Complexité d'un programme universel

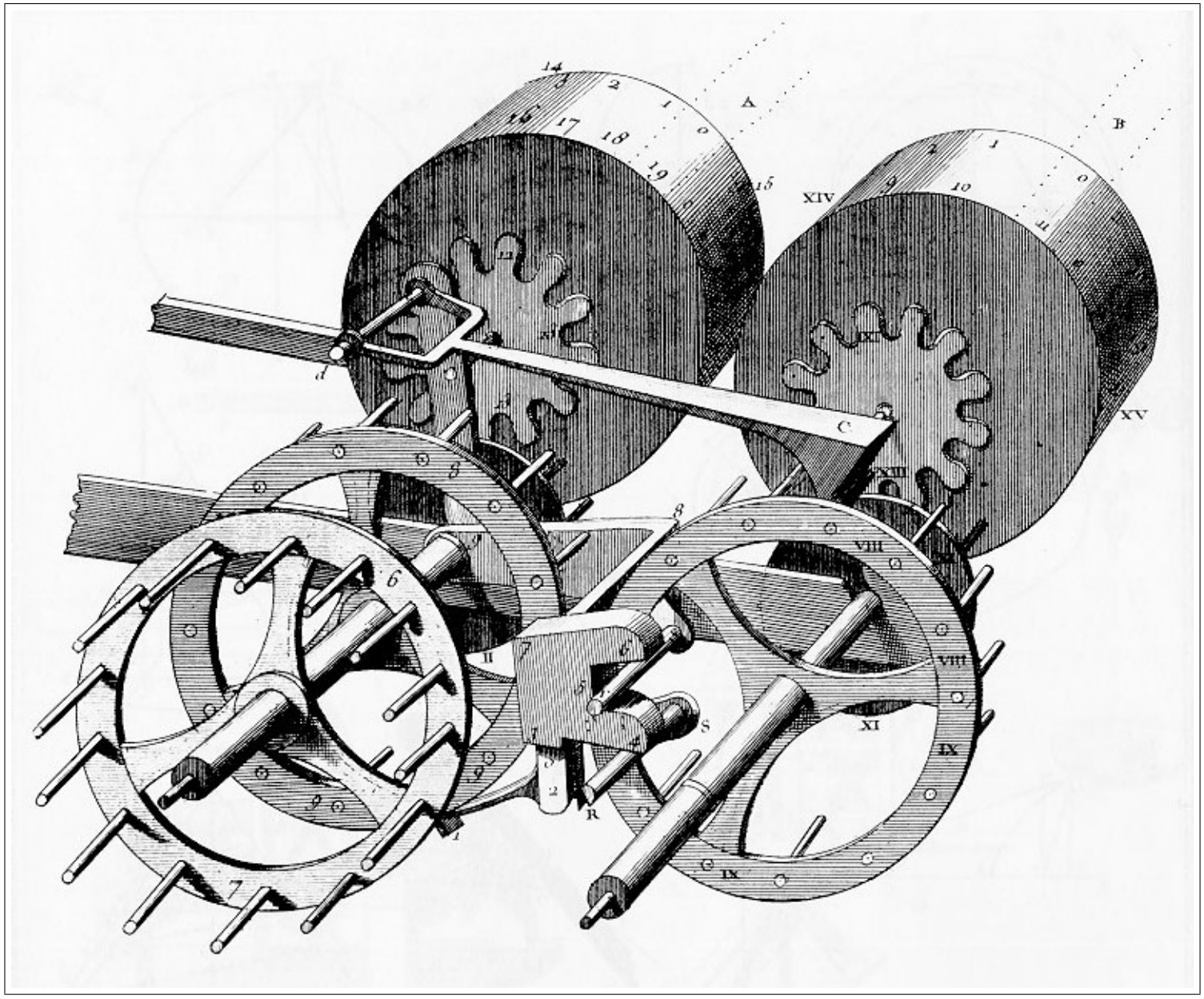
Alain Colmerauer

Luminy, 15 mars 2004

Laboratoire d'Informatique de Marseille
CNRS, Universités de Provence
et de la Méditerranée

Table des matières

Qu’est-ce qu’une machine ?

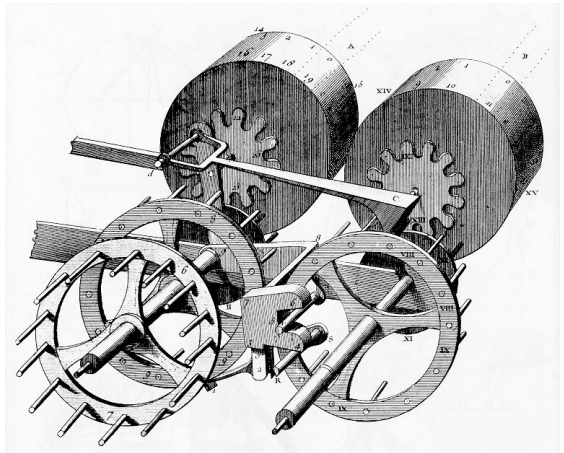

A
B
C
XIV
XV
VIII
VIII
XI
IX
IX
R
S

Une machine à traduire

$\Sigma = \{\sqcup, 0, 1, \ldots, 9, a, b, \ldots, z\}$

`green␣ideas␣sleeep␣furiously`

$\Downarrow$



Programme de traduction

$\begin{cases} \text{instruction 1,} \\ \text{instruction 2,} \\ \ldots \\ \text{instruction } n \end{cases}$

$\Downarrow$

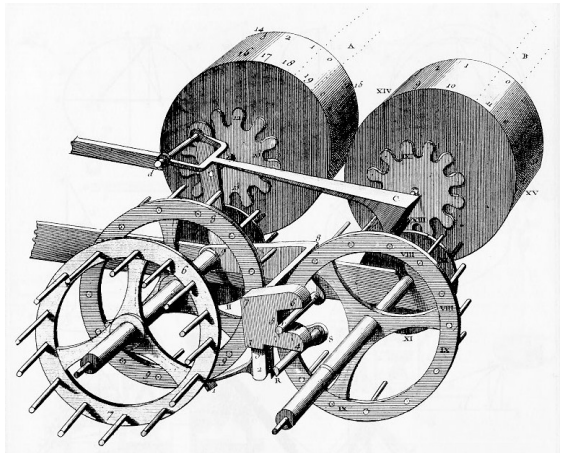
`des␣idees␣vertes␣dorment␣furieusement`

Une machine à multiplier

$\Sigma = \{\sqcup, 0, 1, \ldots, 9, a, b, \ldots, z\}$

`26010x37979721`

$\Downarrow$



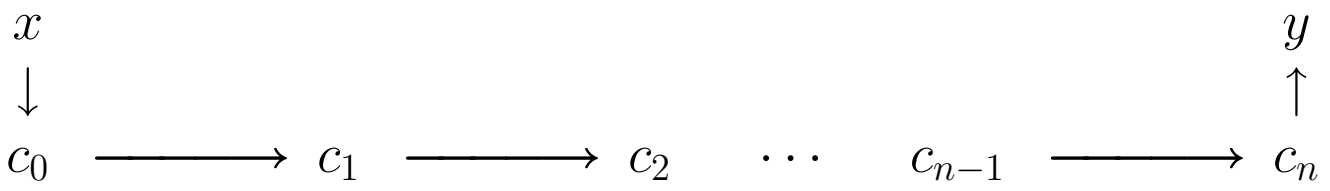
Programme de multiplication

$\left\{\begin{array}{l}\text{instruction 1,}\\ \text{instruction 2,}\\ \ldots\\ \text{instruction } n\end{array}\right\}$

$\Downarrow$

`9876543210`

Une machine est un système dynamique



Les ingrédients composant une machine

• Une machine est un couple (M, P), où M est une machine programmable et P un programme pour M.

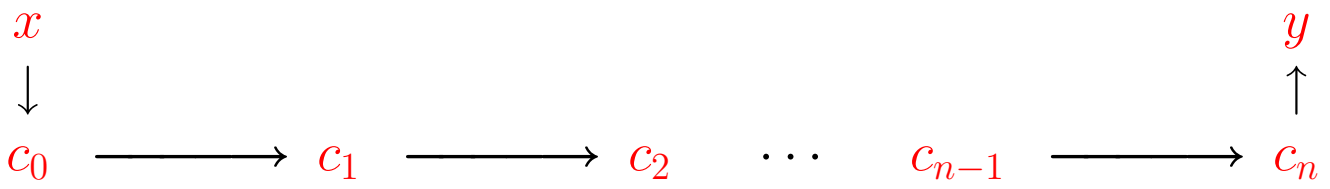
• Une machine programmable M est un quintuplet $(\Sigma, C, \alpha, \omega, I)$, où

Σ, l'alphabet d'entrée-sortie, est un ensemble fini de symboles,
C, est un ensemble, généralement infini, de configurations. Les couples (c_i, c_j) d'éléments de C sont appelés transitions,
α, la fonction d'entrée associe une configuration $\alpha(x)$ à chaque suite finie x de symboles,
ω, la fonction de sortie associe une suite finie $\omega(c_i)$ de symboles à chaque configuration c_i,
I, l'ensemble des instructions possibles est un ensemble d'instructions ; une intruction étant un ensemble de transitions non contradictoires.

• Un programme P pour M est un ensemble d'instructions non contradictoires, prises dans I.

Fonctionnement d'une machine programmable M

• En accord avec le schéma



• pour un programme P et une suite finie x de symboles, on définit

$$orbite(P,\, x) \quad = \begin{cases} \text{la plus longue suite possible } c_0, c_1, c_2, \ldots, \\ \text{avec } c_0 = \alpha(x) \text{ et} \\ \text{chaque } (c_i, c_{i+1}) \text{ élément d'une instruction de } P. \end{cases}$$

$$trace(P,\, x) \quad = \begin{cases} (c_0, c_1)\,(c_1, c_2)\,(c_2, c_3)\ \ldots \\ \text{avec } orbite(P,\, x) = c_0, c_1, c_2, \ldots, \end{cases}$$

$$resultat(P,\, x) \ = \begin{cases} \nearrow, & \text{si } orbite(P,\, x) \text{ est infinie,} \\ \omega(c_n), & \text{si } orbite(P,\, x) \text{ est finie et se termine par } c_n. \end{cases}$$

Simulation d'une machine programmable par une autre (technique)

• Soient M et M' deux machines programmables de même alphabet Σ et soient $\mathcal{P}$ et $\mathcal{P}'$ l'ensemble de leurs programmes.

