QCM : pour répondre il faut savoir :

Langage | décidable, récursif, Calculable = $\exists$ une MT M sans exécution infinie qui l'accepte (s'arrête pour dire oui et non)

$\hookrightarrow$ fct $x \to f(x)$

mot $\to$ [M] $\xrightarrow{\in L}$ oui (accepte) ; $\xrightarrow{\notin L}$ non (se bloque)

$L(M) \in R$

Langage | semi-décidable, récursivement énumérable = $\exists$ une MT M qui l'accepte avec ou sans exécution infinie

$L(M) \in RE$

$\to$ [M] $\xrightarrow{\in L}$ oui (accepte) ; $\xrightarrow{\notin L}$ Non (bloque) ; $\xrightarrow{?}$ boucle)

$\boxed{R \subset RE}$

Langage | régulier = reconnu par une automate fini

$L(M) \in R_L$

cas particulier des MT qui s'arrêtent

$\boxed{R_L \subset R}$

un langage fini $\Rightarrow$ alphabet fini $\Rightarrow$ régulier $\Rightarrow$ automate fini $\Rightarrow$ MT $\Rightarrow$ décidable

exemple : RE

L = equations polynomiales with coeff entiers dont les solutions sont dans $\mathbb{N}$

$x^3 + y^3 + z^3 = 0 \Rightarrow$ accept

$x^2 + y^2 + 1 = 0 \Rightarrow$ reject

$x^4 + 2y^3 + z^4 = 5 \Rightarrow$?

si c'était décidable on aurait alors une méthode qui détermine si une équation donnée a une solution entière ou non.

ni R ni RE
Langage universel :
$LU = \{<M, w> / M \text{ accepte } w\}$
indécidable

L = programmes qui crashent sur un input.
si c'était décidable la vie aurait été si belle

exemple : indécidable
une machine analyse le code d'une autre + input et décide si cette dernière s'arrête ou non sur cet input.

problème de l'arrêt.

Pour mieux comprendre les relations entre Langages

soient L_1 et L_2 décidés par M_1 et M_2

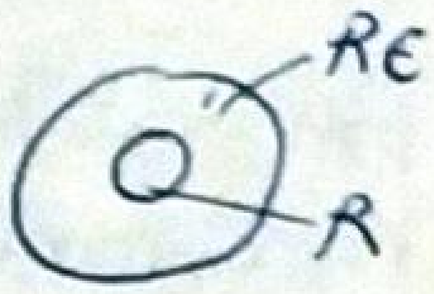
$\cap$ intersection	$\cup$ union	$\bar{}$ ou ^{c} complémentaire
savoir si $x \in L_1 \cap L_2$	savoir si $x \in L_1 \cup L_2$	savoir si $x \in \bar{L_1}$ ou L_1^c
on simule M_1 et M_2 sur x et on accepte ssi les deux acceptent	on simule M_1 et M_2 sur x et on accepte ssi l'une des deux accepte	on simule M_1 sur x et on accepte ssi M_1 rejette

$L_1.L_2$ concaténation.
juxtaposition des deux

fermeture de kleene *
L^* savent dire ε inclus (vide)

La classe des langages récursifs est close pour ces opérations
" " " " semi décidables n'est pas close pour le $^c L$ ou $\bar{L}$

