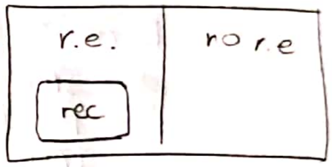
DECIDIBILIDAD

- PROBLEMA(x)   X entradas,   Y soluciones
  $R : X \times Y \longrightarrow \{0,1\}$ $R(x,y) = 1$, not $R(x,y)$ $R(x,y) = 0$ not $\neg R(x,y)$
- ALG resuelve PROBLEMA(x) si $ALG(x) = y \in Y$ tq $R(x,y) = 1$
- L (PROBLEMA(x)) $= \{x \in A^* .$ PROBLEMA(x) = 'SI' $\}$ (en decisión)
  $\Leftarrow$ L sobre A genera problema de decisión: PROBLEMA(x) = $[\![x \in L]\!]$

Problema Decisión + Codificación ≡ Problema Computacional 'SI' 'NO' ≡ Lenguaje

- Problema con lenguaje
  - recursivo ⟶ decidible o calculable
  - recursivamente enumerable ⟶ semidecidible o parc. calculable
  - no recursivo ⟶ indecidible o no calculable

recursivo $\Leftrightarrow \exists$ MT siempre termina   rec. enum $\Leftrightarrow \exists$ MT



- Problema contrario

  PROBLEMA(x) = 'SI' $\Leftrightarrow$ C-PROBLEMA(x) = 'NO'

  L (C-PROBLEMA) = $\overline{\text{L(PROBLEMA)}}$

- Palabras ↔ números : $A = \{a_1, \dots, a_n\}$

  $w = a_{i_k} \cdots a_{i_1} \rightsquigarrow Z(w) = \sum_{j=1}^{k} i_j n^{j-1}$, $Z(\varepsilon) = 0$. $Z(w) \in \mathbb{N}$

  $m \in \mathbb{N} \rightsquigarrow C(m)$ $(\equiv w_m)$ $\longrightarrow \begin{cases} m = 0 \Rightarrow C(m) = \varepsilon \\ m > 0, \Rightarrow C(m) = C(p)\, a_i, \text{ con} \end{cases}$

- suficiente trabajar en un alfabeto
  $A = \{0,1\}$ (normalmente)

  $i = \begin{cases} m \,\%\, n & \text{si } n \nmid m \\ n & \text{si } n \mid m \end{cases}$

  $p = \begin{cases} m \,//\, n & \text{si } n \nmid m \\ m \,//\, n - 1 & \text{si } n \mid m \end{cases}$

  orden total

  $u_1 \leq u_2 \Longleftrightarrow (1) \vee (2) \vee (3)$

  (1) $u_1 = u_2$
  (2) $|u_1| < |u_2|$
  (3) $|u_1| = |u_2|$ y u_1 precede a u_2 alfabéticamente

- Codificación MT. $A = \{0,1\}$

  Estados $\{q_1, q_2, \dots, q_k\}$ q_1 inicial, q_2 final
  $B = \{a_1, a_2, \dots, a_m\}$ $a_1 = 0$, $a_2 = 1$, $a_3 = \#$
  $u(M) = 1$ si $M = I$, $u(M) = 2$ si $M = D$
  codificación: cada trans: $\delta(q_i, a_j) = (q_k, a_l, M)$ cod. $0^i 1 0^j 1 0^k 1 0^l 1 0^{u(M)}$
  se separan por 11

  $w = R(M) \rightsquigarrow Z(w) = Z(M)$

  $T(n)$ = MT tq $Z(MT) = n$ ó 'Nula' si no hay MT asoc. a número n.
  $T(w)$ = MT tq $R(MT) = w$