• **Définition** La transformation de programmes $f : \mathcal{P} \to \mathcal{P}'$ simule M sur M' si, pour tous $P \in \mathcal{P}$ et $x \in \Sigma^\star$,

$$resultat_M(P, x) = resultat_{M'}(f(P), x).$$

• **Définition** La transformation $f : \mathcal{P} \to \mathcal{P}'$, qui simule M sur M', le fait parfaitement si, pour tous $P \in \mathcal{P}$ et $x \in \Sigma^\star$,

$$|trace_M(P, x)| = |trace_{M'}(f(P), x)|.$$

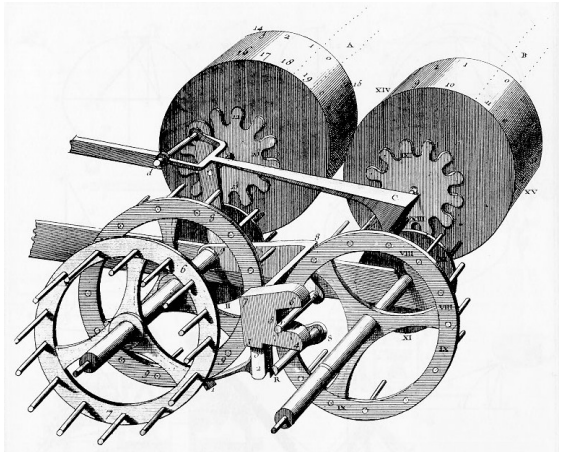
Qu’est-ce qu’un programme universel ?

Programme universel pour traduire

$\Sigma = \{\texttt{␣,0,1,...,9,a,b,...,z}\}$

Programme de traduction	`green␣ideas␣sleeep␣furiously`

⇓



Programme universel

$\begin{cases} \text{instruction 1,} \\ \text{instruction 2,} \\ \dots \\ \text{instruction } n \end{cases}$

⇓

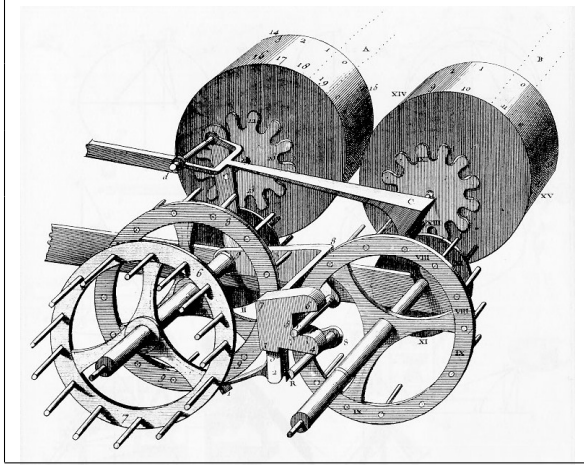
`des␣idees␣vertes␣dorment␣furieusement`

Même programme universel pour multiplier

$\Sigma = \{\sqcup, \texttt{0}, \texttt{1}, \ldots, \texttt{9}, \texttt{a}, \texttt{b}, \ldots, \texttt{z}\}$

Programme de multiplication	`26010x37979721`

$\Downarrow$

	Programme universel $\left\{\begin{array}{l}\text{instruction 1,}\\ \text{instruction 2,}\\ \ldots\\ \text{instruction } n\end{array}\right\}$

$\Downarrow$

`9876543210`

Définition d'un programme universel

Un programme U est universel si, pour tout programme P codé par la suite finie $code(P)$ de symboles et pour toute suite finie x de symboles, on a

$$\boxed{resultat(U,\ code(P)\cdot x) = resultat(P,\ x)}$$

Programme universel tournant sur lui-même

A partir de

$$resultat(U,\ code(P)\cdot x) = resultat(P,\ x),$$

en prenant

$$P = U,$$

on obtient

$$\boxed{resultat(U,\ code(U)\cdot x) = resultat(U,\ x)}$$

Programme universel tournant n fois sur lui-même

A partir de

$$resultat(U,\ code(U)\cdot z) = resultat(P,\ z),$$

en prenant

$$z = code(U)^n \cdot x \qquad \text{et} \qquad P = U,$$

on obtient

$$resultat(U,\ code(U)^{n+1}\cdot x) = resultat(U,\ code(U)^n \cdot x).$$

Il s'ensuit que

$$\boxed{resultat(U,\ code(U)^n \cdot x) = resultat(U,\ x)}$$

Exemple de programme universel tournant sur lui-même

Programme de traduction

1. Donner à i la valeur 0.
2. Augmenter i de 1.
3. Si i est plus grand que le nombre de mots à traduire, aller à 6.
4. Traduire le $i^{\text{ème}}$ mot.
5. Aller à 2.
6. S'arrêter.

Entrée

`green␣ideas␣sleep␣furiously`

Programme universel

1. Donner à n la valeur 0.
2. Augmenter n de 1.
3. Chercher l'instruction numéro n au début de la donnée.
4. Si l'instruction trouvée est une instruction d'arrêt, aller à 12.
5. Si l'instruction trouvée est de la forme *« si test alors aller à p »*, aller à 8.
6. Exécuter l'instruction trouvée.
7. Aller à 2.
8. Effectuer le test demandé par l'instruction trouvée.
9. Si le test n'est pas vérifié, aller à 2.
10. Donner à n la valeur p.
11. Aller à 3.
12. S'arrêter.

Entrée plus compliquée

1. Donner à i la valeur 0.
2. Augmenter i de 1.
3. Si i est plus grand que …, aller à 6.
4. Traduire le $i^{\text{ème}}$ mot.
5. Aller à 2.
6. S'arrêter.

```
green␣ide
as␣sleep␣
furiously
```

Même programme universel

1. Donner à n la valeur 0.
2. Augmenter n de 1.
3. Chercher l'instruction numéro n au début de la donnée.
4. Si l'instruction trouvée est une instruction d'arrêt, aller à 12.
5. Si l'instruction trouvée est de la forme *« si test alors aller à p »*, aller à 8.
6. Exécuter l'instruction trouvée.
7. Aller à 2.
8. Effectuer le test demandé par l'instruction trouvée.
9. Si le test n'est pas vérifié, aller à 2.
10. Donner à n la valeur p.
11. Aller à 3.
12. S'arrêter.

Entrée encore plus compliquée

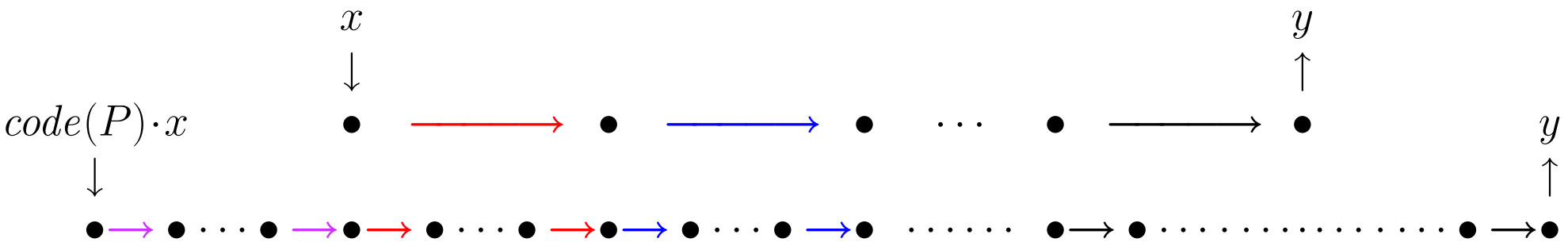
1. Donner . . .
2. Augmen. . . .
3. Chercher . . .
4. Si . . .
5. Si . . .
6. Exécuter . . .
7. Aller à 2.
8. Effectuer . . .
9. Si le test . . .
10. Donner . . .
11. Aller à 3.
12. S'arrêter.

1. Donner à i la valeur 0.
2. Augmenter i de 1.
3. Si i est plus grand que . . ., aller à 6.
4. Traduire le $i^{\text{ème}}$ mot.
5. Aller à 2.
6. S'arrêter.

```
green␣ide
as␣sleep␣
furiously
```

Comment évaluer la complexité d’un programme universel ?

Perte d'efficacité d'un programme universel



Différents types de complexités

Complexité qui dépend de 2 paramètres,

$$\lambda(P,\, x) = \frac{|trace(U,\, code(P)\cdot x)|}{|trace(P,\, x)|}$$

Complexité qui dépend de 1 paramètre,

$$\lambda(x) = \frac{|trace(U,\, code(U)\cdot x)|}{|trace(U,\, x)|}$$

Coefficient d'introspection, une complexité susceptible de dépendre de 0 paramètre,

$$\boxed{\lambda = \lim_{n\to\infty} \frac{|trace(U,\, code(U)^{n+1}\cdot x)|}{|trace(U,\, code(U)^{n}\cdot x)|}}$$

Passage d'une machine universelle à une autre (technique)

• **Hypothèses** Soient M et M' deux machines programmables, de même alphabet Σ, pour lesquelles il existe :
  – une transformation de programme f qui simule parfaitement M sur M',
  – une transformation de programme f' qui simule M' sur M.

• **Propriété** Si le couple programme-codage $(U,\ code)$
  – est universel pour M,
  – a pour coefficient d'introspection λ,
  – est tel que $f'(f(U)) = U$

alors le couple $(U',\ code')$, avec $U' = f(U)$ et $code'(P') = code(f'(P'))$,
  – est universel pour M',
  – a pour coefficient d'introspection λ.

