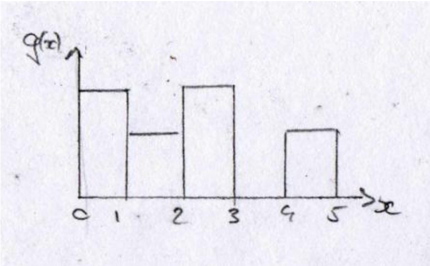
Calculabilité & Complexité – cours du 17/02

Exemple de minimalisation



$\mu_x g_{x=0} = 3$

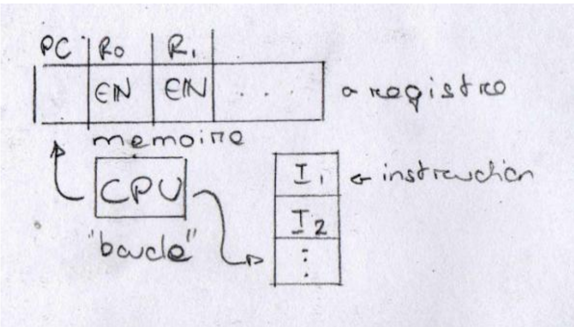
Algorithme descriptif :

Pour toujours si g(x) == 0

Alors renvoyer x

Sinon x = x + 1

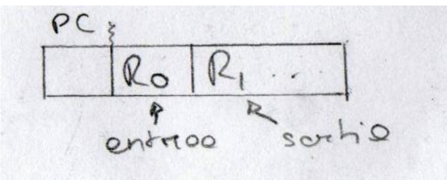
Constitution d'un ordinateur



CPU lit le registre mit en mémoire dans le PC (selon l'entrée clavier, ect …), puis lance l'instruction correspondante en bouclant

Le CPU s'arrête si il n'y a pas d'instruction qui s'exécute

On limite les entrées et sorties :



On fait alors une bijection : $\mathbb{N}^2 \rightarrow \mathbb{N}\_\{0\}$

$$< x, y > = \frac{(x+y)(x+y+1)}{2} + y + 1$$

	0	1	2	3	4
0	1	3	6	10	15
1	2	5	9	14	
2	4	8	13		
3	7	12			
4	11				

→ Fonction PR par composition de fonction

Avec $\mathbb{N}^3 \rightarrow \mathbb{N}\_\{0\}$

$(x, y, z) \rightarrow < x < y, z >> \rightarrow < x < y < z, 0 >>> \Rightarrow$ encodage imbriqué

Ainsi en choisissant R_0 en entree et R_1 en sortie, on a les possibilités de calcul ci-dessous :

$R_k = R_k + 1$

$R_k = R_k + 1$

$IF\ R_k \neq 0\ THEN\ GOTOB\ n$ (B = back)

$IF\ R_k \neq 0\ THEN\ GOTOF\ n$ (F = forward)

$$[PC, R_0, R_1, \ldots, R_k]$$

$$R_k \neq 0 \qquad\qquad R_k = 0$$

$$[PC = n, R_0, R_1, \ldots, R_k] \qquad [PC + 1, R_0, R_1, \ldots, R_k]$$

Machines RAM

PC	initial R_0	final R_1	R_2	R_3
1	x	0	0	0
2	x	0	0	1
3	x	0	0	2
4	x	0	1	2

1. $R_3 = R_3 + 1$
2. $R_3 = R_3 + 1$
3. $R_3 = R_3 + 1$

Au demarrage : PC = 1, $R_0 = x$

Gödelisation

A) $R_k = R_k + 1 \underset{\rightarrow}{g} 3k$

S) $R_k \dot{-} 1 \underset{\rightarrow}{g} 3k+1$

B) $IF\ R_k \neq 0\ THEN\ GOTOF\ n \underset{\rightarrow}{g} 3 < k < 1 < n, 0 >>> \dot{-} 1$

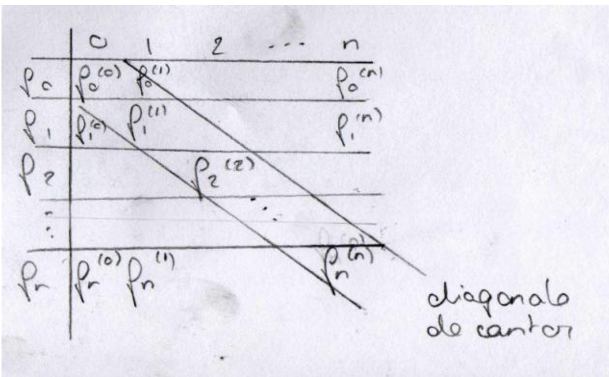
$IF\ R_k \neq 0\ THEN\ GOTOB\ n \underset{\rightarrow}{g} 3 < k < 0 < n, 0 >>> \dot{-} 1$

G(P) ≡ $\begin{matrix} i_1 \\ i_2 \\ \vdots \\ i_n \end{matrix}$ $< g(i_1) < g(i_2) < \cdots < g(i_n), 0 >> \cdots >$

G est injective, et définie par G : RAM → ℕ, |RAM| = ℕ (somme des programmes dénombrables)

On veut prouver que : $\mathbb{N} \rightarrow \mathbb{N} = \mathbb{N}^{\mathbb{N}} \supset \mathbb{N}$

On va supposer que $\mathbb{N}^{\mathbb{N}} = \mathbb{N}$



$n \rightarrow fn^{(n)+1}$

On va imaginer que la fonction $fn^{(n)+1}$ ait la position j dans le tableau

Mais $f_j^{(j)} = f_j^{(j)+1}$ ⟹Il y a donc une contradiction !

Donc $\mathbb{N}^{\mathbb{N}} \neq \mathbb{N} \implies \mathbb{N}^{\mathbb{N}} \supset \mathbb{N}$ vrai !

$R_k = R_k + 1 \implies (PC, R_0, \ldots, R_k) \underset{\rightarrow}{\delta} (PC+1, R_{0,} \ldots, R_{k+1})$

$R_k = R_k \dot{-} 1 \implies (PC, R_0, \ldots, R_k) \underset{\rightarrow}{\delta} (PC \dot{-} 1, R_{0,} \ldots, R_{k \dot{-} 1})$

δ(photo, instruction)= nouvelle photo

IF $R_k \neq 0$ THEN GOTOB n

$$\delta\big((PC, R_0, \ldots, R_k), I\big) = \begin{cases} \big(PC+1, R_{0,} \ldots, R_k\big) si\ R_k = 0 \\ (PC \dot{-} n, R_{0,} \ldots, R_k) \text{ sinon} \end{cases}$$