9. Demuestra que los siguientes conjuntos son semi-decidibles:

(a) $\{\langle x, y\rangle | M_x(y)\downarrow\}$.

$\langle u,v,R\rangle \in L \iff \langle u,v,R\rangle \in L(M_L)$

(b) $\{x | \exists y : M_x(y)\downarrow\}$.

$\langle u,v,R\rangle \in L \iff u \to_R^* v \iff M_L(u,v,R)\downarrow \iff \langle u,v,R\rangle \in L(M_L)$

Fent substitucions para ra sino,

(c) $\{\langle u, v, R\rangle \mid u \to_R^* v\} = C = \{\langle u,v,R\rangle \mid u,v \in \Sigma^+ \wedge R \in \Sigma^+ \times \Sigma^+ \wedge u \to_R^* v\}$

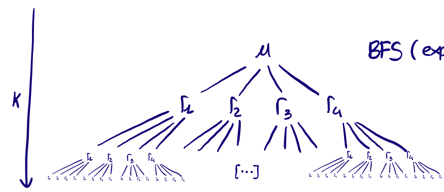
Conjunto de numeros naturales que codifican tripletas
- $u \in \Sigma^+$ (palabra no vacía)
- $v \in \Sigma^+$ (palabra no vacía)
- $R \in \Sigma^+ \times \Sigma^+$ (sistema de reescritura con reglas de palabras no vacías)

tales que desde u se puede acceder a v usando R.

pseudocode

R es finito

· vectos de niveles
· nivel contiene palabras



BFS (exploración en anchura)

zona ≡ nivel ≡ #reescrituras aplicadas.

```
K = 0
loop:
    go to zone K
    it = 0;
    for each word en la zona K
        for each r in R
            newword = reescribir(word, r)    TM reescribe y obtenemos ρ_ri
            if(newword ≠ "null")
                if (newword == v) accept;    TM dice si son iguales (simular)
                write newword in zone K+1 in position it   // llegas a la ultima zona y contar it separadores
                it++
        endfor
        if (it == 0) reject;   // no se han producido reescrituras ⇒ Palabra no incluido
        else it = 0;
    endfor
    ++ K
goto loop;
```

semidecidible, porque puede que nunca lleguemos aquí.

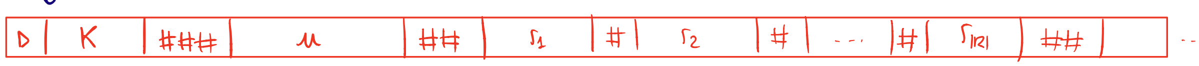
M_{r_i} entrada $\langle w, r_i\rangle$

M_{r_i} Maquina que hace la funcion de reescritura, segla r_i,

[imaginacion de la cinta]

u,v,R

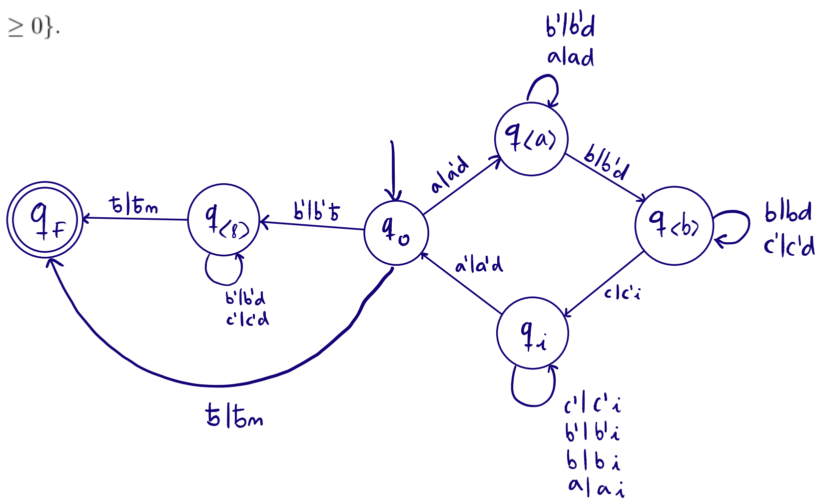
Zona K=0



(d) $\{G \in \texttt{CFG} \mid G \text{ ambigua}\}$.

1. Escrivid TM sencillas para los siguientes lenguajes:

(a) $\{a^n b^n c^n | n \geq 0\}$.



(c) $\{ww | w \in \{0,1\}^*\}$.

3. Argumenta a grandes rasgos que las máquinas de Turing no-deterministas no son más expresivas que las máquinas de Turing deterministas.

- Determinista → arbol de computación con hojas aceptadoras
- No determinista → Busca en el arbol que simula, con un BFS, la config aceptadora.

4. Considera el modelo de máquina que definimos a grandes rasgos así: una variante de los autómatas con pila donde, en vez de una pila, tenemos dos pilas, las transiciones dependen del contenido de la cima de ambas pilas, y en la acción de cada transición se puede o bien borrar el elemento de la cima o bien añadir nuevos elementos, todo ello en ambas pilas. Justifica a grandes rasgos que este modelo puede simular una máquina de Turing, y que, por tanto, es Turing-completo.

PDA con 2 pilas

6. Demuestra que los lenguajes decidibles son cerrados por las siguientes operaciones:

(a) Intersección.

