Université de Clermont1
IUT d'Informatique
1ière année -
Denis RICHARD

INTRODUCTION aux COURS de

LOGIQUE

ARITHMÉTIQUE et AUTOMATES

ALGÈBRE de BOOLE

LES NOTATIONS d'ALGORITHME ;
de SYSTÈME FORMEL ;
de SYNTAXE ;
de SÉMANTIQUE

vues par des EXEMPLES.

1 À propos d'algorithme

1.1 Une existence prouvée de façon non algorithmique :

Théorème 1.1 $\exists\alpha \in \mathbb{R}\backslash\mathbb{Q} \quad \exists\beta \in \mathbb{R}\backslash\mathbb{Q} \quad \alpha^\beta \in \mathbb{Q}$.
En clair : Il existe un irrationnel α dont une puissance irrationnelle α^β est rationnelle (= fraction).

Preuve 1 $\sqrt{2} \notin \mathbb{Q}$ *: connu, soit* $\alpha = \sqrt{2}$ *et* $\beta = \sqrt{2}$ *alors*

- *ou bien* $\sqrt{2}^{\sqrt{2}} \in \mathbb{Q}$ *et le théorème est prouvé,*
- *ou bien* $\sqrt{2}^{\sqrt{2}} \notin \mathbb{Q}$

alors on pose $\alpha' = \sqrt{2}^{\sqrt{2}} \qquad \beta' = \sqrt{2}$

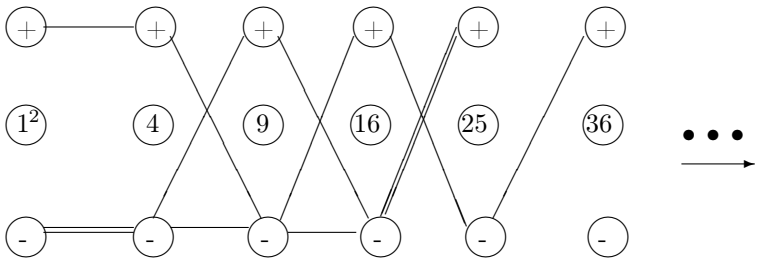
et alors $\alpha'^{\beta'} = (\sqrt{2}^{\sqrt{2}})^{\sqrt{2}} = (\sqrt{2})^2 = 2 = \dfrac{2}{1} \in \mathbb{Q}.$

Remarque :

Preuve non constructive : pas d'algorithme en découlant pour savoir si $\sqrt{2}^{\sqrt{2}} \in \mathbb{Q}$.

1.2 Un problème et sa solution algorithmique :

- Considèrons la ***suite infinie*** des ***carrés***, et des ***chemins finis connexes,, partant*** de 1.



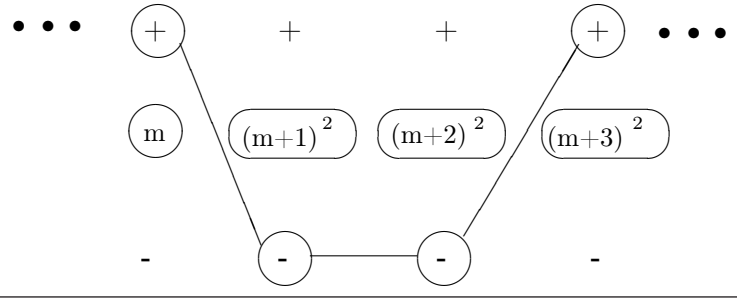
- Chaque chemin possède une ***somme*** :
  $\sum(-) = 1 + 4 - 9 + 16 - 25 + 36 = 23$
  $\sum(-) = -1 + 9 - 16 + 25 = 13$
  $\sum(-) = -1 - 4 - 9 - 16 + 25 = -5$
  $\sum\sum((=)) = -1 - 4 = -5.$

Problème : (Résolu par Erdös et Suraynii.) : ***Tout entier*** $n \in \mathbb{N}$ est-il ***somme*** d'un ***chemin*** ?