Passage d'une machine universelle à une autre, suite (technique)

• Démontrons que sous les conditions mentionnées, $(U',\ code')$ est universel

$resultat_{M'}(P',\ x)\ =$ (du fait que f' simule M' sur M)
$resultat_{M}(f'(P'),\ x)\ =$ (du fait que $(U,\ code)$ est universel pour M)
$resultat_{M}(U,\ code(f'(P'))\cdot x)\ =$ (du fait que f simule M sur M')
$resultat_{M'}(f(U),\ code(f'(P'))\cdot x)\ =$
$resultat_{M'}(U',\ code'(P')\cdot x)$

• Démontrons que sous les conditions mentionnées, $(U',\ code')$ a λ pour coefficient d'introspection

$|trace_{M}(U,\ code(U)^n\cdot x)|\ =$ (du fait que f simule parfaitement M sur M')
$|trace_{M'}(f(U),\ code(U)^n\cdot x)|\ =$ (du fait que $U = f'(f(U))$)
$|trace_{M'}(f(U),\ code(f'(f(U)))^n\cdot x)|\ =$
$|trace_{M'}(U',\ code'(U')^n\cdot x)|$

et donc

$$\lim_{n\to\infty}\frac{|trace_{M'}(U',\ code'(U')^{n+1}\cdot x)|}{|trace_{M'}(U',\ code'(U')^{n}\cdot x)|} = \lim_{n\to\infty}\frac{|trace_{M}(U,\ code(U)^{n+1}\cdot x)|}{|trace_{M}(U,\ code(U)^{n}\cdot x)|} = \lambda.$$

Existence et calcul du coefficient d’introspection

Hypothèse de coloriage sur un programme universel U

• Il est possible d'attribuer à chaque transition t de U une couleur $couleur(t)$, (prise dan un ensemble fini), telle que pour tout couple de traces de la forme

$$\begin{bmatrix} r_1 \cdots r_m \\ s_1 \cdots s_n \end{bmatrix} = \begin{bmatrix} trace(U,\ x) \\ trace(U,\ code(U) \cdot x)) \end{bmatrix}$$

on ait

$$couleur(s_1) \cdots couleur(s_n) = debut \cdot \varphi(couleur(r_1)) \cdots \varphi(couleur(r_m))$$

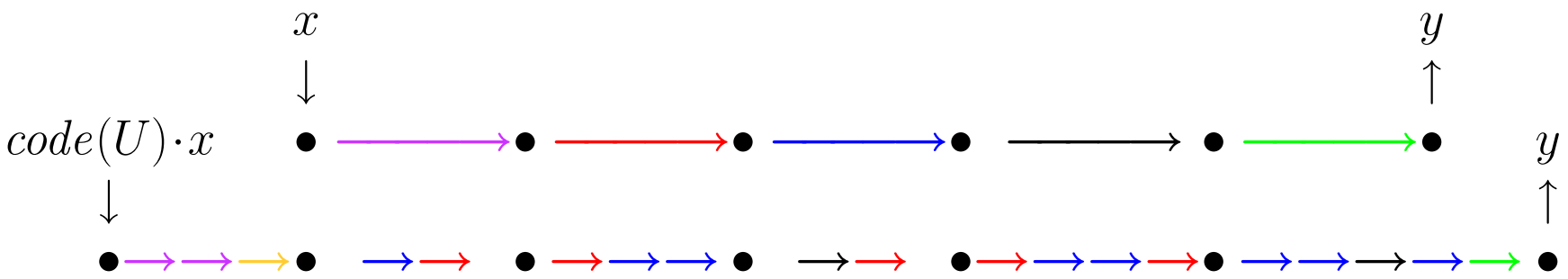
• où $debut$ est toujours la même suite finie de couleurs

• et où chaque $\varphi(k)$ est une suite finie de couleurs ne dépendant que de la couleur k.

Exemple de coloriage

- Ensemble de couleurs $\{\rightarrow, \rightarrow, \rightarrow, \rightarrow, \rightarrow, \rightarrow\}$,

- coloriage d'un couple de traces de la forme $\begin{bmatrix} trace(U,\ x) \\ trace(U,\ code(U)\cdot x)) \end{bmatrix}$



- valeur de $debut$

$$debut = \rightarrow\rightarrow\rightarrow,$$

- valeurs de $\varphi(k)$

$$\varphi(\rightarrow) = \rightarrow\rightarrow \qquad \varphi(\rightarrow) = \rightarrow\rightarrow\rightarrow \qquad \varphi(\rightarrow) = \rightarrow\rightarrow\rightarrow\rightarrow$$
$$\varphi(\rightarrow) = \rightarrow\rightarrow \qquad \varphi(\rightarrow) = \rightarrow\rightarrow \qquad \varphi(\rightarrow) = \rightarrow\rightarrow\rightarrow\rightarrow\rightarrow$$

Vecteur B pour coder $debut$

• On introduit la matrice colonne

$$B = \begin{bmatrix} b_1 \\ \vdots \\ b_n \end{bmatrix}, \qquad b_i = \text{nombre d'occurrences de la couleur no } i \text{ dans } debut.$$

Par exemple pour l'ensemble de couleurs

$$\mathbf{K} = \{\rightarrow, \rightarrow, \rightarrow, \rightarrow, \rightarrow, \rightarrow\}$$

et

$$debut = \rightarrow\rightarrow\rightarrow,$$

on obtient

$$B = \begin{matrix} \\ \blacksquare \\ \blacksquare \\ \blacksquare \\ \blacksquare \\ \blacksquare \\ \blacksquare \end{matrix} \begin{bmatrix} 2 \\ 1 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}$$

Matrice A pour coder φ

• On introduit la matrice carrée

$$A = \begin{bmatrix} a_{11} \ldots a_{1n} \\ \vdots \quad\quad \vdots \\ a_{1n} \ldots a_{nn} \end{bmatrix}, \qquad a_{ij} = \text{nombre d'occurrences de la couleur no } i \text{ dans } \varphi(j).$$

Par exemple pour l'ensemble de couleurs

$$\mathbf{K} = \{\rightarrow, \rightarrow, \rightarrow, \rightarrow, \rightarrow, \rightarrow\}$$

et

$$\begin{array}{llll} \varphi(\rightarrow) = \rightarrow\rightarrow & \varphi(\rightarrow) = \rightarrow\rightarrow\rightarrow & \varphi(\rightarrow) = \rightarrow\rightarrow\rightarrow\rightarrow & \\ \varphi(\rightarrow) = \rightarrow\rightarrow & \varphi(\rightarrow) = \rightarrow\rightarrow & \varphi(\rightarrow) = \rightarrow\rightarrow\rightarrow\rightarrow\rightarrow & , \end{array}$$

on obtient

$$A = \begin{array}{c} \\ \blacksquare \\ \blacksquare \\ \blacksquare \\ \blacksquare \\ \blacksquare \\ \blacksquare \end{array} \begin{array}{c} \begin{array}{cccccc} \blacksquare & \blacksquare & \blacksquare & \blacksquare & \blacksquare & \blacksquare \end{array} \\ \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 1 & 1 & 2 & 0 \\ 1 & 0 & 2 & 0 & 2 & 3 \\ 0 & 2 & 0 & 1 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} \end{array}$$

Propriétés de A et B (technique)

• **Propriété** Pour tout $x \in \Sigma^\star$ avec $resultat(U,\, x) \neq \nearrow$, en introduisant le vecteur

$$X^{(k)} = \begin{bmatrix} x_1^{(k)} \\ \vdots \\ x_n^{(k)} \end{bmatrix}, \qquad x_i^{(k)} = \begin{cases} \text{nombre d'occurrences de la couleur } i \text{ dans} \\ couleur^\star(trace(U,\, code(U)^k \cdot x)), \end{cases}$$

on a

$$\boxed{X^{(k+1)} = AX^{(k)} + B}$$

• **Remarque**

$$|trace(U,\, code(U)^k \cdot x)| = ||X^{(k)}||, \quad \text{donc} \quad \frac{|trace(U,\, code(U)^{k+1} \cdot x)|}{|trace(U,\, code(U)^k \cdot x)|} = \frac{||X^{(k+1)}||}{||X^{(k)}||}.$$

Théorème

Sous l'hypothèse de coloriage, si la matrice A admet une valeur propre réelle λ, dont le degré de multiplicité égal à 1 et dont la valeur est strictement supérieure à 1 et strictement supérieure au plus grand module λ' des autres valeurs propres de A, alors une et une seule des situations suivantes se présente :

• soit, pour tout réel α tel que $\lambda' < \alpha < \lambda$, la suite de vecteurs

$$X_0, X_1, X_2, \ldots \qquad \text{avec} \qquad X_0 = B, \;\; X_{n+1} = \frac{1}{\alpha} A X_n,$$

converge vers le vecteur nul,

• soit, pour tout réel α tel que $\lambda' < \alpha < \lambda$ la suite de vecteurs

$$X_0, X_1, X_2, \ldots \qquad \text{avec} \qquad X_0 = B, \;\; X_{n+1} = \frac{1}{\alpha} A X_n,$$

diverge.