$L_1, L_2 \in DEC$

$L = L_1 \cap L_2 \in DEC$

M_1 decideix L_1

M_2 decideix L_2

$\forall w \in \Sigma^*$:

simula $M_1(w)$

simula $M_2(w)$

$\rightarrow$ si ambdos accepten $\Rightarrow$ acceptem $w \Rightarrow w \in L$

else rebutjem $w \Rightarrow w \notin L$

(b) Complementario.

$A \in DEC$

M decideix A

$M'(w)$

simular $M(w)$

si M accepta $\Rightarrow$ rebutjar

else accepta

(c) Resta (de conjuntos).

$L = L_1 - L_2 \equiv L_1 \cap \overline{L_2}$

[· intersección
· complementación]

como ambas son operaciones cerradas en el conjunto de los decidibles $\Rightarrow L \in DEC$

(e) Concatenación.

8. Demuestra que los lenguajes semi-decidibles son cerrados por las siguientes operaciones:

(b) Concatenación.

$L_1, L_2 \in$ DEC

$x \in L_1$

$y \in L_2$ $\quad w = xy$

TM M:

$\forall$ cortes posibles de w separando en $w_1 w_2$:

simula $M_A(w_1)$

simula $M_B(w_2)$ [un numero de pasos concretos]

(d) Estrella.

(e) Morfismo directo.

$L \in$ semi-DEC $\Rightarrow \sigma(L) \in$ semi-DEC

$\Downarrow$

$(\exists M_2 \in TM \mid M_2$ reconeix $L)$

6.11)

$C \in DEC \iff \exists f$ funcio computable, total, injectiva, creixent amb $Im(f) = C$

$\Leftarrow)$ f comp. $\Rightarrow \exists M_f \in TM \mid M_f$ computa f

$C \in DEC \Rightarrow \exists M_c \in TM \mid \mathcal{L}(M_c) = C \wedge \forall x \; M_c(x)\downarrow$

$\Rightarrow)$ M_f computa f

```
Mf:
  Entrada i
  m = 0   // fuerza bruta, entradas que se estan probando
  k = 0   // contador de cuantas aceptaciones (imagenes)
  While (k < i) {
      if (Mc(m) accepta) ++k;
      ++m;
  }
  return x_m;
```

potence x a c??

siempre para

creixent

```
Mc:
  entrada x :
  y = 0
  loop :
      Simula Mf(y)
      Si (φf(y) < x) rebutja
      Si (φf(y) == x) accepta;
      ++y;
      goto loop;
  endloop;
```

6.12)

$C \in DEC \iff \exists f$ funcio computable, total, injectiva amb $Im(f) = C$

$\Leftarrow)$ f comp. $\Rightarrow \exists M_f \in TM \mid M_f$ computa f

$C \in DEC \Rightarrow \exists M_c \in TM \mid \mathcal{L}(M_c) = C \wedge \forall x \; M_c(x)\downarrow$

$\Rightarrow)$ M_f computa f

```
Mf
  Entrada i
  m = 0   // fuerza bruta, entradas que se estan probando
  k = 0   // contador de cuantas aceptaciones (imagenes)
  While (k < i) {
      if (Mc(m) accepta) ++k;
      ++m;
  }
  return x_m;
```

Hacerlo en k pasos explorando p diagonals o por cuadrados. Porque Mc puede no parar.

potence x a c??

siempre para

```
Mc:
  entrada x :
  y = 0
  loop :
      Simula Mf(y);
      Si (φf(y) == x)          accepta
      ++y;
      goto loop;
  endloop;
```

;

6.13) es computable, esta es la TM

$M_{f^{-1}}$: entrada y
$t \leftarrow 0$
simular $M_f(w_0), \dots, M_f(w_t)$ amb t passos
if ($\exists i \leq t$ $M_f(w_i)$ acaba en $\leq t$ passos) return w_i;
$t \leftarrow t+1$

6.14) $f: \mathbb{N} \to \mathbb{N}$ estrictament decreixent. És computable?

f es total

$f(0) > f(1) > f(2) > \dots > f(n)$

$f(f(0)+1) \leq f(0) - f(0) - 1 = -1 \Rightarrow -1 \notin \mathbb{N}$ no existe ninguna función

$\Downarrow$

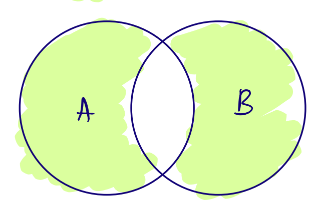
es computable

Domini finit $\Rightarrow$ computable

6.16) $\left.\begin{array}{l}(A \cup B) - (A \cap B) \in DEC \\ A \in \text{semi-}DEC\end{array}\right\} \Rightarrow B \in \text{semi-}DEC$

$L \in DEC$

$(A \cup B) - (A \cap B) \equiv (A \cup B) \cap \overline{(A \cap B)} \longrightarrow$



$A =$ halt $\in$ semi-DEC

$(A \cup B) - (A \cap B) = \Sigma^* \in DEC$

$B = \overline{\text{halt}} \to$ no semi-DEC

$\Sigma^* - \emptyset = \Sigma^*$

6.6) f) $L \in DEC$, $L^* \in DEC$?

Suposem que M decideix L

$M_*(w)$:
- si ($w = \lambda$) accepta
- $\forall$ partició de w en $w_1, w_2, w_3 \ldots$
  - simula $M(w_1), \ldots, M(w_k)$
  - si M accepta k vegades $\Rightarrow$ accepta
- rebutja;