

¿ MÁQUINA DE TURING
AUTORRERENCIAL: UNA
IMPOSIBILIDAD ?

Andrés Sicard Ramírez

Departamento de Ciencias Básicas

Universidad EAFIT

Medellín, Colombia, S.A.

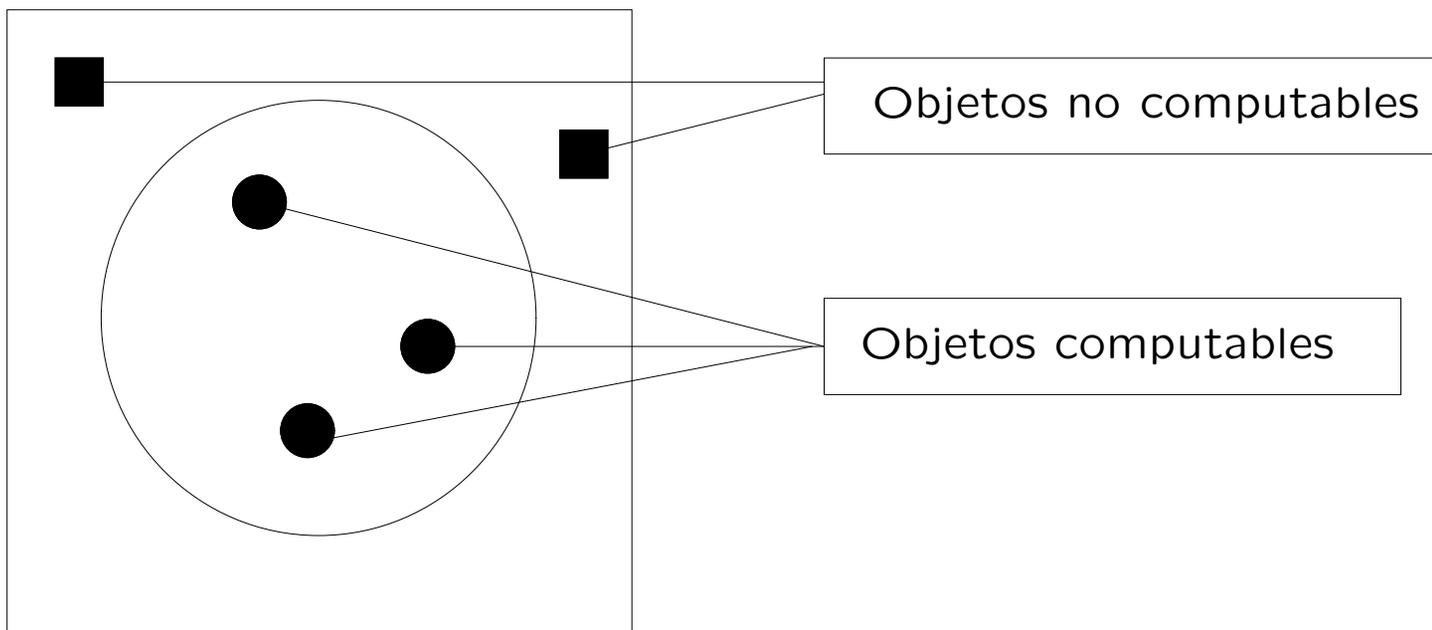
email: asicard@eafit.edu.co

COMPUTABILIDAD

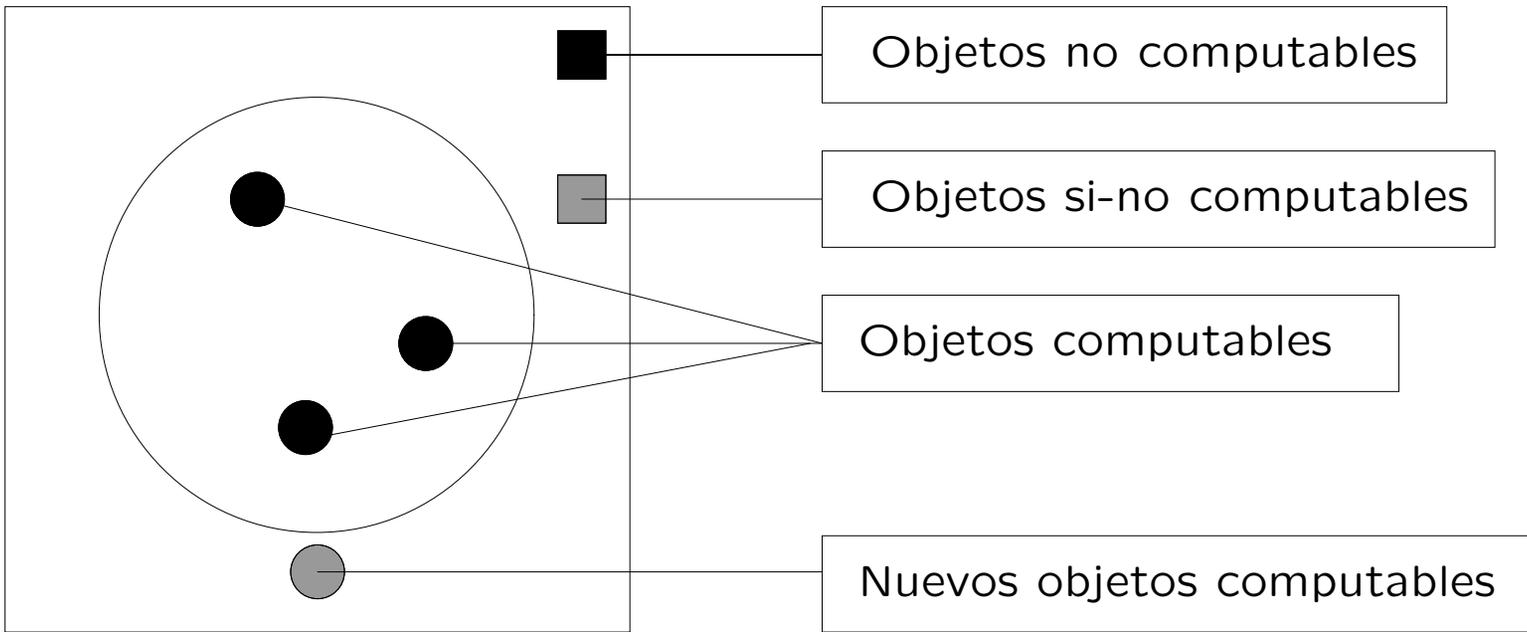
Una **computación** es un proceso por el cual se procede a partir de unos objetos iniciales, llamados **entradas**; de acuerdo a un conjunto fijo de reglas, llamadas el **programa-procedimiento-algoritmo**; a través de una serie de pasos precisos y definitivos; y al finalizar éstos, se llega a un resultado, llamado la **salida**.

El concepto de **computabilidad** concierne con aquellos objetos, los cuales pueden ser especificados-determinados-explicados **en principio** por computaciones.

OBJETOS COMPUTABLES



¿ NUEVOS OBJETOS COMPUTABLES ?



MÁQUINA DE TURING DETERMINÍSTICA

$\mathcal{M} = \langle Q, \Sigma, M, \delta \rangle$ donde:

$Q = \{q_0, q_1, q_2, \dots, q_n\}$: Conjunto finito de estados de la máquina.

$\Sigma = \{s_0, s_1, s_2, \dots, s_m\}$: Alfabeto. Conjunto finito de símbolos de entrada - salida. Por convención que $s_0 = \square$ (símbolo vacío).

$M = \{L, R, N\}$: Conjunto de movimientos (L : izquierda, R : derecha, N : no movimiento).

δ : Función definida de un subconjunto $K \times \Sigma$ en $\Sigma \times M \times K$. δ también puede ser definida como un conjunto *finito* de instrucciones $\delta = \{i_0, i_1, i_2, \dots, i_p\}$ donde cada i_j es una quintupla de la forma: $q_m \ s_m \ s_n \ m \ q_n$.

