

Motivación

Hablar de *Computación* como un área del conocimiento científico requiere de nuestra parte incluir un vasto compendio de tópicos que crece día a día vertiginosamente. Dada esta permanente expansión de los temas relacionados con computación se ha hecho necesario el estudio formal de muchos de estos temas, este estudio profundo ha pasado a conformar en sí parte de los tópicos de mayor interés y desarrollo en Computación, estos temas conforman lo que hoy llamamos *Fundamentos de la Ciencia de Computación*. Esta área pretende responder preguntas centrales en la *Ciencia de la Computación*, estudiar *soluciones algorítmicas*, explorar y desarrollar *estructuras discretas* que mejoren esas soluciones. Proponer *modelos de computación* cada vez más complejos y ajustados a los requerimientos actuales. Estudiar la *complejidad* intrínseca de problemas algorítmicos, así como estudiar las fortalezas y debilidades de los distintos paradigmas de *lenguajes de programación*. Todos estos elementos y construcciones terminan por aparecer al formular aquellas preguntas fundamentales y en la búsqueda de sus soluciones.

Algoritmía

El primer peldaño en el estudio de los fundamentos de la Computación, es establecer lo que entenderemos por *problemas algorítmicos* y sus *soluciones algorítmicas*, este primer paso es aparentemente simple, pero no es tal. La historia nos ha demostrado que el planteamiento de un problema, requiere echar mano de un *lenguaje formal*, lo que podemos asumir por ahora como un lenguaje sin ambigüedades. Teniendo a nuestra disposición tal lenguaje, podremos establecer y proponer la solución algorítmica, por supuesto la que también representaremos en ese lenguaje formal!

Estructura Discreta

La forma de representar la información es crucial al momento de establecer una solución. Cada estructura a considerar tiene sus propiedades y características que la constituyen como la apropiada, o bien no, ante un determinado problema. Estas estructuras provienen de distintas fuentes, algunas provienen desde el contexto matemático como *grafos* y otras estructuras afines, otras en cambio provienen de tabulaciones tales como *tablas* y *diccionarios*.

Árboles Binarios

Un ejemplo significativo de estructura discreta son los árboles binarios. Estos pueden ser presentados en cualquiera de las siguientes maneras.

Definición 1 (Estructura de Datos)

$$T_A = \mu\alpha 1 + A \times \alpha \times \alpha$$

Definición 2 (Grafos) Se define un árbol binario T recursivamente con un grafo dirigido como sigue. El árbol binario T es uno de los siguientes

- Un vértice.
- Un grafo formado desde dos árboles binarios T_1 y T_2 , agregados un vértice r y aristas dirigidas desde el nuevo vértice a las raíces de cada uno de estos árboles binarios.

Seudo-Código

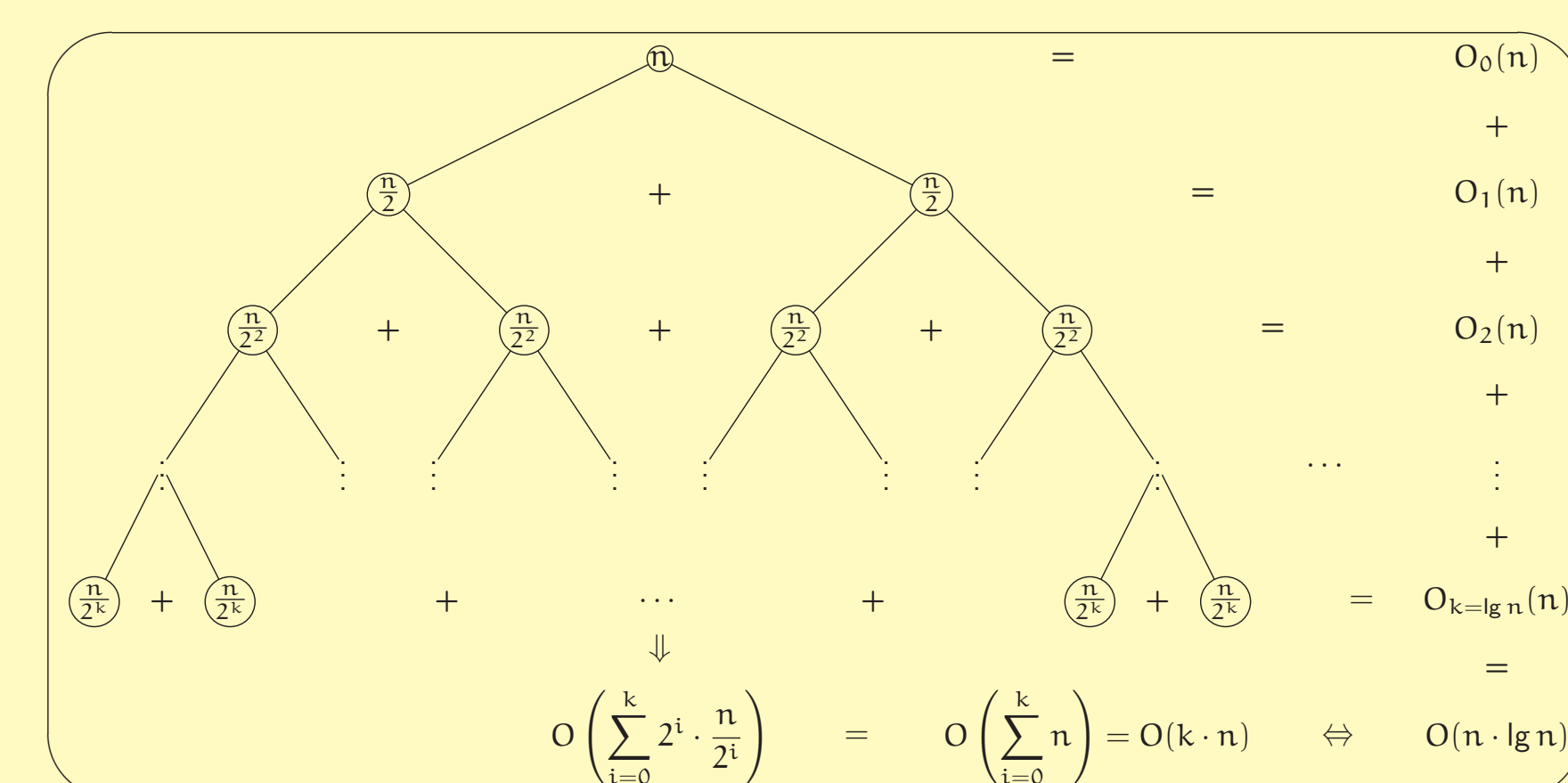
Algorithm 0.1: MERGESORT(n, X)