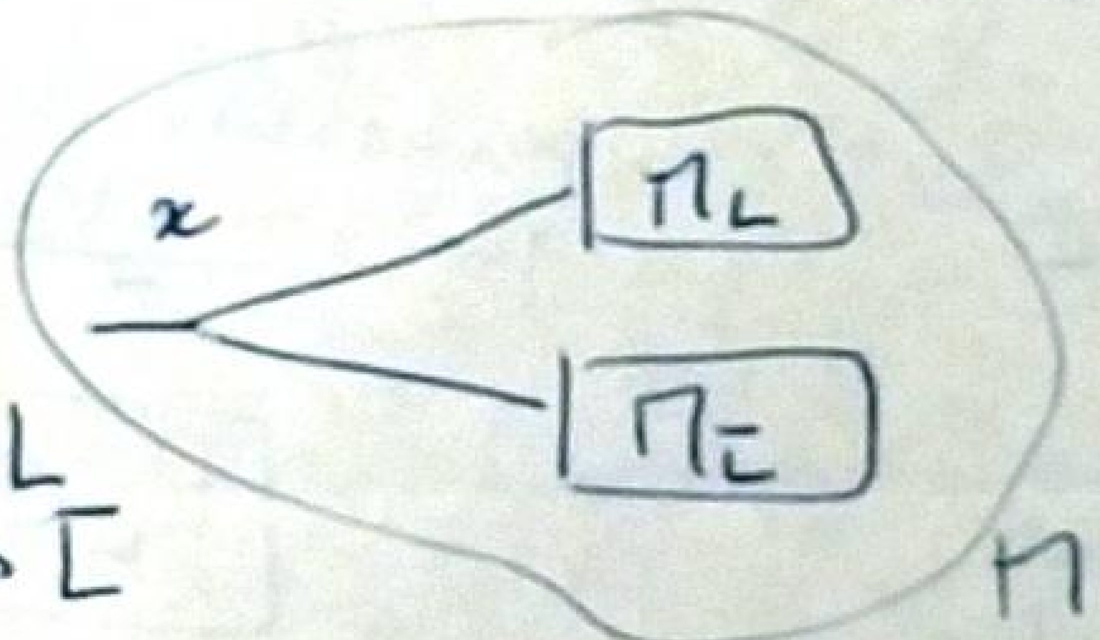
→ il existe des langages RE mais pas R.



$\boxed{\text{mais si } L \in RE \text{ et } \bar{L} \in RE \text{ alors } L \in R \text{ et } \bar{L} \in R}$

$L \in RE$ — $\bar{L} \in RE$ ou non

Preuve :



x soit dans L
soit dans $\bar{L}$

si M_L accepte alors M accepte
si $M_{\bar{L}}$ accepte alors M rejette.
donc M s'arrête toujours.

→ Tout langage accepté par une MT non déterministe est accepté par une MT déterministe

→ les fcts calculables par une P.E sont calculer par une MT.

proc. effectif

Thèse de Church-Turing.

(exactement la marche à suivre pour résoudre le problème

f calculable par une MT ⟹ f calculable par un algo ou P.E

$\boxed{\Longleftarrow \text{ est vrai ssi MT s'arrête tjrs}}$

Réduction :
Mq $L_2 \notin R$ sachant $L_1 \notin R$
supposons
$L_2 \in R \Rightarrow \ldots$
...
absurde $\Rightarrow L_1 \in R$
donc $L_2 \notin R$

$L \in RE \setminus R \Rightarrow \underbrace{\bar{L} \notin RE}$
car sinon L et $\bar{L} \in RE$
et $L \in R$ or $L \in RE \setminus R$.
absurde

théorème de Rice : toute propriété sémantique non triviale d'un programme est indécidable

si je trouve que $L(M) \subset \phi$ alors $L(M)$ est indécidable

Ex : $M1$ et $M2$ 2 Machines de Turing. peut on décider

$\boxed{\text{si } L(M1) \subset L(M2) \text{ ?}}$

si $M2$ ne reconnait rien
donc $L(M1) \subset \phi$
donc $L(M)$ indécidable

→ MT mot w et entier k peut-on décider en au plus k pas de calcul si MT accepte w (contrairement au p_p de l'acceptation)

2 cas après k pas de calcul :
- w est accepté
- w non accepté

⇒ MT a décidé le mot.