- Codificación MT + entrada $R(M, w) = R(M)\, 111\, w$
  $(w \in \{0,1\}^*)$

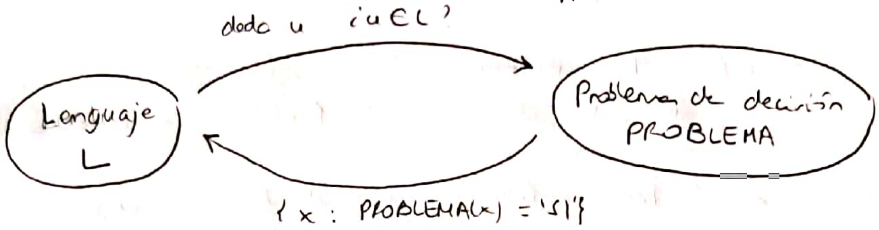
Ld no r.e

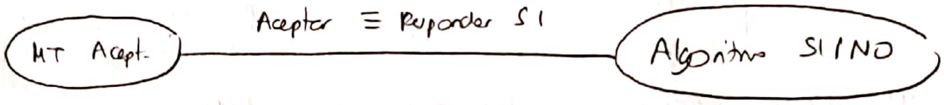
- Lenguaje de diagonalización NO RE

$w \in \{0,1\}^*$ $w \in L_d \iff$ MT cuya codificación es w no acepta w
(si w no es MT correcta, representa MT con un estado y ninguna trans. (siempre rechaza))

- Problema de Diagonalización: versión de problema de decisión de L_d

DIAGONAL(M) = ⟦ M no acepta cuando se lee a ella misma ⟧
$\Updownarrow$
$w_M \in L_d$





L re ≡ PROBLEMA semidecidible
- ∃MT que acepta las palabras del lenguaje
  para las que no están en el lenguaje puede rechazar o ciclar
- ∃ALG que responde correctamente a las entradas cuya salida es SI
  para las entradas de NO el algoritmo puede decir NO o ciclar

L rec ≡ PROBLEMA decidible
- ∃MT que acepta las palabras del lenguaje
  para las que no rechaza (nunca cicla)
- ∃ALG que resp. corr. - - - (ídem)
  para las que NO dice NO (nunca cicla)

$\Rightarrow$ → L recursivo $\Rightarrow$ $\overline{L}$ recursivo
→ L, $\overline{L}$ rec. enum. $\Rightarrow$ L recursivo
(cont.) → L no rec. enum. $\Rightarrow$ $\overline{L}$ no recursivo (puede ser r.e.)

ej/ Ld no re $\overline{Ld}$ es re

Lu re, no rec

- Lenguaje universal

$L_u = \{ v \in \{0,1\}^* \mid v = R(M,w)$ con
M Con alfabeto de entrada $\{0,1\}$
$w \in \{0,1\}^*$
M acepta w $\}$

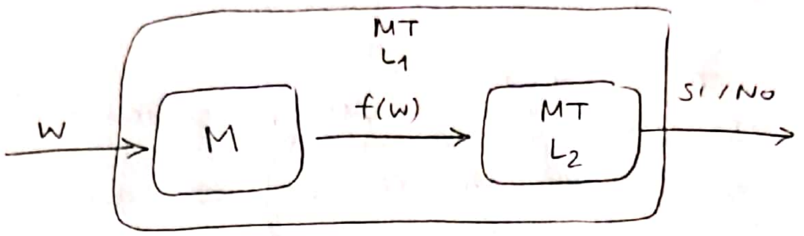
- Problema Universal

UNIVERSAL(M, w) = ⟦ M acepta w ⟧

not// $A \propto B$. A se reduce a B

- Reducción (lenguajes)

$L_1 \subseteq A^*$, $L_2 \subseteq B^*$, L_1 se reduce a L_2 si $\exists$ algoritmo M (MT) que siempre para y calcula una $f: A^* \longrightarrow B^*$ tq $\forall w \in A^*$,
$w \in L_1 \iff f(w) \in L_2$

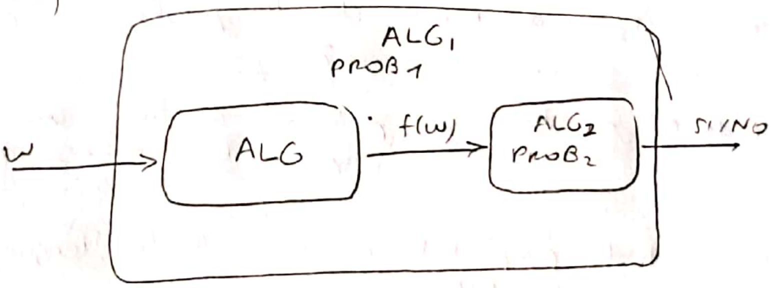


$\Rightarrow$ L_2 recursivo $\Rightarrow$ L_1 recursivo
L_2 rec. enumerable $\Rightarrow$ L_1 rec. enumerable