EJEMPLO MÁQUINA DE TURING

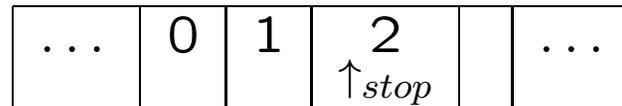
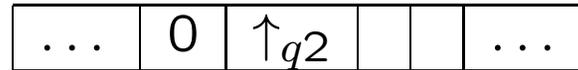
Instrucciones

$i_1: q_1 \square 0 R q_2$

$i_2: q_2 \square 1 R q_3$

$i_3: q_3 \square 2 N stop$

Simulación



CODIFICACIÓN MÁQUINAS DE TURING

De acuerdo a la definición formal de una máquina de Turing, se tiene el siguiente “esquema” para una instrucción

$q_i s_j s_k m q_l$.

Se reemplaza cada instrucción de acuerdo a:

q_i : Se reemplaza por D seguida de A es i veces

s_j : Se reemplaza por D seguida de C es j veces

Las instrucciones se reescriben utilizando este código y se separan por un símbolo “;”. Cuando se utiliza este sistema de codificación para describir las instrucciones, se dice que la máquina está representada en su **descripción estándar**.

EJEMPLO CODIFICACIÓN INSTRUCCIONES

Instrucciones

Codificación

$i_1: q_1 \square 0 R q_2$

D A D D C R D A A

$i_2: q_2 \square 1 R q_3$

D A A D D C C R D A A A

$i_3: q_3 \square 2 N stop$

D A A A D D C C C N D A A A A

MÁQUINA UNIVERSAL DE TURING

Máquina de Turing
(MT)

Entrada

Máquina Universal
de Turing (MUT)

Salida

EJECUCIÓN MT POR LA MUT (segunda parte)

Final ejecución

Descripción de la MT que fue ejecutada por la MUT

Descripción primera instrucción

...	D	A	D	D	C	R	D	A	A	;	...
-----	---	---	---	---	---	---	---	---	---	---	-----

Descripción de la MT que fue ejecutada por la MUT (cont.)