Complexité

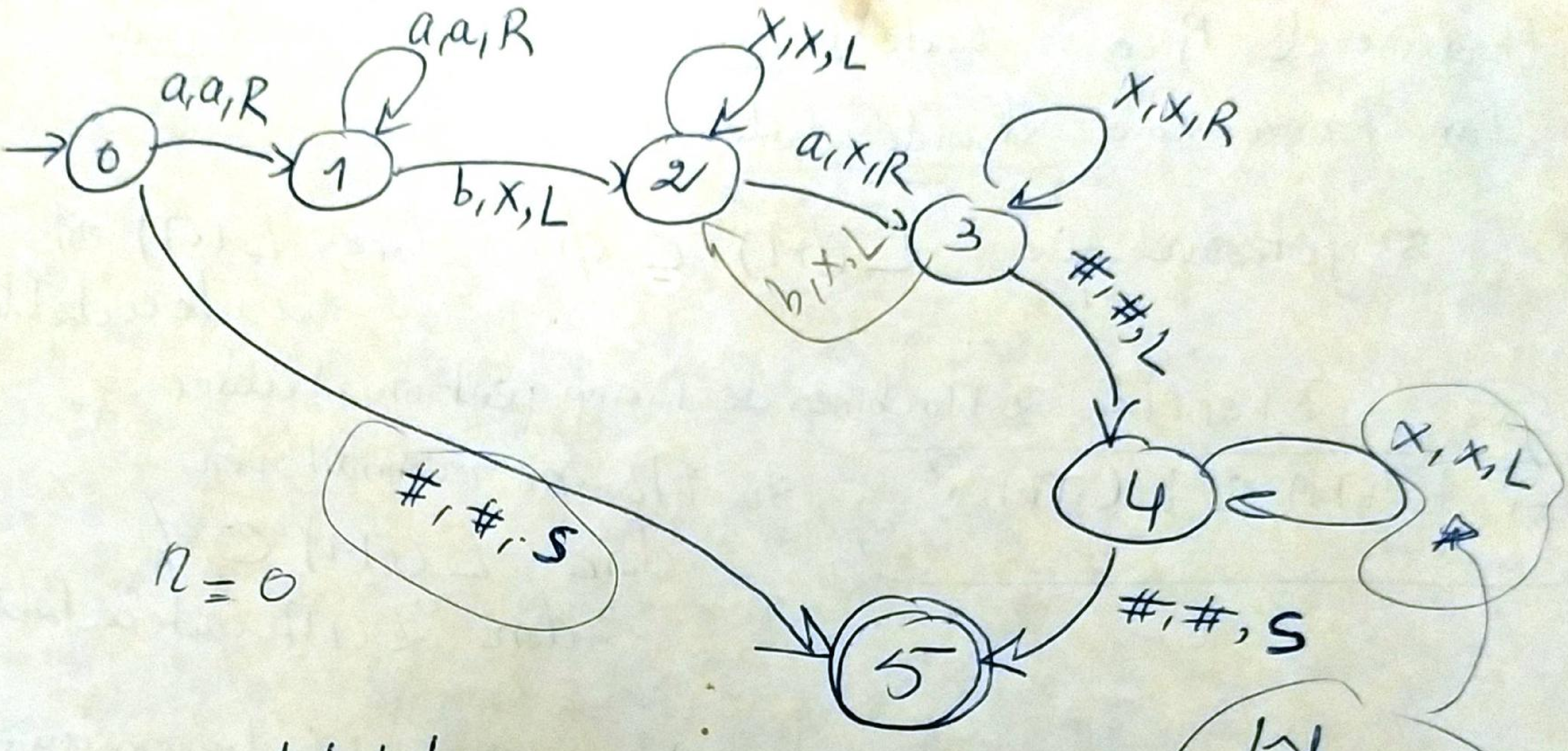
on associe à une taille d'entrée un nombre maximal de pas de calcul que la MT va effectuer avant de s'arrêter.
(calcul par un mot qui remplit les contraintes).
pire des cas

- chaque transition prend une unité de temps
- Grand $O(\,)$ $g(n) \in O(f(n))$ si $\exists k, N$ tq : $\forall n > k : |g(n)| \leq k f(n)$

f croît toujours plus que g.

fcts de référence triés en O

$\log(n)$, n , $n\log(n)$, n^k , k^n , $n!$, n^n →



$n = 0$

tête au début.

aaaa bbbb

aaaa Xbbb

$\cancel{2n}$

$$C = n + \sum_{i=1}^{2n} i$$

$$+ 2n \qquad = 3n + \sum_{i=0}^{2n} i$$

$$= 3n + \frac{2n(2n+1)}{2}$$

$$= 3n + 2n^2 + n$$

$$= 2n^2 + 4n$$

$$= O(n^2)$$

$$\sum_{n=0}^{n} n = \frac{n(n+1)}{2}$$