(usamos contrarrecíproco)
L_1 no r $\Rightarrow$ L_2 no r
L_1 no re $\Rightarrow$ L_2 no re

- Reducción (problemas)

PROBLEMA$_i$ de decisión, $i=1,2$ — $PROB_1$ se reduce a $PROB_2$ si $\exists$ ALG(w) que siempre para y realcula una función f(w) tq $\forall w$ entrada a $PROB_1$, tenemos que $PROB_2$ produce la misma respuesta para f(w) = ALG(w)



una reducción de $PROB_1$ a $PROB_2$ es un ALG(x) de las entradas de $PROB_1$ a las de $PROB_2$ tq

$PROB_1(x) = PROB_2(ALG(x))$

$\Rightarrow$ $PROB_1$ ~~indecidible~~ $\Rightarrow$ $PROB_2$ ~~indecidible~~
$PROB_1$ ~~no semidecidible~~ $\Rightarrow$ $PROB_2$ no semidecidible ⟵ ⊕

ALG_2 hace $PROB_2$ semidecidible $\Rightarrow$ ALG_1 hace $PROB_1$ semidecidible
ALG_2 hace $PROB_2$ decidible $\Rightarrow$ ALG_1 hace $PROB_1$ decidible

$PROB_1$ no semidecidible $\Rightarrow$ $PROB_2$ no semidecidible
$PROB_1$ no decidible (indecidible) $\Rightarrow$ $PROB_2$ no decidible

L_{ne} re no rec

- $L_e = \{ w \in \{0,1\}^* \mid$ MT M sobre $\{0,1\}$ cuya codificación es w no acepta ninguna palabra $(L(M) = \phi) \}$

- $L_{ne} = \{ w \in \{0,1\}^* \mid$ MT M sobre $\{0,1\}$ con codificación w acepta alguna palabra $(L(M) \neq \phi) \}$

- VACIO(M) $= [\![L(M) = \phi]\!]$ - C-VACIO(M) $= [\![L(M) \neq \phi]\!]$

$\Rightarrow$ - L_{ne} rec. enum, y no rec.
- L_e no rec. enum (VACIO(M) no es semidecidible)

dem/ $\left(\begin{array}{l} L_{ne} \text{ re} \\ L_e \text{ re} \\ (L_e = \overline{L_{ne}}) \end{array} \Rightarrow L_{ne} \text{ re } \lightning \right) \Rightarrow L_e$ no re

- Propiedades de las re

Preguntan sobre el lenguaje que acepta una MT, no sobre la MT en sí

✓ L(MT) finito? ✗ MT más de 10 estados?

Pregunta sobre la salida de un programa, no sobre la forma del programa.

- trivial si $\forall$ MT su lenguaje verifica (o no) una propiedad
  - ¿MT tiene siempre o menos de 10 palabras o más de 8? sí
  - ¿acepta 01 y cicla para 01? no

no trivial. L de MT es finito — podemos construir una MT que acepte un L infinito y otra que sea finito

- teorema (de Rice) toda propiedad no trivial de los lenguajes re es indecidible

"no puedo hacer un algoritmo que resuelva una pregunta sobre lo que calcula un programa"
(lenguaje)

Dado M ¿es cierto que L(M) cumple P?
(propiedad)

MANUAL

NO DECIDIBLE

- UNIVERSAL (M, w) = ⟦ M acepta w ⟧ ⇝ L_u (lenguaje universal)
- PARADA (M, w) = ⟦ M para con entrada w ⟧
- C-DIAGONAL (M) = ⟦ M acepta su codificación <M> = w_M ⟧

NO SEMIDECIDIBLE: dem. que su complementario es semidecidible y no decid.

