Análisis de Complejidad
Máquina de Turing para la Sucesión de Fibonacci

Proyecto 1 - Análisis y Diseño de Algoritmos

27 de febrero de 2026

Índice

1. Introducción

Este documento presenta el análisis formal de complejidad de una Máquina de Turing determinista de una sola cinta que calcula la sucesión de Fibonacci.

El objetivo principal es estudiar el comportamiento temporal del algoritmo implementado y justificar formalmente por qué su crecimiento es exponencial.

2. Modelo de Entrada

La entrada está codificada en notación unaria. Si el número de entrada es n, entonces la longitud de la cinta inicial es proporcional a n.

Por ejemplo:

$$5 \rightarrow 11111$$

El tamaño de la entrada es entonces:

$$|entrada| = n$$

3. Definición Matemática de Fibonacci

La sucesión de Fibonacci se define como:

$$F(n) = F(n-1) + F(n-2)$$

con condiciones iniciales:

$$F(1) = 1, \quad F(2) = 1$$

4. Crecimiento Exponencial

La solución cerrada de Fibonacci está dada por la fórmula de Binet:

$$F(n) = \frac{\varphi^n - (1-\varphi)^n}{\sqrt{5}}$$

donde:

$$\varphi = \frac{1+\sqrt{5}}{2}$$

Para valores grandes de n, el segundo término se vuelve despreciable, por lo tanto:

$$F(n) \approx \frac{\varphi^n}{\sqrt{5}}$$

Por lo tanto:

$$F(n) \in \Theta(\varphi^n)$$

Esto demuestra que la sucesión crece exponencialmente.

5. Análisis Temporal en la Máquina de Turing

La máquina implementada:

- Utiliza una sola cinta
- Recorre la cinta múltiples veces
- Reescribe y copia bloques completos
- No utiliza almacenamiento auxiliar externo

Cada término $F(n)$ se construye sumando explícitamente $F(n-1)$ y $F(n-2)$ en la cinta.

Dado que:

$$F(n) \approx \varphi^n$$

y la máquina debe escribir todos esos símbolos en la cinta, el número de operaciones es proporcional a la magnitud del número generado.

Por tanto:

$$T(n) \in \Theta(\varphi^n)$$

Es decir, la complejidad temporal es exponencial.

6. Intuición del Crecimiento

En cada iteración:

- Se escanean bloques completos
- Se duplican segmentos
- Se realizan múltiples desplazamientos de cabeza

El tamaño de la cinta crece exponencialmente, por lo que el tiempo necesario para recorrerla también crece exponencialmente.

7. Complejidad Espacial

La máquina almacena todos los términos anteriores en la misma cinta.

Dado que el tamaño del último término es:

$$F(n) \approx \varphi^n$$

La memoria utilizada también es:

$$S(n) \in \Theta(\varphi^n)$$

Por lo tanto, la complejidad espacial también es exponencial.

8. Comparación con Implementaciones Óptimas

Una implementación iterativa clásica en un lenguaje de alto nivel puede calcular Fibonacci en:

$$O(n)$$

Sin embargo, en este caso:

- La representación es unaria
- Cada unidad requiere un símbolo
- La máquina debe simular cada operación elemental

Por ello, incluso si el número de iteraciones es n, el tamaño del número crece exponencialmente, lo que domina el tiempo de ejecución.

9. Conclusión

La Máquina de Turing que calcula la sucesión de Fibonacci presenta:

- Complejidad temporal: $\Theta(\varphi^n)$
- Complejidad espacial: $\Theta(\varphi^n)$

El crecimiento exponencial se debe a:

- La definición recursiva de Fibonacci
- La representación unaria
- El uso de una sola cinta
- La necesidad de escribir explícitamente cada símbolo generado

Este comportamiento confirma empíricamente y teóricamente que el algoritmo pertenece a la clase de crecimiento exponencial.