comment: Sort the array X of length n

```

if  $n = 2$ 
  then
    if  $X[0] > X[1]$ 
      then
         $T \leftarrow X[0]$ 
         $X[0] \leftarrow X[1]$ 
         $X[1] \leftarrow T$ 
    else if  $n > 2$ 
      then
         $m \leftarrow \lfloor n/2 \rfloor$ 
        for  $i \leftarrow 0$  to  $m-1$ 
          do  $A[i] \leftarrow X[i]$ 
        for  $i \leftarrow m$  to  $n-1$ 
          do  $B[i] \leftarrow X[i]$ 
        comment: Now sort the subarrays  $A$  and  $B$ 
        MERGESORT( $m, A$ )
        MERGESORT( $n-m, B$ )
      then
         $i \leftarrow 0$ 
         $j \leftarrow 0$ 
        for  $k \leftarrow 0$  to  $n-1$ 
          if  $A[i] \leq B[j]$ 
            then
               $X[k] \leftarrow A[i]$ 
               $i \leftarrow i+1$ 
            else
               $X[k] \leftarrow B[j]$ 
               $j \leftarrow j+1$ 
  
```

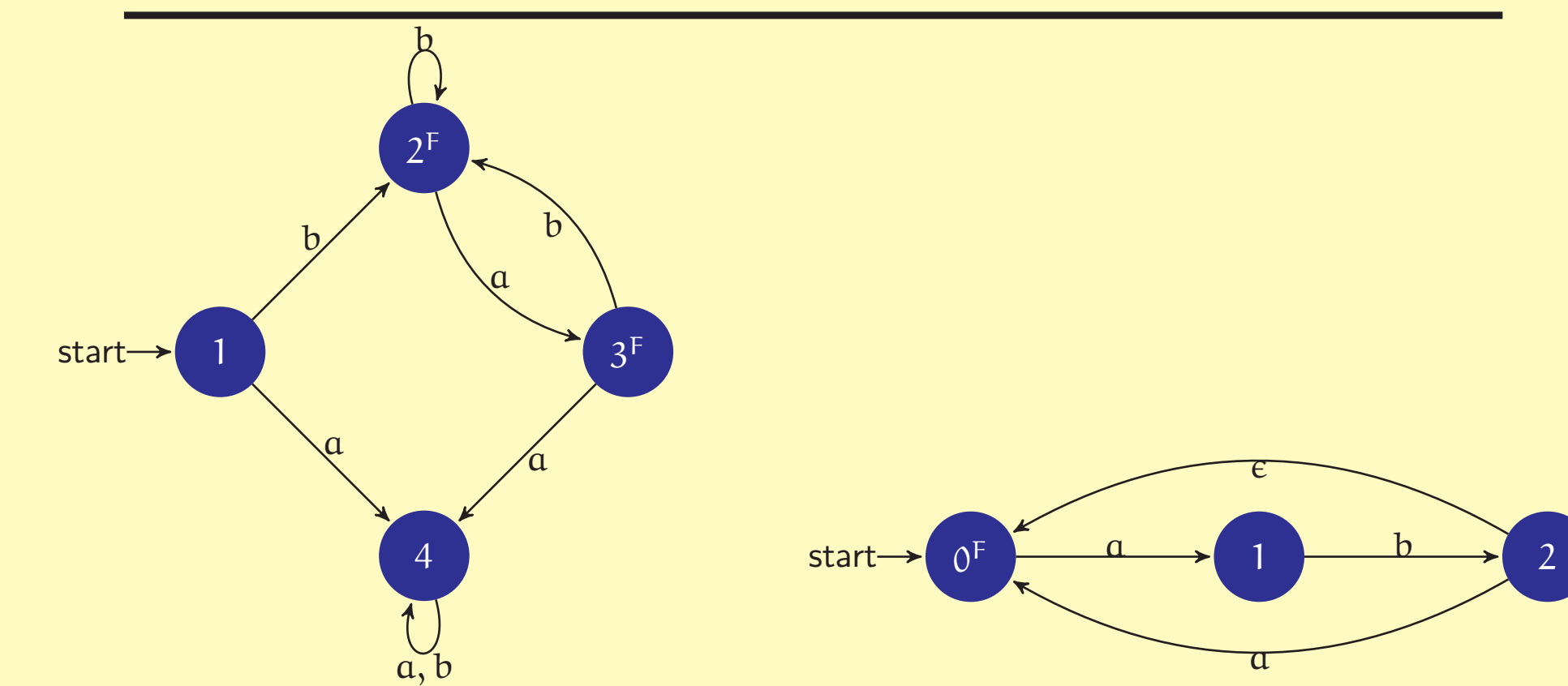
Complejidad



Lenguajes Formales, Máquinas Abstractas y Computabilidad

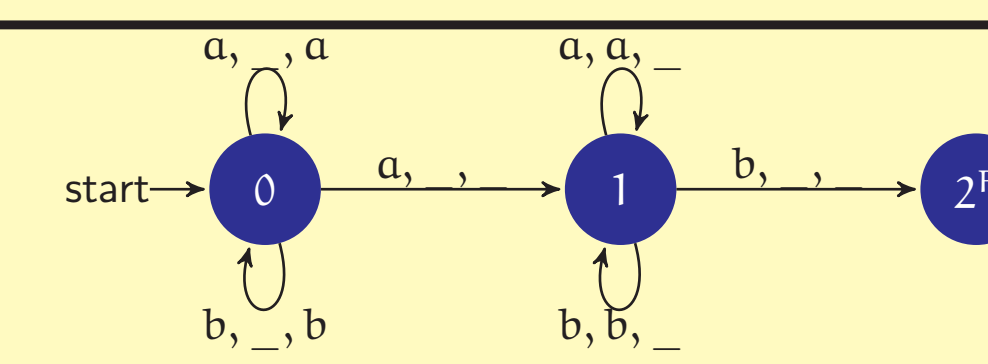
Un tema no trivial es establecer *modelos computacionales* la equivalencia de todos estos. Uno de las aristas de es que nos permitan capturar la noción de lo que es *computable*. A lo largo de la historia de la computación han aparecido diversos enfoques y propuestas de modelos. En una primera instancia, estos modelos no tenían relación alguna, con el correr del tiempo se establecieron

Lenguajes Regulares