- DIAGONAL(M) = ⟦ M no acepta su codificación <M> ⟧ ⇝ L_d (leng. diagonal)
- C-UNIVERSAL (M, w) = ⟦ M no acepta w ⟧ ⇝ $\overline{L_u}$
- EMPTY (M) = ⟦ $L(M) = \emptyset$ ⟧ ⇝ L_e

---

① Dados M_1, M_2, determinar si $L(M_1) \subseteq L(M_2)$ $\boxed{\text{NO SEMI}}$

P1 : dada M, determinar si $L(M) = \emptyset$ NO SEMI

P2 : dadas M_1, M_2, determinar si $L(M_1) \subseteq L(M_2)$

$P_1 \propto P_2 \Rightarrow$ P2 NO SEMI

sea $\tilde{M}$ una MT tq $L(\tilde{M}) = \emptyset$, entonces

$P_2(M, \tilde{M}) = $ SÍ

$P_1(M) = $ SÍ

$L(M) = \emptyset \Rightarrow L(M) \subseteq L(\tilde{M})$

$L(M) \neq \emptyset \Rightarrow L(M) \not\subseteq L(\tilde{M})$

$P_1(M) = $ NO

$P_2(M, \tilde{M}) = $ NO

Así que el ALG para la reducción es $ALG(M) = (M, \tilde{M})$

y la MT $\tilde{M}$ puede construirse sencillamente como:

$\tilde{M} = \{ \{q_0\}, \{0,1\}, \{0,1,\#\}, \delta, q_0, \#, \emptyset)$

donde δ no tiene transiciones.

② Dado APND y palabra de entrada, determinar si el autómata acepta la palabra. $\boxed{\text{DECID}}$

P autómata con pila, u palabra

↓

G gramática libre de contexto que genera lenguaje que acepta P

↓

G' mismas palabras que G salvo la palabra vacía (si G genera ε)

G' → u = ε → comprobar si el símbolo inicial es anulable

G' → u ≠ ε → algor. de eliminación de prod. unitarias → G'' → acepta/realiza → C Forma Norm. Chomsky

C → algoritmo de Cocke-Younger-Kasami

¿P acepta u? ⟺ ¿C genera u?

N/

13 - 14

NO DECI

SEMI

③ Dada MT determinar si acepta una palabra de long. ≤ 20.

NO DECI por teorema de Rice
(es una propiedad no trivial de los lenguajes re)
si dos MT aceptan el mismo lenguaje, una acepta una palabra de long ≤ 20 $\longleftrightarrow$ lo acepta la otra MT

SEMI hacemos una MTND, acepta como entrada la codificación de una MT M, $\langle M \rangle$.

Entrada: $\langle M \rangle$

1. Añadimos 111 al final
2. De forma no determinista añadimos palabra de long. ≤ 20, u.
3. Ejecutamos M_u sobre la palabra de la cinta; $\langle M \rangle 111 u$
   si: M_u acepta entonces
   ACEPTAR
   si no
   RECHAZAR

También podemos describir una MTD: recorrer en anchura:

Entrada $\langle M \rangle$

1. Hacer $\langle M \rangle 111$ (añadir 111 al final)
2. Ordenamos las palabras de $B = \{u \in A^* \mid |u| \leq 20\}$ de alguna manera (es finito, $|B| < \infty$)
3. Para $i = 1, 2, 3, \ldots, \infty$:
   ~~Copio $\langle M \rangle 111 u$ en una segunda cinta~~
   Ejecuto M_u i pasos sobre la segunda cinta
   si: M_u Acepta entonces
   ACEPTAR

④ Dada M determinar si, para cualquier entrada, para antes de 10 pasos de cálculo.

DECI

Basta comprobar si para con todas las palabras de long ≤ 10
si para con todas $\to$ SÍ
si no $\to$ NO

NO SEMI

⑤ ~~HALTING~~ dada MT M, determinar si para para cualquier entrada

L_u semi, no deci $\Longrightarrow$ $\overline{L_u}$ no semi

ALG: dada MT, determinar si para para cualquier entrada
C-UNIVERSAL: dada MT M palabra w determinar si M no acepta w

C-UNIVERSAL(M, w) = SÍ $\Longrightarrow$ ALG(F(M,w)) = SÍ
M no acepta w — F(M,w) siempre para

C-UNIVERSAL(M, w) = NO $\Longrightarrow$ ALG (F(M, w)) = NO
M acepta w — F(M, w) cicla para alguna entrada

F(M,w) tiene que ser una MT $\longrightarrow$

F(M,w)
entrada una palabra $v \in \{0,1\}^*$
1. almacenar $|v|$ en segunda cinta
2. Borrar v de primera cinta y colocar $\langle M \rangle 111 w$
3. Simulamos $|v|$ pasos de M con entrada w
4. Si M acepta w en $|v|$ pasos: entrar en bucle infinito $\Rightarrow$ se rechaza (cicla)
   si no, acepta entrada

⑥ Dada M determinar si no acepta ningún palíndromo.