Dans ce dernier cas, pour toute suite finie x de symboles telle que $trace(U, x) \neq \nearrow$, on a

$$\lim_{k \to \infty} \frac{|trace(U,\ code(U)^{k+1} \cdot x)|}{|trace(U,\ code(U)^{k} \cdot x)|} = \lambda$$

Exemple d'utilisation du théorème

- Soit notre matrice et notre vecteur

$$A = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 1 & 1 & 2 & 0 \\ 1 & 0 & 2 & 0 & 2 & 3 \\ 0 & 2 & 0 & 1 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} \qquad B = \begin{bmatrix} 2 \\ 1 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}$$

- Les valeurs propres de A sont

$$2,732 \quad 1,000 \quad -1,000 \quad -0,732 \quad 0,000 \quad 0,000.$$

- En prenant $\alpha = 1,5$, on trouve pour $X_0, X_1, X_2, X_3, X_4, X_5, X_7, X_8, X_9$,

$$\begin{bmatrix} 2,00 \\ 1,00 \\ 0,00 \\ 0,00 \\ 0,00 \\ 0,00 \end{bmatrix}, \begin{bmatrix} 0,00 \\ 0,00 \\ 1,33 \\ 1,33 \\ 1,33 \\ 0,00 \end{bmatrix}, \begin{bmatrix} 0,00 \\ 0,00 \\ 3,55 \\ 3,55 \\ 0,88 \\ 0,00 \end{bmatrix}, \begin{bmatrix} 0,00 \\ 0,00 \\ 5,92 \\ 5,92 \\ 2,37 \\ 0,00 \end{bmatrix}, \begin{bmatrix} 0,00 \\ 0,00 \\ 11,06 \\ 11,06 \\ 3,95 \\ 0,00 \end{bmatrix}, \begin{bmatrix} 0,00 \\ 0,00 \\ 20,01 \\ 20,01 \\ 7,37 \\ 0,00 \end{bmatrix}, \begin{bmatrix} 0,00 \\ 0,00 \\ 36,52 \\ 36,52 \\ 13,34 \\ 0,00 \end{bmatrix}, \begin{bmatrix} 0,00 \\ 0,00 \\ 66,48 \\ 66,48 \\ 24,34 \\ 0,00 \end{bmatrix}, \begin{bmatrix} 0,00 \\ 0,00 \\ 121,11 \\ 121,11 \\ 44,32 \\ 0,00 \end{bmatrix}, \begin{bmatrix} 0,00 \\ 0,00 \\ 220,58 \\ 220,58 \\ 80,74 \\ 0,00 \end{bmatrix}.$$

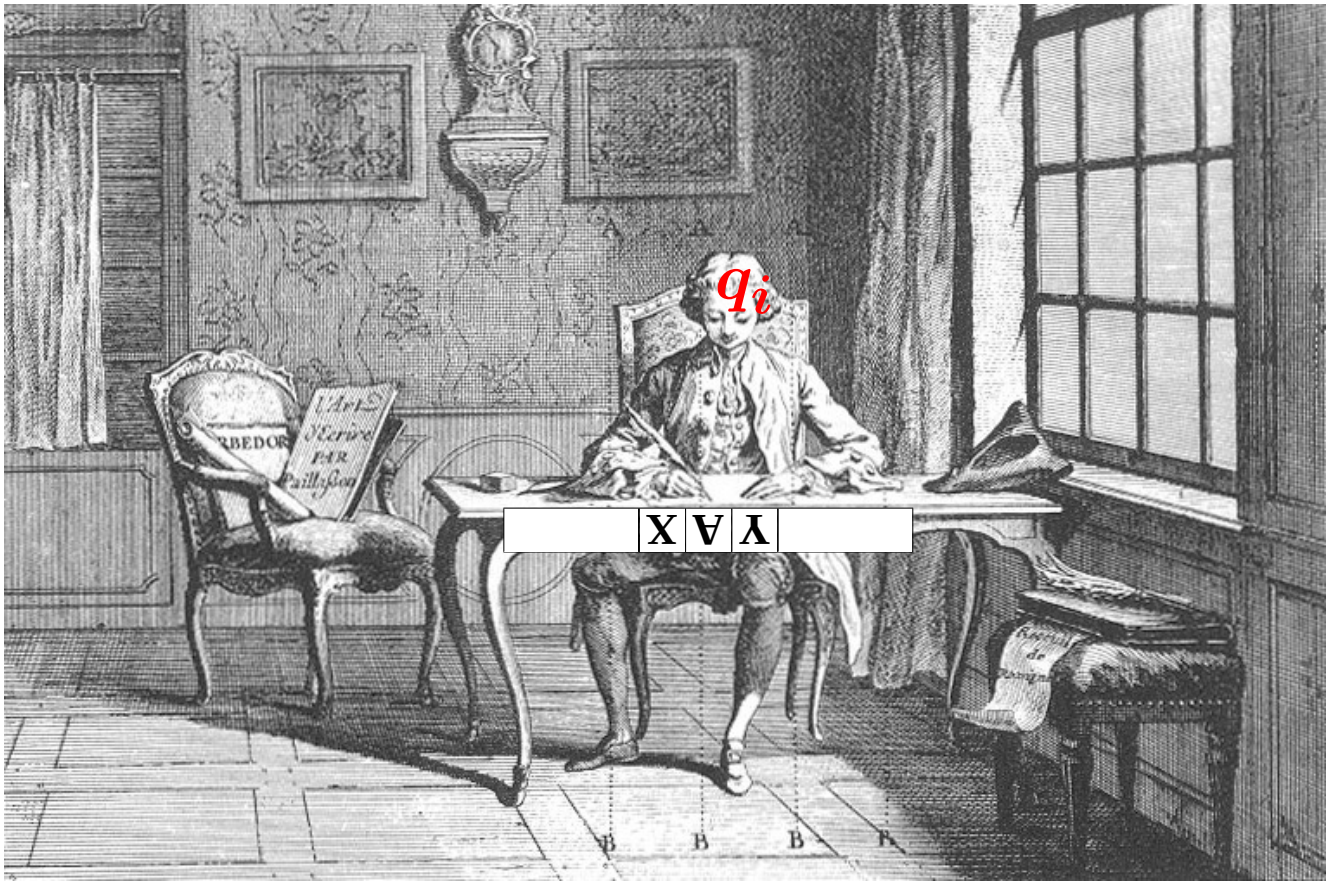
- Le coefficient d'introspection existe donc et est $\lambda = 2,732$.

Machine de Turing

Alan Turing

Alan M. Turing, «On computable numbers with an application to the Entsheidungsproblem», *Proc. London Math. Society*, 2, 42, pp. 230–265, 1936.

Configuration d’avant



Instruction exécutée

$$[q_i, \mathrm{A}, \mathrm{B}, D, q_j]$$

Configuration d’après

Instruction exécutée

$$[q_i, \mathrm{A}, \mathrm{B}, G, q_j]$$

Configuration d’après

Exemple de programme

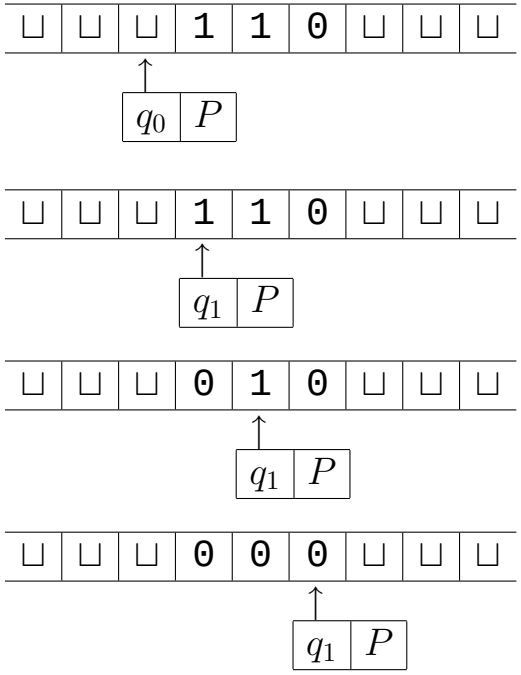
• Avec le programme

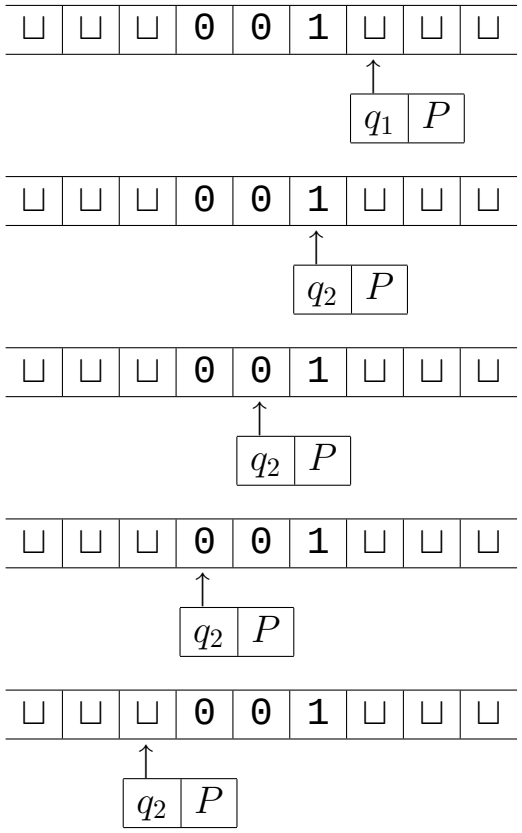
$$P = \left\{ \begin{array}{l} [q_0, \sqcup, \sqcup, D, q_1], \\ [q_1, \mathtt{0}, \mathtt{1}, D, q_1], \\ [q_1, \mathtt{1}, \mathtt{0}, D, q_1], \\ [q_1, \sqcup, \sqcup, G, q_2], \\ [q_2, \mathtt{0}, \mathtt{0}, G, q_2], \\ [q_2, \mathtt{1}, \mathtt{1}, G, q_2] \end{array} \right\},$$