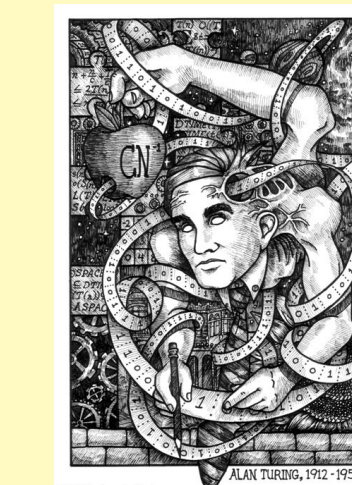
Las ilustraciones corresponden a dos versiones de autómatas finitos, *determinístico* y *no-determinístico* respectivamente. Aquellos lenguajes caracterizados por alguno de estos dispositivos se dice *regular*. Esta familia de lenguajes es bastante *simple* y constituye una familia de lenguajes que tiene un comportamiento *fuertemente tratable*.

Lenguajes Libres de Contexto



La ilustración muestra un **AP** *autómata finito con pila*. Esta clase de lenguajes corresponde a aquellos lenguajes caracterizados por estos verificadores. Esta segunda clase de lenguajes formales, tiene una fuerte componente práctica, muchos de los *lenguajes de programación* existentes corresponden esencialmente a este tipo de lenguajes

Lenguajes Recursivos



La ilustración representa una *Máquina de Turing*, la que corresponde el dispositivo abstracto bastante conocido y que captura la noción de lo que es *algorítmicamente calculable*. Esta clase de lenguajes representa aquellos lenguajes que tienen una representación finita y excluyen todos aquellos lenguajes *computacionalmente intratables*.

Asignaturas

Las siguientes asignaturas se concentran alrededor de estos temas: **Ideas Teóricas en Ciencias de la Computación, Matemáticas Discretas, Principios de Programación, Fundamentos de Lenguajes de Programación, Autómatas y Lenguajes Formales, Diseño y Análisis de Algoritmos**