C-PAL dada MT M, determinar si acepta un palíndromo

C-PAL no es decidible (teorema de Rice)
pero es semidecidible

MT entrada: <M> (codificación de M)
1. seleccionamos de forma no determinista $w \in \{0,1\}^*$
2. si w es palíndromo (lo comprobamos antes)
3. simula M con entrada w
si M acepta w
ACEPTAR
si no
RECHAZAR
si no
RECHAZAR

MT determinista: búsqueda en anchura:
$i = 1, 2, 3, \to \infty$
A_i = todas las palabras de long. menor que i
para cada $a \in A_i$:
si a palíndromo
simular i pasos de M con entrada a
si acepta
ACEPTO
else
~~RECHAZO~~ | no rechazar — dejamos que cicle hasta que ∃
else
~~RECHAZO~~

8
0

C-PAL $\boxed{\text{NO, S}}$ $\Rightarrow$ PAL $\boxed{\text{NS}}$

⑦ Dada M, determinar si termina escribiendo un 1 cuando comienza con una cinta completamente en blanco. ≡ ESC-1

UNIVERSAL $\propto$ ESC-1 $\Longrightarrow$ ESC-1 no decidible

UNIVERSAL(M, w) = sí $\Longrightarrow$ ESC-1(F(M, w)) = sí
M acepta w — cinta termina escribiendo 1

UNIVERSAL(M, w) = NO $\Longrightarrow$ ESC-1(F(M, w)) = NO
M no acepta w — cinta no termina escribiendo 1

F(M, w) entrada: $v \in \{0,1\}^*$
1 if $v \neq \varepsilon$
2 rechazar la entrada
3. else
copiamos M a $\tilde{M}$ donde sustituimos el símbolo 1 por $\tilde{1}$
4 llamamos $\tilde{w}$ a w sustituyendo 1 por $\tilde{1}$.
5. colocamos en la cinta $\langle \tilde{M} \rangle 111 \tilde{w}$
6 simulamos con M.. $\tilde{M}$ con entrada $\tilde{w}$
7 (M acepta $w \Leftrightarrow \tilde{M}$ acepta $\tilde{w}$)
8 si $\tilde{M}$ acepta $\tilde{w}$
9 escribimos un 1

M2

RELACIÓN 2

(5) PARADA: dada M y w, determinar si M para cuando tiene a w como entrada.

→ PARADA es semidecidible: construimos MTND

entrada: $\langle M \rangle 111 w$

1. ejecutar Mu sobre la cinta
2. si para
   ACEPTAR
3. si no
4. RECHAZAR

→ PARADA no es decidible: veamos UNIVERSAL $\propto$ PARADA

Para ello hacemos que:

$F(M, w) = \tilde{M}, w$, donde: la máquina $\tilde{M}$ se construye como sigue:

MT $\tilde{M}$ entrada: $v \in \{0,1\}^*$

colocamos en la cinta $\langle M \rangle 111 v$
ejecutamos Mu sobre la cinta
si para:
 si acepta: ACEPTAR
 si no acepta: CICLAR

claramente, F es un proceso algorítmico (podemos escribir la codificación de $\tilde{M}$ como hemos visto en clase).

Ahora vemos que:

$\text{UNIVERSAL}(M, w) = \text{SÍ} \implies \text{PARADA}(\overbrace{F(M, w)}^{(\tilde{M}, w)}) = \text{SÍ}$

si M acepta w, $\tilde{M}$ parará

$\text{UNIVERSAL}(M, w) = \text{NO} \implies \text{PARADA}(F(M, w)) = \text{NO}$

si M no acepta w, puede ser porque:
 M cicla con w, en cuyo caso $\tilde{M}$ también cicla
 M para y rechaza w, en cuyo caso $\tilde{M}$ cicla.

