ +\ 61} \boxed{517}$$

Donde podemos ver que la relación posiblemente tenga la siguiente forma:

$$a_{n+1} = 2 * a_n + ruido_0$$

Lo siguiente destacable es que el residuo que queda en forma de suma sigue la progresión 8, 15, 30, 61. Esta sucesión resulta fácil de relacionar con las potencias de dos. Así:

| | | | |
|---|---|---|---|
| 8 | 15 | 30 | 61 |
| $2^3$ | $2^4$ -1 | $2^5$ - 2 | $2^6$ - 3 |

Así podemos despejar un poco el ruido, obteniendo una relación más precisa:

$$a_{n+1} = 2 * a_n + 2^{n+1} + ruido_1$$

Por último, tratamos de ver la relación del ruido, en el paso a T(N) = 42, a lo que es lo mismo desde N = 2 a N+1, podemos observar que el ruido es 0. De ahí en adelante, el ruido disminuye una unidad en cada paso, al mismo ritmo que se incrementa N. Por ello podemos deducir que el ruido actual es lineal respecto a N, con coeficiente -1, y constante por determinar.

$$ruido_1 = -1 * (N + 1) + c$$

$$0 = -1 * 3 + c$$

$$c = 3$$

$$Por\ tanto: \quad ruido_1 = -1 * N + 3$$

Finalmente uniendo todos los pasos:

$$a_{n+1} = 2 * a_n + 2^{n+1} - (n + 1) + 3$$

$$Con\ valor\ inicial\ a_2 = 17$$

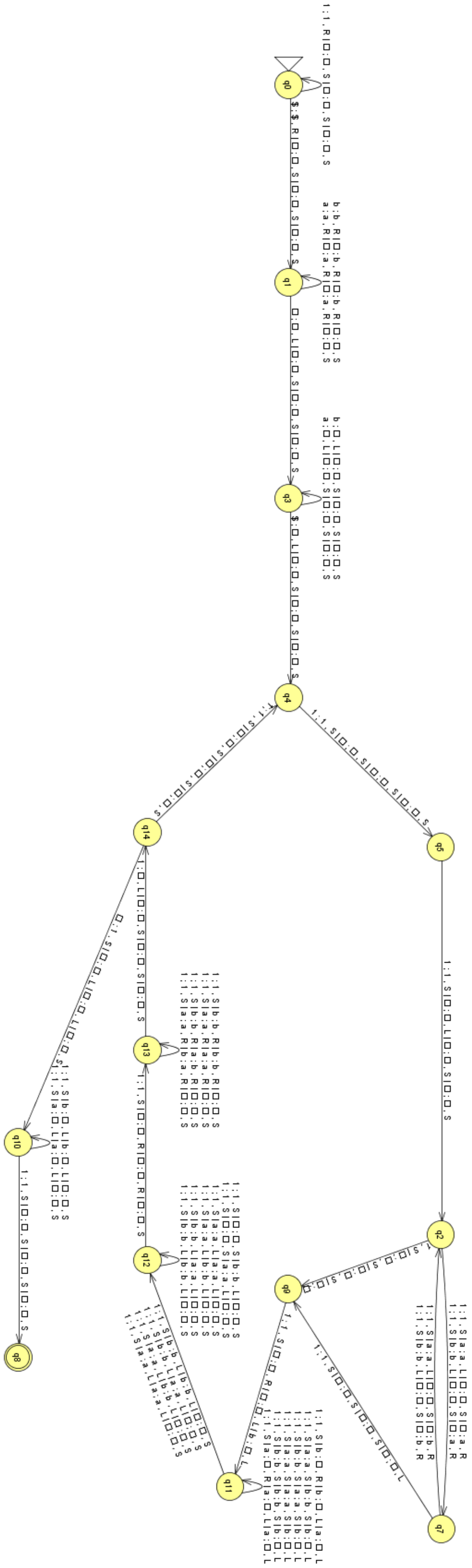
Se puede comprobar que todos los valores se pueden obtener a partir del anterior mediante esta ecuación de recurrencia, no presentaremos todas las comprobaciones en este trabajo, pero han sido hechas a través de una hoja Excel, presentamos el paso a $a_3$.

$$a_3 = 2 * a_2 + 2^3 - (3) + 3$$

$$a_3 = 2 * 17 + 2^3 - (3) + 3$$

$$a_3 = 42$$

## 5.2. Visualización ampliada de MT determinista multicinta



# 6. Bibliografía

https://www.wolframalpha.com/examples/mathematics/discrete-mathematics/recurrences/