• à partir de l'entrée

$$\mathtt{110},$$

• la machine passe à travers la suite de configurations



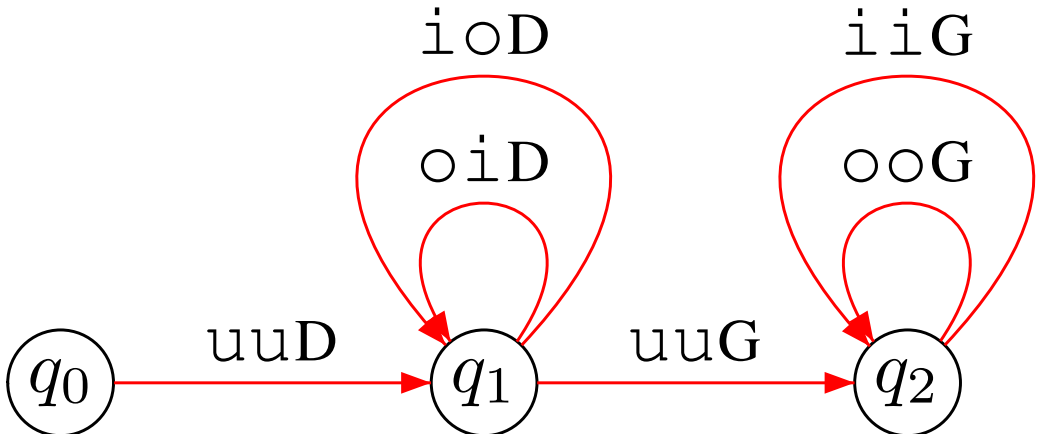


• et produit le résultat

$$\mathtt{001}$$

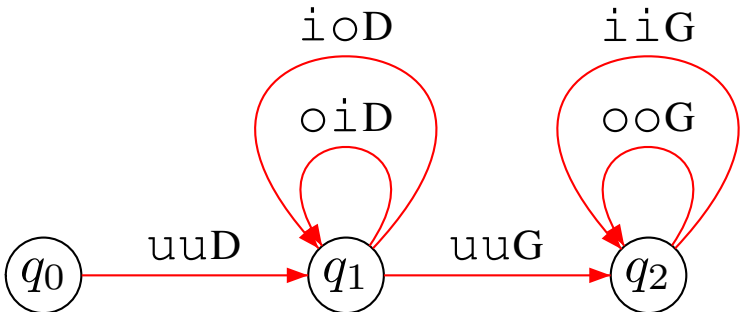
Schématisation d'un programme

$$\left\{\begin{array}{l}
[q_0, \sqcup, \sqcup, +1, q_1],\\
[q_1, \texttt{0}, \texttt{1}, +1, q_1],\\
[q_1, \texttt{1}, \texttt{0}, +1, q_1],\\
[q_1, \sqcup, \sqcup, -1, q_2],\\
[q_2, \texttt{0}, \texttt{0}, +1, q_2],\\
[q_2, \texttt{1}, \texttt{1}, +1, q_2]
\end{array}\right\}$$



Exemple de programme, revu

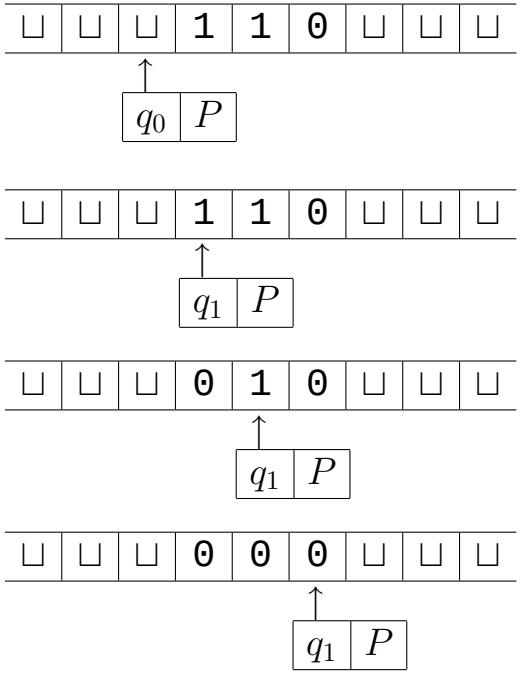
• Avec le programme P

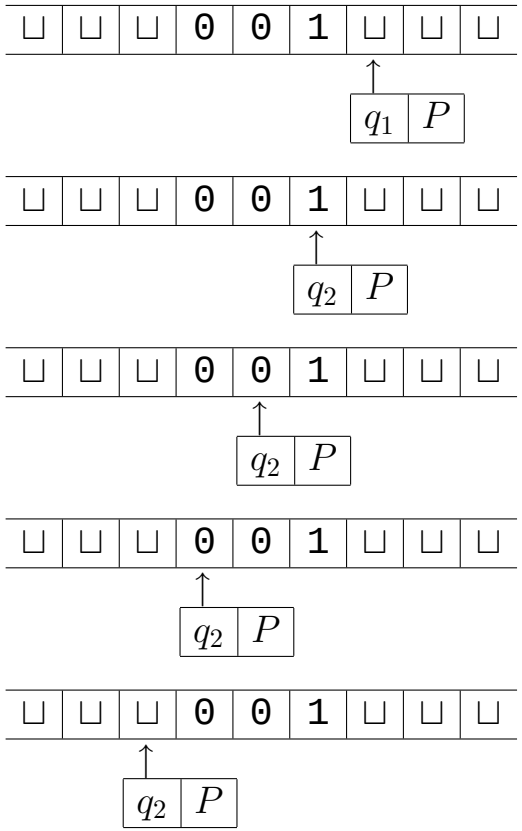


• à partir de l'entrée

`110`,

• la machine passe à travers la suite de configurations





• et produit le résultat

`001`

Machine de Turing avec polarité

Exemple de programme pour machine avec polarité

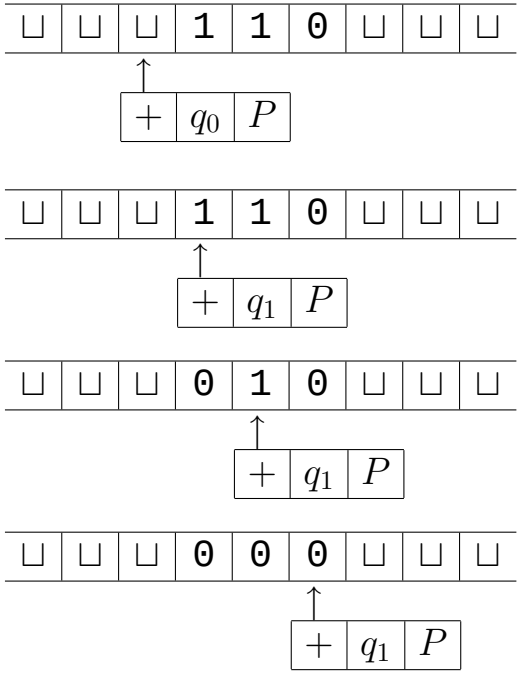
• Avec le programme

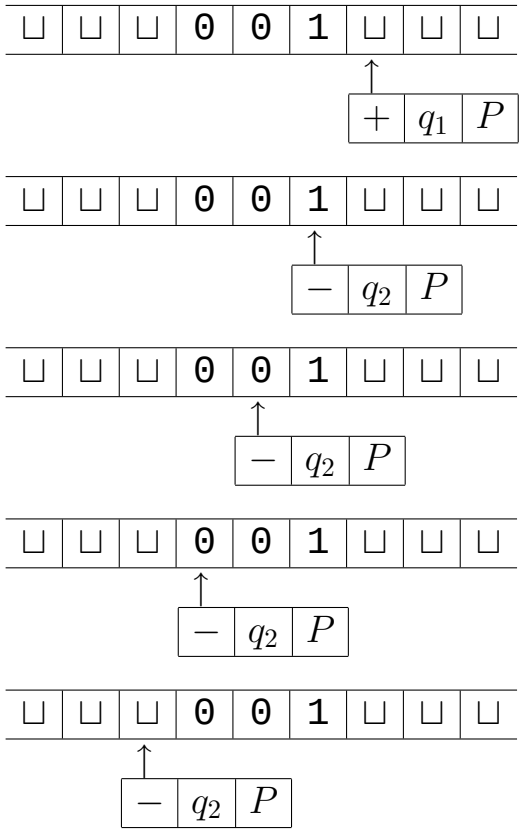
$$P = \left\{ \begin{matrix} [q_1, \mathtt{0}, \mathtt{1}, +1,\ q_1], \\ [q_1, \mathtt{1}, \mathtt{0}, +1,\ q_1], \\ [q_1, \sqcup, \sqcup, -1,\ q_2], \\ [q_2, \mathtt{0}, \mathtt{0}, +1,\ q_2], \\ [q_2, \mathtt{1}, \mathtt{1}, +1,\ q_2] \end{matrix} \right\},$$

• à partir de l'entrée

$$\mathtt{110},$$

• la machine passe à travers la suite de configurations



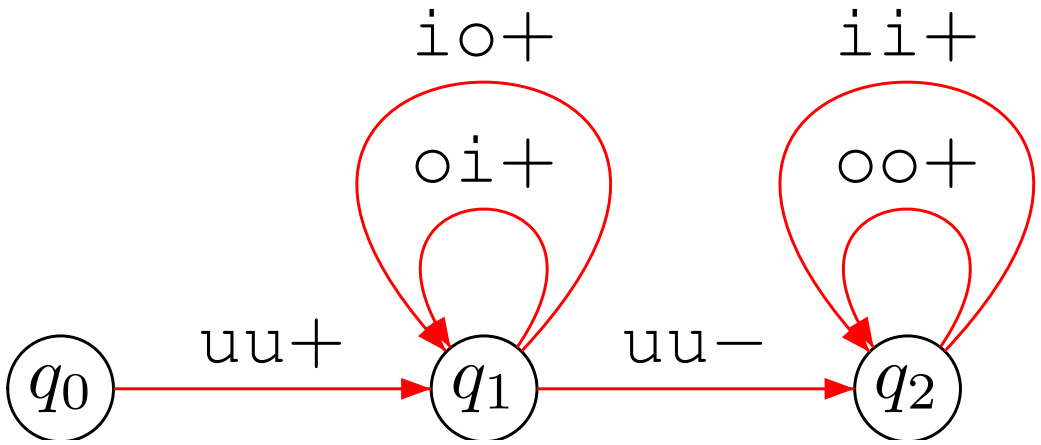


• et produit le résultat

$$\mathtt{001}$$

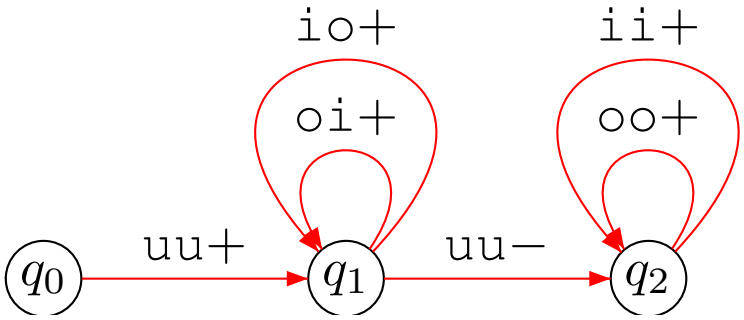
Schématisation d'un programme pour machine avec polarité

$$\left\{\begin{array}{l}[q_0, \sqcup, \sqcup, +1, q_1],\\ [q_1, \texttt{0}, \texttt{1}, +1, q_1],\\ [q_1, \texttt{1}, \texttt{0}, +1, q_1],\\ [q_1, \sqcup, \sqcup, -1, q_2],\\ [q_2, \texttt{0}, \texttt{0}, +1, q_2],\\ [q_2, \texttt{1}, \texttt{1}, +1, q_2]\end{array}\right\}$$