$\left.\begin{array}{l}\text{UNIVERSAL no decidible} \\ \wedge \\ \text{UNIVERSAL} \propto \text{PARADA}\end{array}\right\} \implies$ PARADA no decidible

⑥ NOTA: una lista $\{n_1, \dots, n_k\}$ se corresponderá con la salida de la MT $0^{n_1}1 \dots 10^{n_k}1$

a) Colocamos un 0 en la primera cinta y llamamos a la subrutina CUADRADO, que calculará el cuadrado de 0 en la última cinta, añadimos un 1 y un 0 a la primera cinta, y así sucesivamente

contenido de 1ª cinta

b) Colocamos en C1 (0^n) comprobamos si n es primo y si lo es colocamos en la última cinta 0^n1, añadimos otro 0 en C1 y repetimos. Iniciamos con 00 en C1.

c) Hacemos que la lista sea $u_1 2 u_2 \dots 2 u_k 2$, donde u_i es la codificación en $\{0,1\}^*$ de n_i $(i = 1, \dots, k)$

Denotamos $C(n)$ la codificación en $\{0,1\}^*$ de $n \in \mathbb{N}$.

Ciclo:
- colocar $C(n)$ en C1
- si $C(n)$ es una MT correcta (la codificación de ella):
  - colocar $C(n)\,111\,C(n)$
  - ejecutar Mu en C1
  - si Mu acepta
    - colocar $C(n)\,2$ en C2
- $n \leftarrow n+1$

comenzar con $n=0$

⑦ L_i re $\forall i \in \{1, \dots, k\}$, $A^* = \biguplus_{i=1}^{k} L_i$

Basta ver que

unión de re es re

$\overline{L_j} = \bigcup_{i=1, i\neq j}^{k} L_i$ recurs. en.

L_j re

$\Rightarrow L_j$ recursivo

cierto $\forall j \in \{1, \dots, k\}$

Escaneado con CamScanner

(8) L re no rec $L' = \{0w \mid w \in L\} \cup \{1w \mid w \notin L\}$

¿puede asegurarse de que L' o $\overline{L'}$ son rec, re o no re?

$\overline{L}$ no re (si fuese $\overline{L}$ re, entonces L re, $\overline{L}$ re $\Rightarrow$ L rec $\#$)

Veamos que L' no re reduciendo $\overline{L}$ a L'

Si existiera una MT que aceptase L', podríamos construir una MT que aceptase $\overline{L}$ (de la parte $\{1w \mid w \notin L\}$, pero esto no es posible $\# \Rightarrow L'$ no re

Del mismo modo, $\overline{L'}$ no re, pues:

$\overline{L'} = \{0w \mid w \notin L\} \cap \{1w \mid w \in L\}$ ______ NO

(9)

	Unión	Intersección	Concatenación	Clausura	Homom.	Hom. inverso
Recursivo	finita	finita	✓	✓	✓	✓
Recurs. enum	numerable	finita	✓	✓	✓	✓

(10) MT que acepta al menos un palíndromo es indecidible.

Consec del teorema de Rice: aceptar al menos un palíndromo es una propiedad no trivial de los lenguajes rec. enum.

(11) MT con dos tipos de estados: campana (c) y silbato (s) determinar que saber si una MT entrará en un estado s es indecidible

Hacemos C − VACIO (no decidible) $\propto$ SILBATO $\Longrightarrow$ SILBATO no decidible

C−VACIO(M) = SÍ $\Longrightarrow$ SILBATO(F(M)) = SÍ
L(M) ≠ ∅ entra a SILBATO

C−VACIO(M) = NO $\Longrightarrow$ SILBATO(F(M)) = NO
L(M) = ∅ no entra a SILBATO

F(M) convierte los estados finales de M en SILBATO y el resto en CAMPANA.

(12) indecidible saber si una MT termina escribiendo un 1 cuando comienza completamente en blanco

UNIVERSAL $\propto$ ESC-1

M acepta w | | al parar ε termina escribiendo un 1
UNIVERSAL(M, w) = SÍ $\Rightarrow$ ESC-1 (F(M, w)) = SÍ
UNIVERSAL(M, w) = NO $\Rightarrow$ ESC-1 (F(M, w)) = NO
M no acepta w | | no termina escribiendo 1 al parar ε

F(M, w) es la siguiente MT

entrada : $v \in \{0,1\}^*$

si $v \neq \varepsilon$: RECHAZAR
creamos $\tilde{M}$ MT copia de M sust. 1 por $\tilde{1}$
colocamos en la cinta $\langle \tilde{M} \rangle 111 \tilde{w}$ (sust. 1 por $\tilde{1}$)
ejecutamos M_u sobre la cinta
si acepta :
escribir 1
ACEPTAR

