Lógica y Computabilidad

2do cuatrimestre 2020 - **A DISTANCIA**

Departamento de Computación - FCEyN - UBA

Computabilidad - clase 1

Introducción, máquinas de Turing, funciones parciales, funciones Turing computables, ejemplos

Orígenes

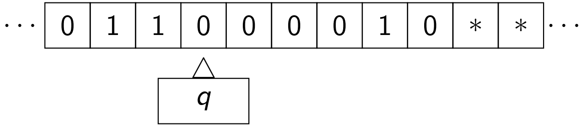
- fines del siglo XIX y principios del siglo XX: interés por los fundamentos de la matemática
- dos grandes búsquedas (Hilbert)
  1. completitud de la aritmética
     - se buscaba un sistema axiomático que capturara todas las verdades de la aritmética
     - Gödel (1931): cualquier sistema axiomático suficientemente poderoso es incompleto o inconsistente
  2. el problema de la decisión (*Entscheidungsproblem*)
     - se buscaba un procedimiento efectivo para decidir si cualquier fórmula de primer orden era válida o no
     - Turing (1936): no existe tal procedimiento efectivo

David Hilbert

Kurt Gödel

Alan Turing

Máquinas de Turing



Se compone de :

- una cinta
  - dividida en celdas
  - infinita en ambas direcciones
  - cada celda contiene un símbolo de un alfabeto dado Σ.
    - $* \in \Sigma$
    - $L, R \notin \Sigma$

    $*$ representa el blanco en una celda
    L y R son símbolos reservados (representarán acciones que puede realizar la cabeza)
- una cabeza
  - lee y escribe un símbolo a la vez
  - se mueve una posición a la izquierda o una posición a la derecha
- una tabla finita de instrucciones
  - dice qué hacer en cada paso

Tabla de instrucciones

- Σ es el alfabeto. $L, R \notin \Sigma$, $* \in \Sigma$.
- Q es el conjunto finito de estados
- $A = \Sigma \cup \{L, R\}$ es el conjunto de acciones
  - un símbolo $s \in \Sigma$ se interpreta como "escribir s en la posición actual"
  - L se interpreta como "mover la cabeza una posición hacia la izquierda"
  - R se interpreta como "mover la cabeza una posición hacia la derecha"

Una tabla de instrucciones T es un subconjunto (finito) de

$$Q \times \Sigma \times A \times Q$$

La tupla

$$(q, s, a, q') \in T$$

se interpreta como

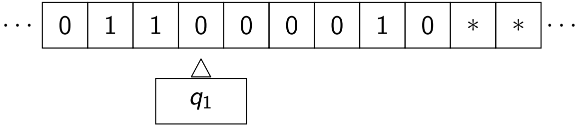
> *Si la máquina está en el estado q leyendo en la cinta el símbolo s, entonces realiza la acción a y pasa al estado q'*

Ejemplo de ejecución de una instrucción

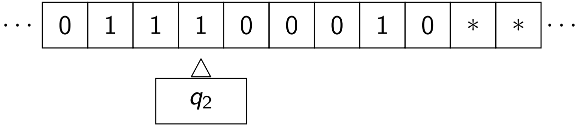
Supongamos un alfabeto $\Sigma = \{0, 1\}$.
Una máquina con esta tabla de instrucciones:

$$\{\ (q_1, 0, 1, q_2) \quad , \quad (q_2, 1, R, q_1)\ \}$$

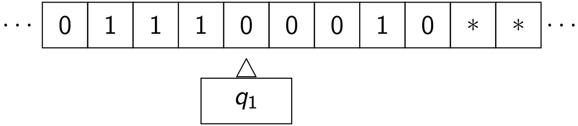
Si empieza en esta configuración



pasa a



y luego a



Definición de máquina de Turing

Una máquina de Turing es una tupla

$$(\Sigma, Q, T, q_0, q_f)$$

donde

- Σ (finito) es el conjunto símbolos ($L, R \notin \Sigma, * \in \Sigma$)
- Q (finito) es el conjunto de estados
  - tiene dos estados distinguidos:
    - $q_0 \in Q$ es el estado inicial
    - $q_f \in Q$ es el estado final
- $T \subseteq Q \times \Sigma \times \Sigma \cup \{L, R\} \times Q$ es la tabla de instrucciones
  - va a ser finita porque Σ y Q lo son

- cuando no hay restricciones sobre T decimos que $\mathcal{M}$ es una máquina de Turing no determinística
- cuando no hay dos instrucciones en T que empiezan con las mismas primeras dos coordenadas, decimos que $\mathcal{M}$ es una máquina de Turing determinística