Solution algorithmique :

- Ici, pour prouver qu'il a un ***algorithme***, l'orateur trouvera des ***chemins*** correspondants à des nombres fournis par l'auditoire.

- Les exemples conduisent à s'intéresser au ***chemin*** ou plutôt au "***morceau de chemin***" suivant



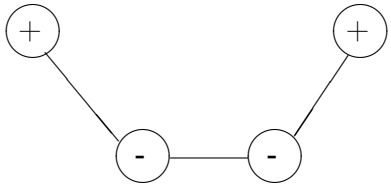
dont la somme est : $m^2 - (m^2 + 2m + 1) - (m^2 + 4m + 4) + (m^2 + 6m + 9) = 4$
m est un ***paramètre***, on le choisit comme on veut.

Or tout entier $n \in \mathbb{N}$ s'écrit

$$\text{ou} \begin{cases} n = 4k = k4 \\ n = 4k + 1 = 1 + k4 \\ n = 4k + 2 = 2 + k4 \\ n = 4k + 3 = 3 + k4 \end{cases}$$

donc tout n est somme d'un chemin fait

- d'un <u>petit chemin</u> de somme $0, 1, 2$, ou 3 se terminant en x^2
- <u>suivi d'un morceau</u> de chemin fermé de morceaux successifs



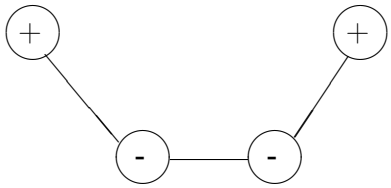
(Il en faut k) et qui commence en $\oplus$ de $(x + 1)^2$.

- Reste à inventer les quatre "***petits chemins***".

$$(*) \text{ ou } \begin{cases} 1 = 1^2 \\ 2 = -1^2 - 2^2 - 3^2 + 4^2 \\ 3 = -1^2 + 2^2 \\ 0 = 0^2 \\ 0 = \underbrace{+1^2 - 2^2 - 3^2 + 4^2}_{4} - \underbrace{5^2 + 6^2 + 7^2 - 8^2}_{4} \end{cases}$$

Remarque : L'***algorithme*** est

- Trouver le quotient k et le reste r de n divisé par 4.
- Le chemin est celui de r (voir $(*)$) suivi de k fois les morceaux du type :



Questions :

1) Trouver d'autres algorithmes.

2) De ***combien*** de ***chemins*** un ***entier peut-il être somme*** ?

1.3 Un algorithme sous-forme d'automate

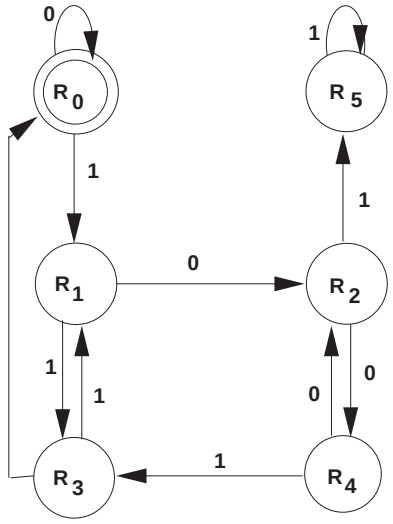
Problème : Calculer le reste r de n divisé par 6 (pour n donné en numération binaire)

$77 = 12 \times 6 + 5$

$n = 77$ et $r = 5$

$77 = 64 + 8 + 4 + 1 = \overbrace{1001101}^{77 \text{ en binaire}}$.

Solution :



Calcul :

lettre lue : 1 0 0 1 1 0 1

état : $R0 \to R1 \to R2 \to R4 \to R3 \to R1 \to R2 \to$ résultat $R5$ $R = 5$.