Exemple de programme pour machine avec polarité, revu

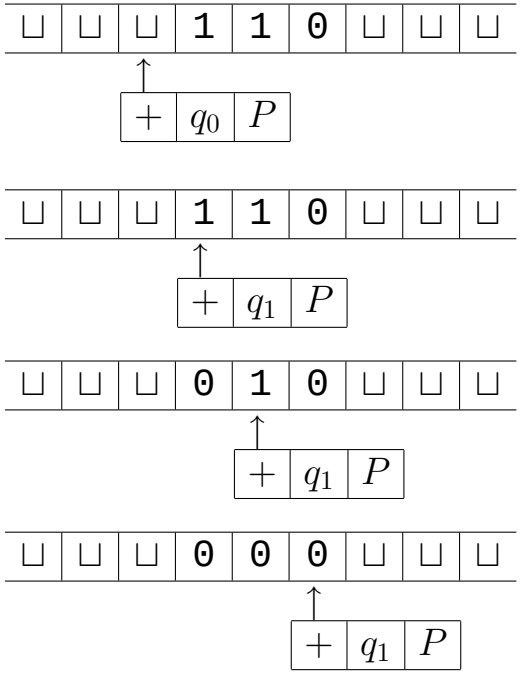
- Avec le programme P

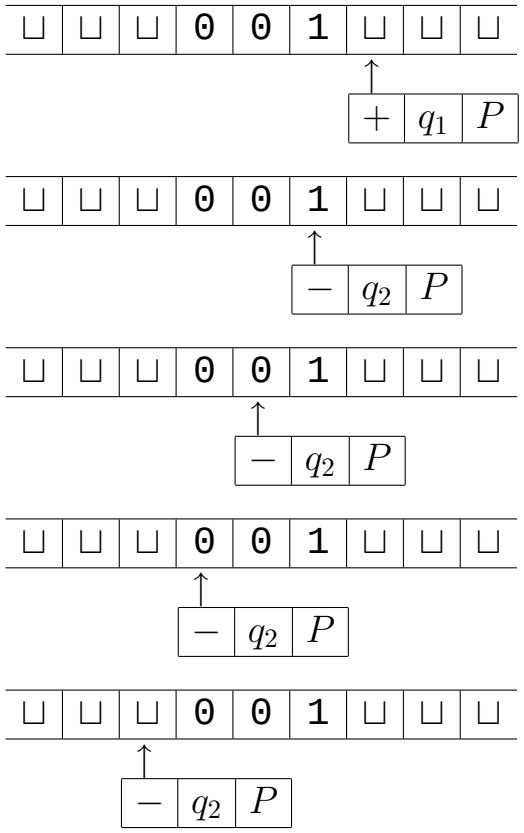


- à partir de l'entrée

`110`,

- la machine passe à travers la suite de configurations





- et produit le résultat

`001`

Lien entre machines programmables classiques et avec polarité

• Soit $M^{\pm}$ la machine programmable de Turing, avec polarité, sur l'alphabet Σ et soit M la machine programmable de Turing classique sur le même alphabet Σ.

• **Propriété** Il existe des transformations de programmes qui simulent parfaitement $M^{\pm}$ sur M et M sur $M^{\pm}$.

• **Conséquence** Si $(U^{\pm},\ code^{\pm})$ est un couple programme-codage, universel pour $M^{\pm}$, ayant λ pour coefficient d'introspection, alors il existe un couple programme-codage $(U,\ code)$, universel pour M, ayant aussi λ pour coefficient d'introspection.

Notre machine de Turing universelle à trois symboles, blanc non compris

Complexité de notre machine de Turing universelle

Complexité générale, en posant $p = code(P)$, pour toute suite finie x de symboles,

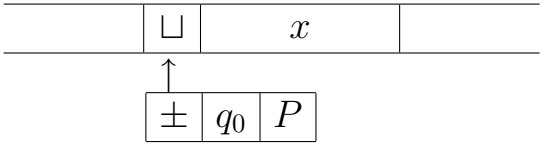
$$\lambda(P, x) = \frac{|trace(U,\, px)|}{|trace(P,\, x)|} = \mathcal{O}(|p|log(|p|))$$

Coefficient d'introspection,

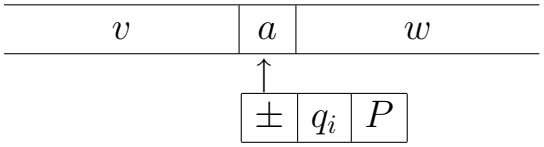
$$\lambda = \lim_{n\to\infty} \frac{|trace(U,\, code(U)^{n+1} \cdot x)|}{|trace(U,\, code(U)^{n} \cdot x)|} = 3672,983961$$