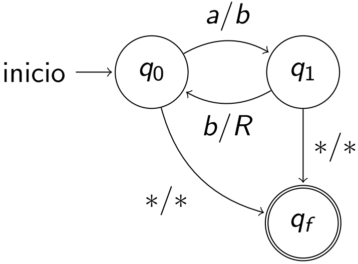
Ejemplo

Sea $\mathcal{M} = (\Sigma, Q, T, q_0, q_f)$ con

- $\Sigma = \{*, a, b\}$
- $Q = \{q_0, q_1, q_f\}$
- tabla de instrucciones

$$T = \begin{array}{cccc} q_0 & a & b & q_1 \\ q_1 & b & R & q_0 \\ q_0 & * & * & q_f \\ q_1 & * & * & q_f \end{array}$$

Visto como autómata



- si empieza en $\begin{array}{cccccc} * & a & a & a & a & * \\ & q_0 & & & & \end{array}$ termina en $\begin{array}{cccccc} * & b & b & b & b & * \\ & & & & & q_f \end{array}$
- si empieza en $\begin{array}{cccccc} * & a & a & b & a & * \\ & q_0 & & & & \end{array}$ termina en $\begin{array}{cccccc} * & b & b & b & a & * \\ & & & q_0 & & \end{array}$
- si empieza en $\begin{array}{cccccc} * & a & a & b & a & * \\ q_0 & & & & & \end{array}$ termina en $\begin{array}{cccccc} * & a & a & b & a & * \\ q_f & & & & & \end{array}$

Representación de números y tuplas

Fijamos $\Sigma = \{*, 1\}$.

- representaremos a los números naturales en unario (con palotes).
  - el número $x \in \mathbb{N}$ se representa como

$$\overline{x} = \underbrace{1 \ldots 1}_{x+1}$$

- representamos a las tuplas $(x_1, \ldots, x_n)$ como lista de (representaciones de) los x_i separados por blanco
  - la tupla $(x_1, \ldots, x_n)$ se representa como

$$*\overline{x_1} * \overline{x_2} * \cdots * \overline{x_n}*$$

Por ejemplo,

- el número 0 se representa como 1
- el número 3 se representa como 1111
- la tupla $(1, 2)$ se representa como $*11 * 111*$
- la tupla $(0, 0, 1)$ se representa como $*1 * 1 * 11*$

Funciones parciales

Siempre vamos a trabajar con funciones $f : \mathbb{N}^n \to \mathbb{N}$.

Pero van a ser funciones parciales. Una función parcial f es una función que puede estar indefinida para algunos (tal vez ninguno; tal vez todos) sus argumentos.

- notamos $f(x_1, \ldots, x_n) \downarrow$ cuando f está definida para $x_1, \ldots, x_n$. En este caso $f(x_1, \ldots, x_n)$ es un número natural.
- notamos $f(x_1, \ldots, x_n) \uparrow$ cuando f está indefinida para $x_1, \ldots, x_n$

El conjunto de argumentos para los que f está definida se llama dominio de f, notado $\mathrm{dom}(f)$.

$$\mathrm{dom}(f) = \{(x_1, \ldots, x_n) : f(x_1, \ldots, x_n) \downarrow\}$$

f es total si $\mathrm{dom}(f) = \mathbb{N}^n$.

Cómputo de funciones parciales en máquinas de Turing

Una función parcial $f : \mathbb{N}^n \to \mathbb{N}$ es Turing computable si existe una máquina de Turing determinística $\mathcal{M} = (\Sigma, Q, T, q_0, q_f)$ con $\Sigma = \{*, 1\}$ tal que cuando empieza en la configuración inicial

$\cdots$	$*$	$*$	$*$	$\overline{x_1}$	$*$	$\overline{x_2}$	$*$	$\cdots$	$*$	$\overline{x_n}$	$*$	$*$	$*$	$\cdots$
											$\triangle$			
											q_0			

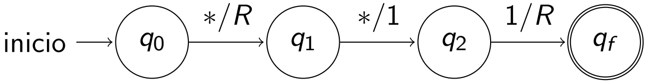
(con los enteros x_i representados en unario y nada más en la entrada salvo la representación de la entrada):

- si $f(x_1, \ldots, x_n) \downarrow$ entonces siguiendo sus instrucciones en T llega a una configuración final de la forma

$\cdots$	$*$	$\overline{f(x_1, \ldots, x_n)}$	$*$	$\cdots$
			$\triangle$	
			q_f	

  (quizá algo más en la cinta)
- si $f(x_1, \ldots, x_n) \uparrow$ entonces nunca termina en el estado q_f.

Cómputo de la función $f(x) = 0$



Cómputo de la función $f(x) = x + 1$

$$\text{inicio} \longrightarrow q_0 \xrightarrow{*/1} q_1 \xrightarrow{1/R} q_f$$

Cómputo de la función $f(x) = 2x$

Idea: por cada 1 que borro de la entrada, pongo 11 bien a la derecha. Repito esto hasta que quede solo un 1 en la entrada. Ahí pongo un 1 más a la derecha.