Remarque :

- Tout algorithme n'est pas automatisable (on verra des exemples).
- Construire les automates, quand c'est possible et simplifier, (ceux-ci fait l'objet du cours 2).

2 À propos des systèmes formels

2.1 Le système *MIU* de POST (1920)

Le système formel MIU utilise 3 lettres M, I et U et les chaînes de caractère (ou MOTS) sont toutes des suites finies de ces 3 lettres.

Exemples : $M\ I$

$M\ U$

$U\ I\ M$

$MMM\ UUU\ III$

$U\ I\ U\ I\ U\ M\ U\ I\ M\ U\ I\ M\ M\ I\ I\ I\ I\ I\ U\ M\ M\ M\ U\ M\ I\ M\ I.$

Problème : On va se donner 4 règles sur les mots, et un mot initial. Il va falloir trouver les mots produits à partir de MI par application exclusive des 4 règles.

• **RÈGLE 1** : $XI \to XIU$
Si un mot se termine par I, on peut lui ajouter U.

• **RÈGLE 2** : $MX \to MXX$
Si un mot commence par M et s'écrit MX, où X est un autre mot, alors on peut lui ajouter X et écrire MXX.

• **RÈGLE 3** : $XIIIY \to XUY$
On peut remplacer dans un mot III par U.
$(*)$ (Attention : X ou Y peut-être le mot vide.)

• **RÈGLE 4** : $XUUY \to XY$
On peut supprimer UU dans un mot.
$(*)$ Même remarque qu'en règle 3.

Exemples

$$\begin{array}{lcll} MI & \to & MIU & \text{(règle 1)} \\ MIUIU & \to & MIUIUIUIU & \text{(règle 2)} \\ MUMU & \to & MUMUUMU & \text{(règle 2)} \\ MIIIM & \to & MUM & \text{(règle 3)} \\ IIIMM & \to & UMM & \text{(règle 3)} \\ MMIIIII & \to & MMIIU & \text{(règle 3)} \\ MUUUU & \to & MUU \to M & \text{(règle 4)} \\ MU^{2n}M & \to & MM & \text{(règle 4).} \end{array}$$

Récapitulons : On se donne

3 lettres : M, I, U
un AXIOME : MI
4 RÈGLES : $XI \to XIU$ - 1
$MX \to MXX$ - 2
$XIIIY \to XUY$ - 3
$XUY \to XY$ - 4.

Et on a ainsi le ***système formel de POST.***

Par définition, un mot produit dans ce système sera dit **THÉORÈME**.
Toute suite de ***mots*** (***discours***) qui conduit depuis MI (l'axiome) jusqu'au mot X sera dite ***preuve formelle*** de X.

Exemple : Théorème : $MUIIU$.

PREUVE FORMELLE :

$$M\underbrace{I,}_{2}\ \underbrace{MI}_{2}\underbrace{I,}_{2}\ \underbrace{MII}_{1}\underbrace{II,}_{3}\ \underbrace{MIIIU}_{2}\underbrace{IU,}_{4}\ MUIIU.$$

[Problème du code de la preuve] (2,2,1,3,2,4) ne suffit pas.
On peut donc ainsi produire des théorèmes. La question est de savoir si un mot donné est un théorème.

Problème 1 : MU est-il un théorème du système de POST ?

Problème 2 : Peut-on produire tous les théorèmes du système ?

Problème 3 : Étant donné un mot, peut-on savoir s'il est un théorème ?

Solution du problème 1 :

- Le nombre de I dans MI est 1,
  dans MU est 0.
- Soit α le nombre de I dans un mot.
- L'application de la
  règle 1 ne change pas α,
  règle 2 multiplie par 2 et donne 2α lettres I,
  règle 3 diminue de 3 le nombre de I et donne $\alpha - 3$.

Si X contient $3k + 1$ lettres I, tous les mots produits dans MIU à partir de X auront $3k' + 1$ lettres I.