Configurations générales

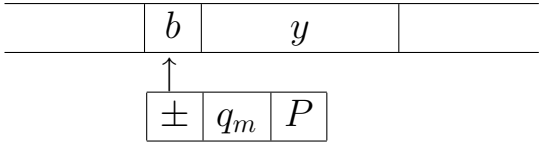
- Configuration initiale



- Configuration courante

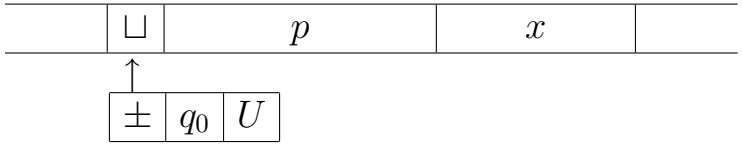


- Configuration finale

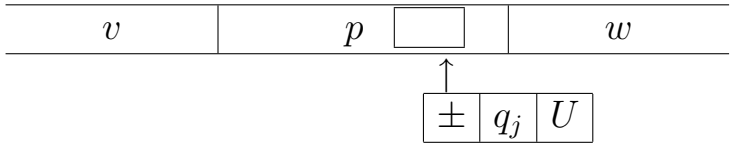


Configurations universelles associées

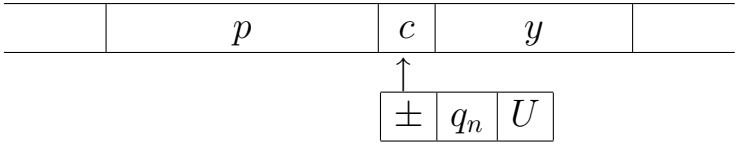
- Configuration initiale



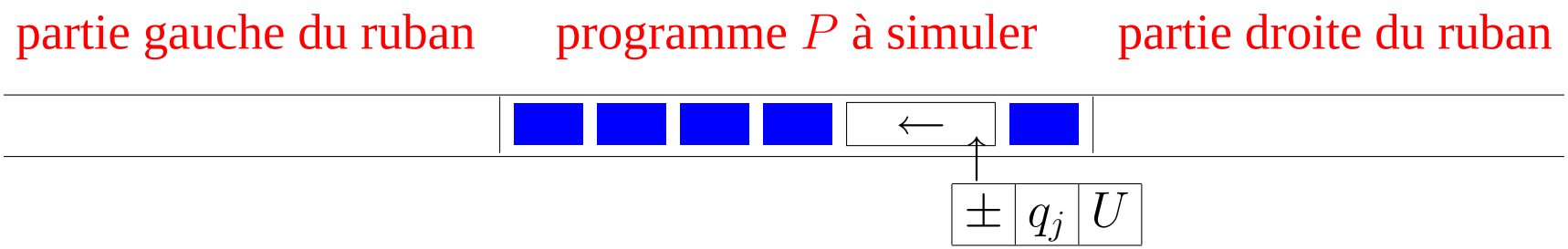
- Configuration courante



- Configuration finale



Détails d'une configuation universelle



Architecture de notre programme universel

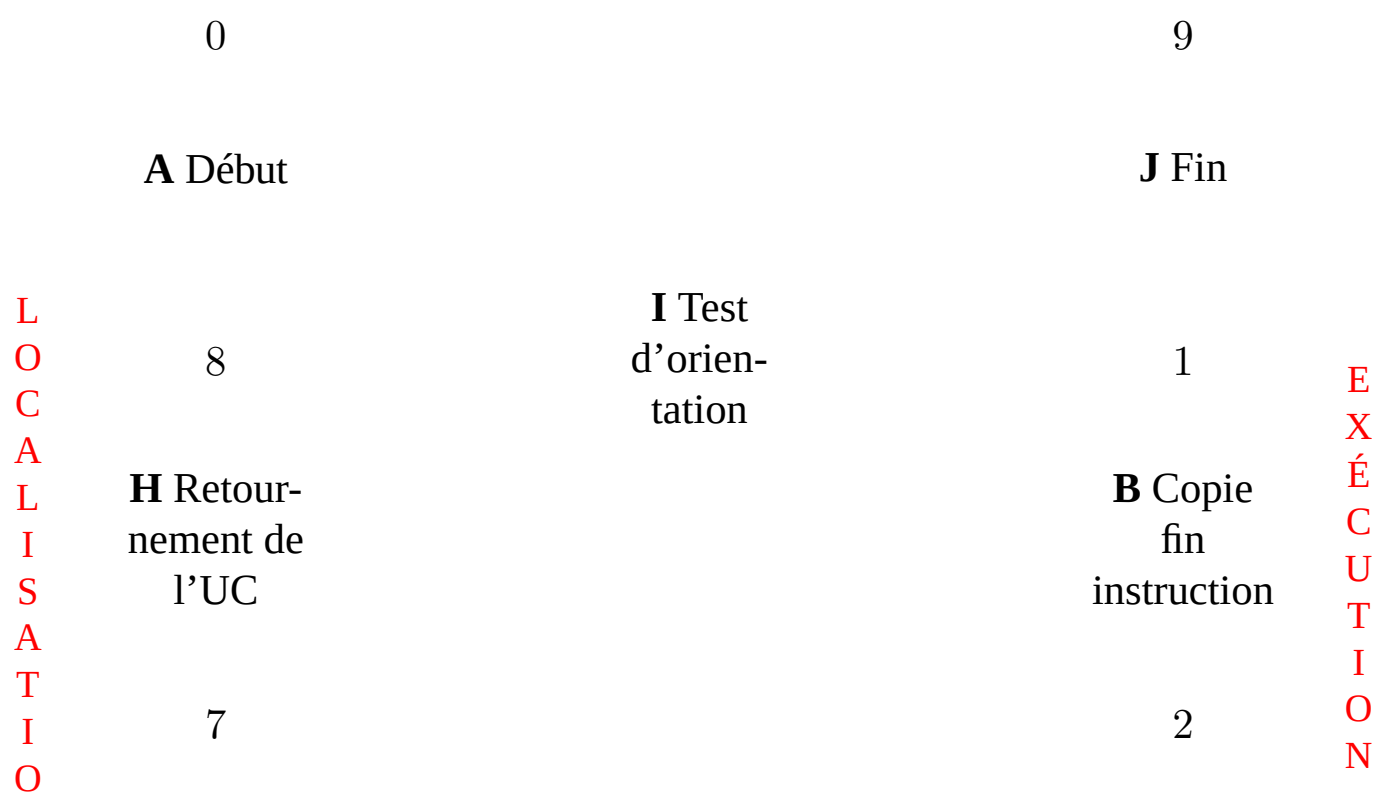
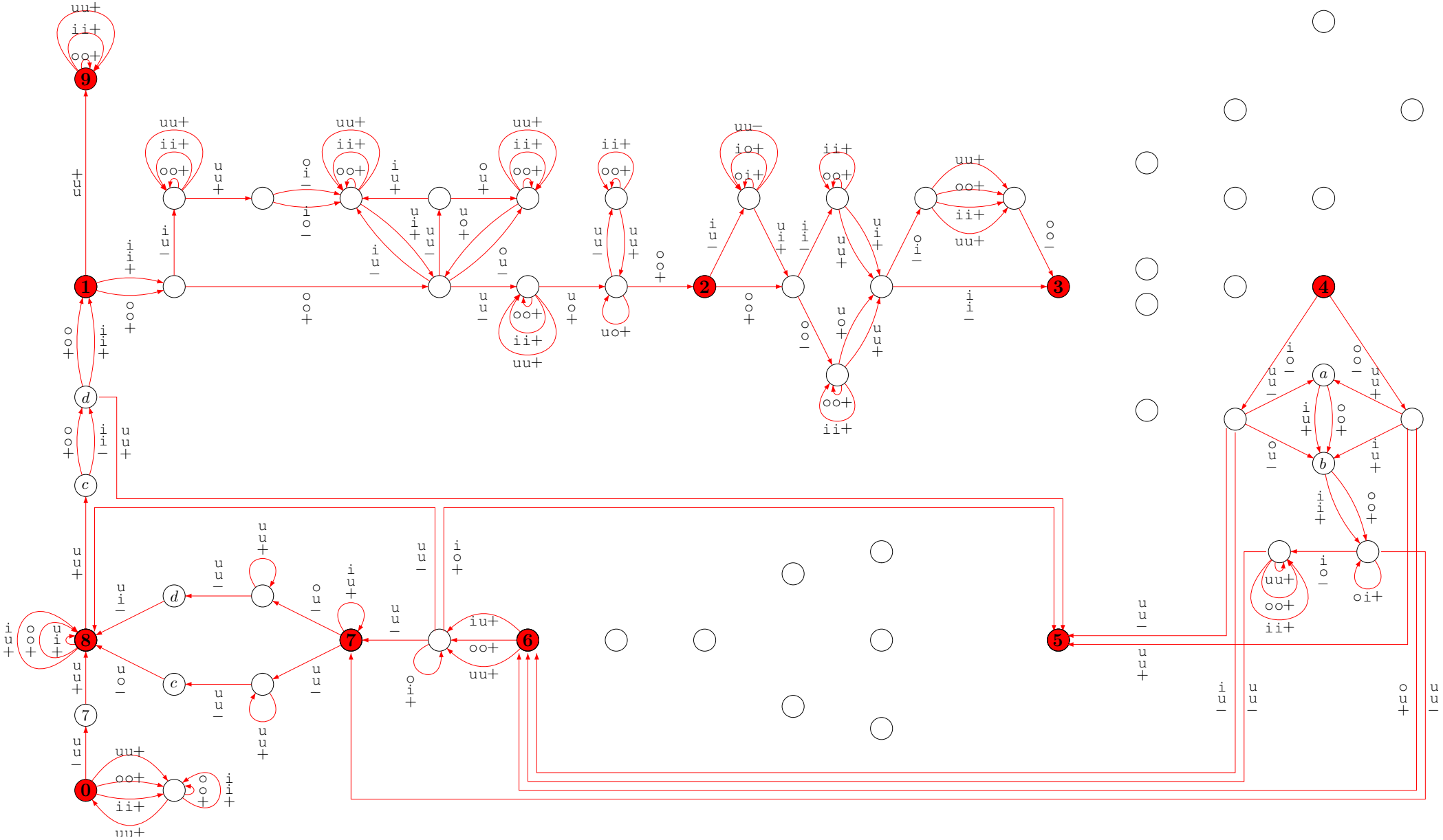


Schéma de notre programme universel

Les modules D et F faisant l'objet d'un exercice, seuls leurs états ont été dessinés.



Notre programme universel à trois symboles, blanc non compris

$\Sigma = \{o, i, z\}$, symbole blanc noté u, 54 états, 184 instructions, $|code(U)| = 1552$.

```
# A DEBUT
[X0,u,z,A1,+1],    [X0,o,o,A1,+1],   [X0,i,i,A1,+1],   [X0,z,u,X7,-1],
                   [A1,o,o,A1,+1],   [A1,i,i,A1,+1],   [A1,z,z,X0,+1],
                                                       [X7,z,z,X8,+1],
# B COPIE DE LA FIN DE L'INSTRUCTION
# Enregistrement de la direction et du nb d'instruction à sauter
                   [X1,o,o,B1,+1],   [X1,i,i,B1,+1],
                   [B1,o,o,B5,+1],   [B1,i,u,B2,-1],
                   [B2,o,o,B2,+1],   [B2,i,i,B2,+1],   [B2,z,z,B3,+1],
                   [B3,o,i,B4i,-1],  [B3,i,o,B4i,-1],
[B4o,u,o,B5,+1],   [B4o,o,o,B4o,+1], [B4o,i,i,B4o,+1], [B4o,z,z,B4o,+1],
[B4i,u,i,B5,+1],   [B4i,o,o,B4i,+1], [B4i,i,i,B4i,+1], [B4i,z,z,B4i,+1],
[B5,u,u,B6,-1],    [B5,o,u,B4o,-1],  [B5,i,u,B4i,-1],  [B5,z,z,B7,-1],
                   [B6,o,o,B4o,+1],  [B6,i,i,B4i,+1],
# Remplacement des u restants par des o
[B7,u,o,B9,+1],    [B7,o,o,B7,+1],   [B7,i,i,B7,+1],   [B7,z,z,B7,+1],
[B9,u,o,B9,+1],    [B9,o,o,X2,+1],                     [B9,z,z,B10,-1],
                   [B10,o,o,B10,+1], [B10,i,i,B10,+1], [B10,z,z,B9,+1],

# C COPIE DU DEBUT DE L'INSTRUCTION
# Création du symbole à écrire
                   [X2,o,o,C2,+1],   [X2,i,u,C1,-1],
[C1,u,i,C2,+1],    [C1,o,i,C1,+1],   [C1,i,o,C1,+1],   [C1,z,u,C1,-1],
                   [C2,o,o,C3o,-1],  [C2,i,i,C3i,-1],
[C3o,u,o,C4,+1],   [C3o,o,o,C3o,+1], [C3o,i,i,C3o,+1], [C3o,z,u,C4,+1],
[C3i,u,z,C4,+1],   [C3i,o,o,C3i,+1], [C3i,i,i,C3i,+1], [C3i,z,i,C4,+1],
# Prise en compte de la direction
                   [C4,o,i,C5,-1],   [C4,i,i,X3,-1],
[C5,u,u,C6,+1],    [C5,o,o,C6,+1],   [C5,i,i,C6,+1],   [C5,z,z,C6,+1],
                   [C6,o,o,X3,-1],
```

Notre programme universel, suite 1


```
# D ECRITURE, MOUVEMENT ET LECTURE
# Ecriture et lecture

          # PARTIE A REFAIRE EN EXERCICE


# Mouvement

          # PARTIE A REFAIRE EN EXERCICE



# E MISE A JOUR DE L'UNITE CENTRALE
# Début de la mise à jour
                    [X4,o,o,E1b,-1],  [X4,i,o,E1a,-1],
[E1a,u,z,E2a,-1],  [E1a,o,z,E2b,-1], [E1a,i,z,X6,-1],  [E1a,z,z,X5,-1],
[E1b,u,z,X5,+1],   [E1b,o,z,X6,+1],  [E1b,i,z,E2b,+1], [E1b,z,z,E2a,+1],
# Fin de la mise à jour
                    [E2a,o,o,E2b,+1], [E2a,i,u,E2b,+1],
                    [E2b,o,o,E4,+1],  [E2b,i,i,E4,+1],
                    [E4,o,i,E4,+1],   [E4,i,o,E5,-1],   [E4,z,z,X7,-1],
[E5,u,u,E5,+1],    [E5,o,o,E5,+1],   [E5,i,i,E5,+1],   [E5,z,z,X6,-1],
```

Notre programme universel, suite 2


```
# F DEPLACEMENT DE L'UNITE CENTRALE D'UNE INSTRUCTION


          # PARTIE A REFAIRE EN EXERCICE


# G DECREMENTATION DE L'UNITE CENTRALE, REMPLACEMENT D'UN i PAR u
[X6,u,u,G1,+1],    [X6,o,o,G1,+1],   [X6,i,u,G1,+1],
[G1,u,u,X7,-1],    [G1,o,i,G1,+1],   [G1,i,o,X5,+1],   [G1,z,z,X8,-1],

# H RETOURNEMENT DE L'UNITE CENTRALE APRES REMISE A BLANC
[X7,u,u,E2a,-1],   [X7,o,u,E2b,-1],  [X7,i,u,X7,+1],
[E2a,u,u,E2a,+1],                                    [E2a,z,z,I1,-1],
[E2b,u,u,E2b,+1],                                    [E2b,z,z,I2,-1],
[I1,u,o,X8,-1],
[I2,u,i,X8,-1],

# I TEST D'ORIENTATION DE L'INSTRUCTION APRES REMISE A BLANC DE l'UC
[X8,u,i,X8,+1],    [X8,o,o,X8,+1],   [X8,i,u,X8,+1],   [X8,z,z,I1,+1],
                   [I1,o,o,I2,+1],   [I1,i,i,I2,-1],
                   [I2,o,o,X1,+1],   [I2,i,i,X1,+1],   [I2,z,z,X5,+1],

# J FIN DU PROGRAMME
                                                     [X1,z,z,X9,+1],
                   [X9,o,o,X9,+1],   [X9,i,i,X9,+1],   [X9,z,z,X9,+1]];
```

Machine à adressage indirect

Machine à adressage indirect

• L'alphabet d'entrée-sortie est $\Sigma = \{o, i, z\}$.