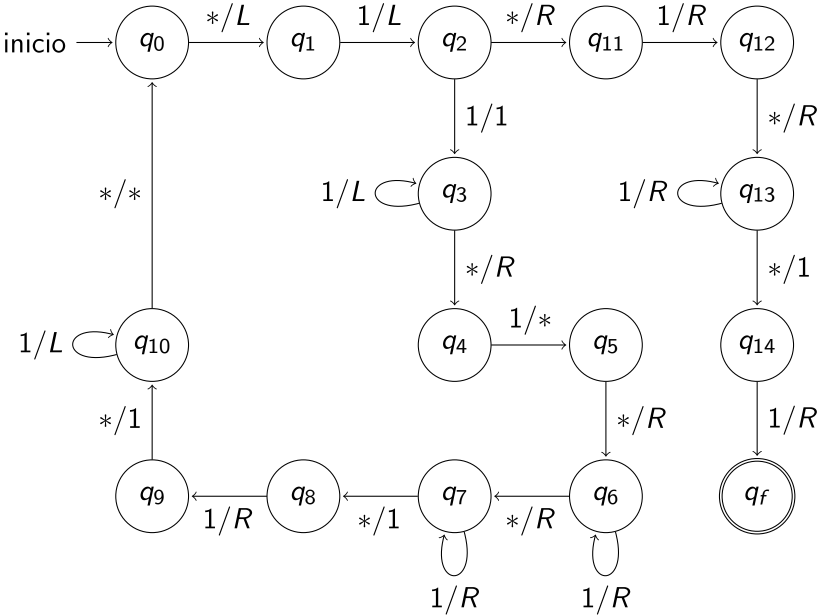
Ejemplo: entrada $= 2$

1. $* \; * \; * \; * \; * \; 1 \; 1 \; 1 \; * \; * \; * \; * \; * \; * \; * \; * \; *$
2. $* \; * \; * \; * \; * \; * \; 1 \; 1 \; * \; 1 \; 1 \; * \; * \; * \; * \; * \; *$
3. $* \; * \; * \; * \; * \; * \; * \; 1 \; * \; 1 \; 1 \; 1 \; 1 \; * \; * \; * \; *$
4. $* \; * \; * \; * \; * \; * \; * \; 1 \; * \; 1 \; 1 \; 1 \; 1 \; 1 \; * \; * \; *$

Invariante: a lo largo de cada iteración, la cinta está así:
$***\underbrace{1\ldots1}_{n}*\underbrace{1\ldots\ldots1}_{2m}***$ para algún $n > 0, m \geq 0$, $n + m - 1 =$ entrada

1. si $n = 1$ entonces pongo un 1 más a la derecha y termina en q_f con $***1*\underbrace{1\ldots\ldots1}_{2m+1}***$
2. si $n > 1$ transformo la cinta en $***\underbrace{1\ldots1}_{n-1}*\underbrace{1\ldots\ldots1}_{2(m+1)}***$ y vuelvo al paso 1

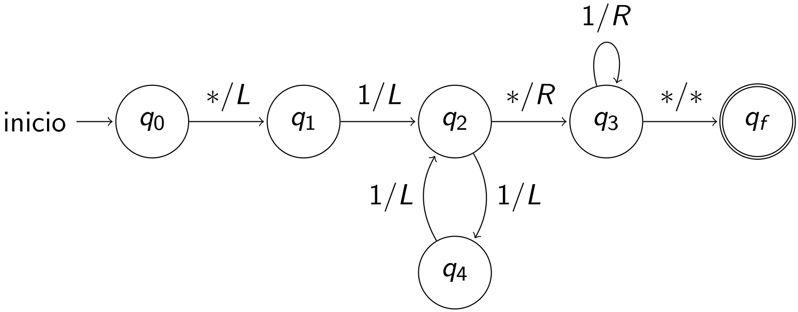
Cómputo de la función $f(x) = 2x$



Cómputo de una función parcial

Supongamos

$$f(x) = \begin{cases} x & \text{si } x \text{ es par} \\ \uparrow & \text{si no} \end{cases}$$



Poder de cómputo

Teorema

Sea $f : \mathbb{N}^m \to \mathbb{N}$ una función parcial. Son equivalentes:

1. *f es computable en Java*
2. *f es computable en C*
3. *f es computable en Haskell*
4. *f es Turing computable*

No es importante

- qué base usamos para representar a los números
  - usamos representación unaria ($\Sigma = \{*, 1\}$)
  - pero podríamos haber elegido la binaria ($\Sigma = \{*, 0, 1\}$)
  - o base 10 ($\Sigma = \{*, 0, 1, 2, \ldots, 9\}$)
- si permitimos que al terminar la cinta tenga otras cosas escritas además de la salida o solo contenga la salida
- si usamos esta variante de arquitectura:
  - una cinta de entrada (solo de lectura)
  - una cinta de salida (solo de escritura)
  - una o varias cintas de trabajo, de lectura/escritura

¡Siempre computamos la misma clase de funciones!