un problème introductif

P programme

x variable

est-ce que $P(x)$ affiche "hello world" ?

solutions

- lancer le programme : mais s'il plante ?
- mettre une contrainte de temps : si on la dépasse ?
- analyser le code : comment ?

dernier théorème de Fermat

$\forall$ a, b, c entiers $\geqslant 2$

$$a^n + b^n \neq c^n$$

très dur à prouver

soit le pseudo-code

pour tout a, b, c entiers ≥ 2

si $a^n + b^n = c^n$

afficher "hello world"

l'analyser c'est résoudre le dernier théorème de Fermat soit un tel analyseur $\mathcal{T}$

soit $\mathcal{T}_1$ tel que

- si $\mathcal{T}(P, x) =$ "oui" $\mathcal{T}_1(P, x) =$ "oui"
- "non" "hello world"

soit $\mathcal{T}_2(x) = \mathcal{T}_1(x, x)$ x séquence de bits (peut être un programme) si $\mathcal{C}_2(\mathcal{C}_2) =$ "oui"

alors $\mathcal{C}_1(\mathcal{C}_2, \mathcal{C}_2) =$ "oui"

$\mathcal{C}(\mathcal{C}_2, \mathcal{C}_2) =$ "oui"

$\mathcal{C}_2(\mathcal{C}_2) =$ "hello world"

absurde

si $\mathcal{C}_2(\mathcal{C}_2) =$ "hello world"

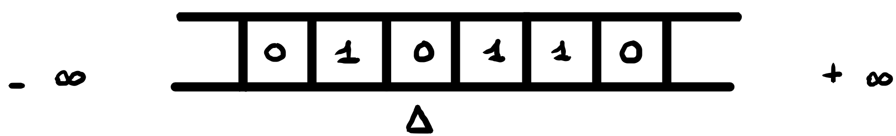
contradiction aussi

T n'existe pas !

un modèle abstrait d'ordinateur

une mémoire infinie (approximation) qu'on lit et qu'on modifie

sous forme de ruban avec des cases



une tête de lecture qui va à gauche et à droite plus un registre avec un nbre fini de valeurs

ok pour l'ordinateur

et pour le programme ?

une suite de règles de la forme

. lit une case

. modifie la case

. bouge la tête de lecture à gauche ou à droite

la machine de Turing

$(Q, q_0, q_a, \Sigma, \Gamma, B, \delta) = M$

- Q états finis (registre) $\quad Q \cap \Gamma = \emptyset$
- $q_0 \in Q$ état initial
- $q_a \in Q$ état final
- Σ alphabet d'entrée (contenu initial des cases)

- Γ alphabet de travail (contenu de la mémoire en cours d'exécution)

  $\Sigma \subseteq \Gamma$

- $B \in \Gamma$ symbole blanc $B \notin \Sigma$

- δ fonction de transition partielle

$$\delta : Q / \{q_a\} \times \Gamma \longrightarrow Q \times \Gamma \times \{L, S, R\}$$

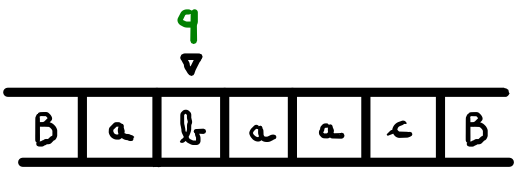
L, left
S, stay
R right

configuration

contenu de la mémoire : ruban + registre

mot infini dans $Conf_M = B^\omega \, \Gamma^* \, Q \, \Gamma^* \, B^\omega$

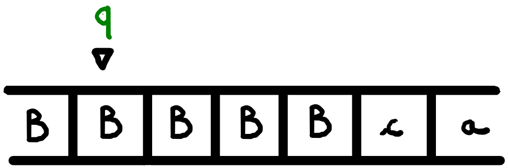
$B^\omega \; a \; q \; baac \; B^\omega$