• La machine est constituée d'une infinité de régistres R_i disposés comme suit :

R_0	R_1	R_2	R_3	R_4	

Chacun de ces registres contient un entier positif ou nul, aussi grand que l'on veut.

• Les instructions possibles sont

$$
\begin{aligned}
[i, constante, j, k] &: \text{si } R_0 = i \text{ alors } R_j := k \text{ et } R_0 := R_0 + 1, \\
[i, plus, j, k] &: \text{si } R_0 = i \text{ alors } R_j := R_j + R_k \text{ et } R_0 := R_0 + 1, \\
[i, moins, j, k] &: \text{si } R_0 = i \text{ alors } R_j := \max\{0, R_j - R_k\} \text{ et } R_0 := R_0 + 1, \\
[i, deindirect, j, k] &: \text{si } R_0 = i \text{ alors } R_j := R_{R_k} \text{ et } R_0 := R_0 + 1, \\
[i, aindirect, j, k] &: \text{si } R_0 = i \text{ alors } R_{R_j} := R_k \text{ et } R_0 := R_0 + 1, \\
[i, sizero, j, k] &: \text{si } R_0 = i \text{ alors } R_0 := \begin{cases} R_k, & \text{si } R_j = 0, \\ R_0 + 1, & \text{si } R_j \neq 0. \end{cases}
\end{aligned}
$$

Architecture générale

- Configuration courante

R_0	R_1	R_2	
0	3	2	

$$P = \left\{ \begin{array}{l} [0, plus, 2, 1], \\ [1, constante, 11, 1], \\ [2, deindirect, 5, 2], \\ [3, sizero, 5, 8], \\ [4, moins, 5, 11], \\ [5, aindirect, 2, 5], \\ [6, plus, 2, 11], \\ [7, constante, 0, 1] \end{array} \right\}$$

- Prochaine configuration

R_0	R_1	R_2	
1	3	5	

Architecture universelle

- Configuration courante

R_0	R_1	R_2					
50			P	0	3	2	

$$U = \left\{ \begin{array}{l} [0, constante, 8, 0], \\ [1, constante, 10, 2], \\ [2, constante, 11, 11], \\ \ldots \\ [99, constante, 0, 49], \\ [100, plus, 9, 1], \\ [101, deindirect, 9, 9], \\ [102, plus, 1, 9] \end{array} \right\}$$

- Environ 26 configuration après

R_0	R_1	R_2					
50			P	1	3	5	

Complexité de notre programme universel à adressage indirect

Complexité générale,

$$\lambda(P,x) = \frac{|trace(U,\ code(P)\cdot x)|}{|trace(P,\ x)|} \leq 35 + k\frac{|code(P)|}{|trace(P,\ x)|}$$

Coefficient d'introspection,

$$\lambda = \lim_{n\to\infty} \frac{|trace(U,\ code(U)^{n+1}\cdot x)|}{|trace(U,\ code(U)^{n}\cdot x)|} = 26.27228613$$

Notre programme universel à adressage indirect


```
# INITIALISATION
# DES REGISTRES
[0,constante,8,0],
[1,constante,10,2],
[2,constante,11,11],
[3,constante,12,20],
[4,constante,13,26],
[5,constante,14,33],
[6,constante,15,67],
[7,constante,16,68],
[8,constante,17,70],
[9,constante,18,76],
[10,constante,19,82],
[11,constante,20,88],
[12,constante,21,94],

# CODIFICATION DU
# PROGRAMME A SIMULER
# INITIALISATION DE LA
# POSITION SOURCE R[1] ET
# DU BOOLEEN c
[13,constante,2,24],
[14,constante,5,1],
[15,constante,0,16],
```

```
# INCREMENTATION POSITION
# SOURCE R[1]
[16,plus,1,9],
# ETUDE DE CAS SUIVANT
# LA VALEUR a DE R[R[1]]
[17,deindirect,3,1],
[18,aindirect,1,8],
[19,plus,3,11],
[20,deindirect,0,3],
# CAS OU a=1
[21,sizero,5,24],
[22,constante,5,0],
[23,constante,4,0],
[24,plus,4,4],
[25,plus,4,9],
[26,constante,0,15],
# CAS OU a=2
[27,sizero,5,30],
[28,constante,5,0],
[29,constante,4,0],
[30,plus,4,4],
[31,plus,4,9],
[32,plus,4,9],
[33,constante,0,15],
```

```
# CAS OU a=3
[34,sizero,5,36],
[35,constante,0,40],
[36,plus,2,9],
[37,moins,4,9],
[38,aindirect,2,4],
[39,constante,5,1],
[40,constante,0,15],
# FIN
[41,aindirect,1,8],
[42,constante,4,1],
[43,plus,4,1],
[44,constante,6,25],
[45,aindirect,4,6],

# SIMULATION DU PROGRAMME
# SAUTER L'INCREMENTATION
# DU COMPTEUR ORDINAL
[46,constante,0,49],
# INCREMENTATION
# DU COMPTEUR ORDINAL
[47,deindirect,6,1],
[48,plus,6,9],
[49,aindirect,1,6],
```

Notre programme universel à adressage indirect, suite


```
# CALCUL POSITION
INSTRUCTION NO ZERO
[50,deindirect,7,1],
[51,constante,6,25],
[52,plus,6,7],
[53,plus,6,7],
[54,plus,6,7],
# TEST D'ARRET
[55,constante,7,0],
[56,plus,7,1],
[57,moins,7,6],
[58,sizero,7,100],
# CALCUL DE a:=R[R[6]],
[59,deindirect,3,6],
# CALCUL DE b:=R[R[6]]+R[1]
[60,plus,6,9],
[61,deindirect,4,6],
[62,plus,4,1],
# CALCUL DE c:=R[R[4]+2];
[63,plus,6,9],
[64,deindirect,5,6],
# ETUDE DE CAS SUIVANT
# LA VALEUR DE a
[65,constante,6,15],
[66,plus,6,3],
[67,deindirect,0,6],
# PAS D'INSTRUCTION
[68,constante,0,99],
# INSTRUCTION CONSTANTE
[69,aindirect,4,5],
[70,constante,0,46],
# INSTRUCTION PLUS
[72,plus,5,1],
[73,deindirect,5,5],
[74,plus,6,5],
[75,aindirect,4,6],
[76,constante,0,46],
# INSTRUCTION MOINS
[77,deindirect,6,4],
[78,plus,5,1],
[79,deindirect,5,5],
[80,moins,6,5],
[81,aindirect,4,6],
[82,constante,0,46],
# INSTRUCTION DEINDIRECT
[83,plus,5,1],
[84,deindirect,5,5],
[85,plus,5,1],
[86,deindirect,5,5],
[87,aindirect,4,5],
[88,constante,0,46],
# INSTRUCTION AINDIRECT
[89,deindirect,4,4],
[90,plus,4,1],
[91,plus,5,1],
[92,deindirect,5,5],
[93,aindirect,4,5],
[94,constante,0,46],
# INSTRUCTION SIZERO
[95,deindirect,4,4],
[96,sizero,4,98],
[97,constante,0,46],
[98,aindirect,1,5],
[99,constante,0,49],
# FIN
[100,plus,9,1],
[101,deindirect,9,9],
[102,plus,1,9].
```

Matrice A et vecteur B

$$A = \begin{bmatrix}
4 & 4 & 4 & 4 & 4 & 4 & 4 & 5 & 5 & 5 & 5 & 5 & 5 & 5 & 5 & 5 & 5 & 5 & 5 & 5 & 5 \\
9 & 11 & 10 & 11 & 11 & 8 & 9 & 10 & 12 & 11 & 12 & 12 & 9 & 10 & 10 & 12 & 11 & 12 & 12 & 9 & 10 \\
1 & 1 & 2 & 1 & 1 & 1 & 1 & 1 & 1 & 2 & 1 & 1 & 1 & 1 & 1 & 1 & 2 & 1 & 1 & 1 & 1 \\
6 & 8 & 8 & 8 & 8 & 6 & 7 & 7 & 9 & 9 & 9 & 9 & 7 & 8 & 7 & 9 & 9 & 9 & 9 & 7 & 8 \\
2 & 2 & 2 & 2 & 2 & 1 & 1 & 2 & 2 & 2 & 2 & 2 & 1 & 1 & 2 & 2 & 2 & 2 & 2 & 1 & 1 \\
0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\
1 & 1 & 1 & 1 & 1 & 1 & 2 & 1 & 1 & 1 & 1 & 1 & 1 & 2 & 1 & 1 & 1 & 1 & 1 & 1 & 2 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0
\end{bmatrix}
\qquad
B = \begin{bmatrix}
1967 \\ 4085 \\ 310 \\ 2041 \\ 1329 \\ 708 \\ 311 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0
\end{bmatrix}$$

Conclusion

Avant de prononcer le moindre mot, ce n’est pas 7 fois qu’il faut tourner sa langue dans sa bouche, mais c’est 3673 fois.