(13) a) determinar si una MT contiene al menos dos palabras distintas

Es RE : MTND

entrada : $\langle M \rangle$

seleccionar ND w
colocar en cinta $\langle M \rangle 111 w$
ejecutar M_u sobre la cinta
si acepta :
seleccionar ND w', $w' \neq w$
colocar en cinta $\langle M \rangle 111 w'$
ejecutar M_u sobre la cinta
si acepta :
ACEPTAR

veamos que no es decidible. Teorema Rice

b) determinar si el lenguaje de una MT es finito o infinito

claramente no decidible por T^a Rice, veamos no semidecidible :

DIAGONAL(M) = ⟦ M no acepta $\langle M \rangle$ ⟧
INFINITY(M) = ⟦ M acepta un número infinito de palabras ⟧

DIAGONAL $\propto$ INFINITY

F(M) : entrada $v \in \{0,1\}^*$

DIAG(M) = SÍ
M no acepta $\langle M \rangle$
$\Downarrow$
INF(F(M)) = SÍ
$|L(\tilde{M})| = \infty$

DIAG(M) = NO
M acepta $\langle M \rangle$
$\Downarrow$
INF(F(M)) = NO
$|L(\tilde{M})| < \infty$

idea: si M no acepta $\langle M \rangle$, hacemos que $L(F(M)) = 0^+$
en caso contrario, M aceptará $\langle M \rangle$ en t pasos, en cuyo caso
hacemos $L(F(M)) = \{0, 0^2, \dots, 0^{t-1}\}$

F(M) entrada: $v \in \{0,1\}^*$
si v no es de la forma 0^n, $n>0$:
RECHAZA
en otro caso
simulamos M con entrada $\langle M \rangle$ n pasos
si acepta:
RECHAZAR
else:
ACEPTAR

M no acepta $\langle M \rangle \Rightarrow$ no lo aceptará en ningún caso, para ningún
nº de pasos, luego $L(F(M)) = 0^+$
M acepta $\langle M \rangle$ en t pasos:
si $n \geq t \Rightarrow$ F(M) rechaza 0^n
si $n < t \Rightarrow$ F(M) acepta 0^n

[otra forma: ver que su complementario es no decidible, pero
semidecidible]

c) Determinar si el lenguaje de una MT es indep. del contexto.

C-UNIVERSAL $\propto$ IC $\Rightarrow$ IC no semi
no semi

C-UNIV(M, w) = sí $\Longrightarrow$ IC(F(M, w)) = sí
M no acepta w
F(M, w) es IC

C-UNIV(M, w) = No $\Longrightarrow$ IC(F(M, w)) = NO
M acepta w
F(M, w) no es IC

Tomamos un lenguaje no indep. del contexto, y otro IC.
$L_{NIC} = \{0^n 1^n 0^n : n \in \mathbb{N}\}$ $\quad L_{IC} = \emptyset$ (leng vacío)

F(M, w): entrada: $v \in \{0,1\}^*$
si v no es de la forma $0^n 1^n 0^n$: RECHAZA
en otro caso:
si M acepta w:
ACEPTA
else
RECHAZA

vemos que $\begin{cases} M \text{ no acepta } w \Rightarrow L(F(M, w)) = L_{IC} \\ M \text{ acepta } w \Rightarrow L(F(M, w)) = L_{NIC} \end{cases}$

(14) $L = \{ (M_1, M_2, k) : |L(M_1) \cap L(M_2)| \geq k \}$ re, no dec

por el teo de Rice, no es decidible, en efecto la propiedad $L(M_1) \cap L(M_2)$ es no trivial sobre los lens r.e.:

$L(M_1)$ acepta cualquier palabra
$L(M_2)$ acepta únicamente las palabras 0^i

si $i = k$, $L(M_1) \cap L(M_2)$ contiene al menos k palabras,
si $i < k$, $L(M_1) \cap L(M_2)$ no contiene ______.

es semidecidible (re). coloco ND la palabra y acepto si M_1 y M_2 aceptan.

(15) a) M tq al comenzar con la cinta en blanco en algún momento escribirán un símbolo no blanco en la cinta

?

¿y si cicla?

b) Podemos codificar MT como hemos hecho en clase y recorrer para comprobar que no hay ningún movimiento a la izquierda.

?

¿Inaccesibles?

c) Modificamos M para que tenga una cinta marcada con × para cada posición leída.

?