$\cancel{B^\omega} \; a \; q \; baac \; \cancel{B^\omega}$

représentation finie

attention aux B entre la tête de lecture et le mot



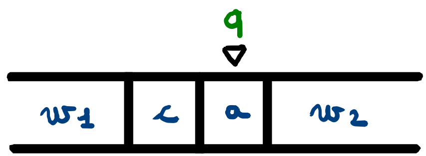
noté q BBBB ca

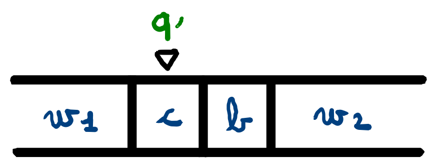
représentation graphique

si $\delta(q, a) = (q', b, x)$

sémantique

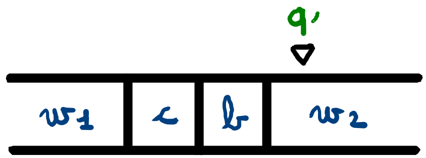
si $\delta(q, a) = (q', b, X)$

partant de 

. si $X = L$ on va à 

noté $w_1 \; c \; q \; a \; w_2 \vdash_M w_1 \; q' \; c \; b \; w_2$

. si $X = R$,

on va à 

noté $w_1 \, c \, q \, a \, w_2 \vdash_M w_1 \, c \, b \, q' \, w_2$

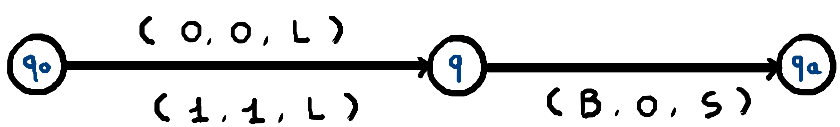
. si $X = S$,

alors $w_1 \, c \, q \, a \, w_2 \vdash_M w_1 \, c \, q' \, b \, w_2$

multiplication par 2 en binaire

$\Sigma = \{0, 1\}$ little endian

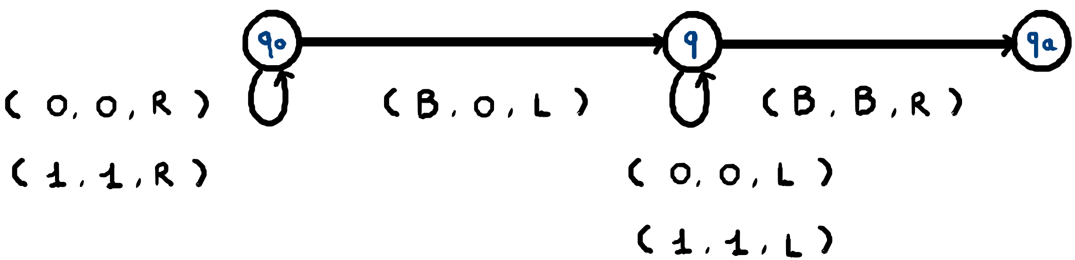
on cherche une MT M telle que pour tout mot w, on a $q_0 \, w \vdash_M q_a \, w'$, $w' = 2 \times w$



multiplication par 2 en binaire

$\Sigma = \{0, 1\}$ big endian

on cherche une MT M telle que pour tout mot w, on a $q_0\, w \vdash_M q_a\, w'$, $w' = 2 \times w$



un run de M sur un mot x est une suite de configurations $(c_i)_i$ telle que

$c_0 = q_0 \, x$

et

$\forall i, \quad c_i \vdash_M c_{i+1}$

finie ou infinie

accepter, rejetter, s'arrêter

M s'arrête sur un mot x ssi

$q_0\, x \vdash_M^* w\, q\, a\, w'$ tq $\delta(q, a)$ n'est pas définie

si $q = q_a$, M accepte x et sa sortie est $M(x) = w\, a\, w'$

sinon M rejette x

équivalence

$M_1(x)$ est équivalente à $M_2(y)$ ssi

$M_1(x)$ s'arrête ssi $M_2(y)$ de même
rejette
accepte même sortie

noté $M_1(x) \equiv M_2(y)$

$M_1 \equiv M_2$ ssi $M_1(x) \equiv M_2(x)$ $\forall x$

le modèle binaire

soit M MT sur un alphabet Σ, on cherche une machine binaire "équivalente" M' sur $\Sigma' = \{0, 1\}$

$\Gamma' = \{0, 1, B\}$

idée : encoder $\Gamma / \{B\}$ en binaire

il faut $k = \lceil \log_2 (|\Gamma| - 1) \rceil$

et un encodage $\beta : \Gamma / \{B\} \longrightarrow \{0, 1\}$

puis représenter une case de Γ par k cases de

Γ'

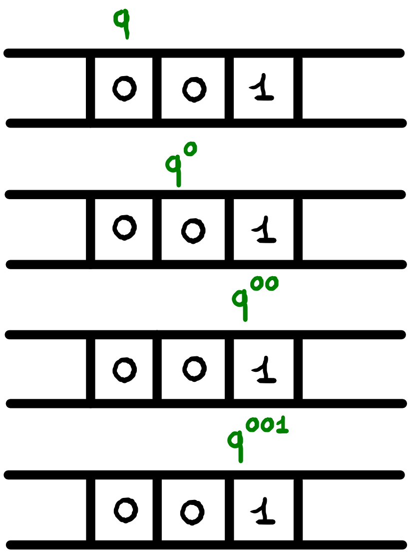
pour lire k cases de Γ' " d'un coup ", on

mémorise dans l'état les symboles lus jusqu'à

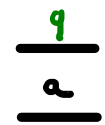
avoir lu k cases

$Q' = Q \times \{0, 1\}^k \cup$ un peu de machinerie par exemple

si $a \xrightarrow{\beta} 001$



équivaut à lire



puis appliquer $\delta(q, a)$

cf polycopié