Descripción última instrucción

...	D	A	A	A	D	D	C	C	C	N	D	A	A	A	A	;
-----	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---

Resultado de ejecutar la MT

	D	C	D	C	C	D	C	C	C	...
--	---	---	---	---	---	---	---	---	---	-----

MÁQUINA DE TURING AUTORREFERENCIAL (MTAR)



EJECUCIÓN MTAR POR LA MUT (primera parte)

Inicio ejecución

Descripción de la MTAR que va a ser ejecutada por la MUT

Descripción primera instrucción

...	D	A	D	D	C	R	D	A	A	;	...
-----	---	---	---	---	---	---	---	---	---	---	-----

Descripción de la MTAR que va a ser ejecutada por la MUT (cont.)

Descripción última instrucción

...	D	A	A	A	D	D	C	C	C	N	D	A	A	A	A	;
-----	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---

Área de impresión del comportamiento de la MTAR

																	...
--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	-----

EJECUCIÓN MTAR POR LA MUT (segunda parte)

Final ejecución

Descripción de la MTAR que fue ejecutada por la MUT

Descripción primera instrucción

...	D	A	D	D	C	L <i>cambia R por L</i>	D	A	A	;	...
-----	---	---	---	---	---	------------------------------	---	---	---	---	-----

Descripción de la MTAR que fue ejecutada por la MUT (cont.)

Descripción última instrucción

...	D	A	A	A	D	D	C	C	C	N	D	A	A	A	A	;
-----	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---

Resultado de ejecutar la MTAR

	D	C	D	C	C	D	C	C	C	...
--	---	---	---	---	---	---	---	---	---	-----

M-FUNCIONES

El concepto de m-función, se interpreta como una máquina de Turing que permite el uso de parámetros (estados y símbolos) en el momento de su definición, los cuales son instanciados en el momento de su invocación. El uso y la combinación de estas máquinas de Turing parametrizadas o m-funciones simplifica la descripción de las máquinas que las usan.

CÓDIGO M-FUNCIÓN cambiarInstrucción

```
proc cambiarInstruccion(instruccionOriginal, instruccionFinal); ≡
  (1)  guardarPosicionInicial();
  (2)  imprimirInstruccion(instruccionOriginal);
  (3)  marcarInstruccion(o);
  for i := 1 to totalIntrucciones do
    (4)  marcarInstruccion(i);
    (5)  compararInstrucciones(i, o);
    if instruccioni = instruccionOriginal
      then
        (6)  imprimirInstrucciones(instruccionFinal);
        (7)  marcarInstruccion(f);
        (8)  reemplazarInstruccion(i, f);
        (9)  eliminarInstruccion(f);
        (10) eliminarInstruccion(o);
        (11) eliminarMarca(i);
        (12) restablecerPosicionInicial();
      else (13) eliminarMarca(i);
    end
  end
end
(14) eliminarInstruccion(o); (15) restablecerPosicionInicial();
end
```

¿ PORQUÉ ES IMPOSIBLE CONSTRUIR UNA MTAR ?

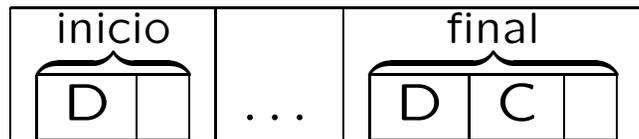
La razón de no constructibilidad de la MTAR es que la MTAR y la MUT manejan por definición diferentes códigos para representar los mismos símbolos.

La MTAR debe desplazarse por la cinta en el área asignada a su descripción, para modificar los símbolos en los cuales está codificada dicha descripción, pero estos símbolos están codificados en un código diferente al utilizado por la MTAR, luego la MTAR es incapaz de modificar su propia descripción.

Se puede afirmar que es una incapacidad **sintáctica**, la MTAR no es capaz de “hablar” el lenguaje en que ella es codificada para ser ejecutada por la MUT.

CAMBIO $q_x \square 0 N q_{x+1}$ **POR** $q_x 0 0 N q_{x+1}$

cambio:



instrucción cambio:

$q_1 \square C N q_2$

codificación símbolos:

\square D
 0 DC
 1 DCC

A $DCCC$
 C $DCCCC$
 L $DCCCCC$
 R $DCCCCC$
 L $DCCCCCCC$
 $;$ $DCCCCCCCC$

codificación instrucción:
 $DADDCCCCNDAA$

La MT opera con el símbolo "C" pero la MUT opera con la secuencia de símbolos "DCCCC".

¿ POSIBLES SOLUCIONES ?
¿ NUEVOS PROBLEMAS ?

1. Código único
2. Meta-meta-código común
3. Computación cuántica
4. Máquinas paraconsistentes
5. Super-Turing teorías (computabilidad continua)