Introduction à l'Informatique Théorique

Paolo Tranquilli

14 Septembre 2010

Quelque renseignement

Cours : Paolo Tranquilli, paolo.tranquilli@ens-lyon.fr

TD : Adrien Figgeri, adrien@figgeri.net

Liste : intro.info.ixxi@listes.ens-lyon.fr

Page : perso.ens-lyon.fr/paolo.tranquilli/?q=ens/introinfo1011

Horaires :

cours : mardi, 13h30–15h30

TD : tous les deux mardis à partir du 28/09, 15h30–17h30 (10h30–12h30 le 23/11)

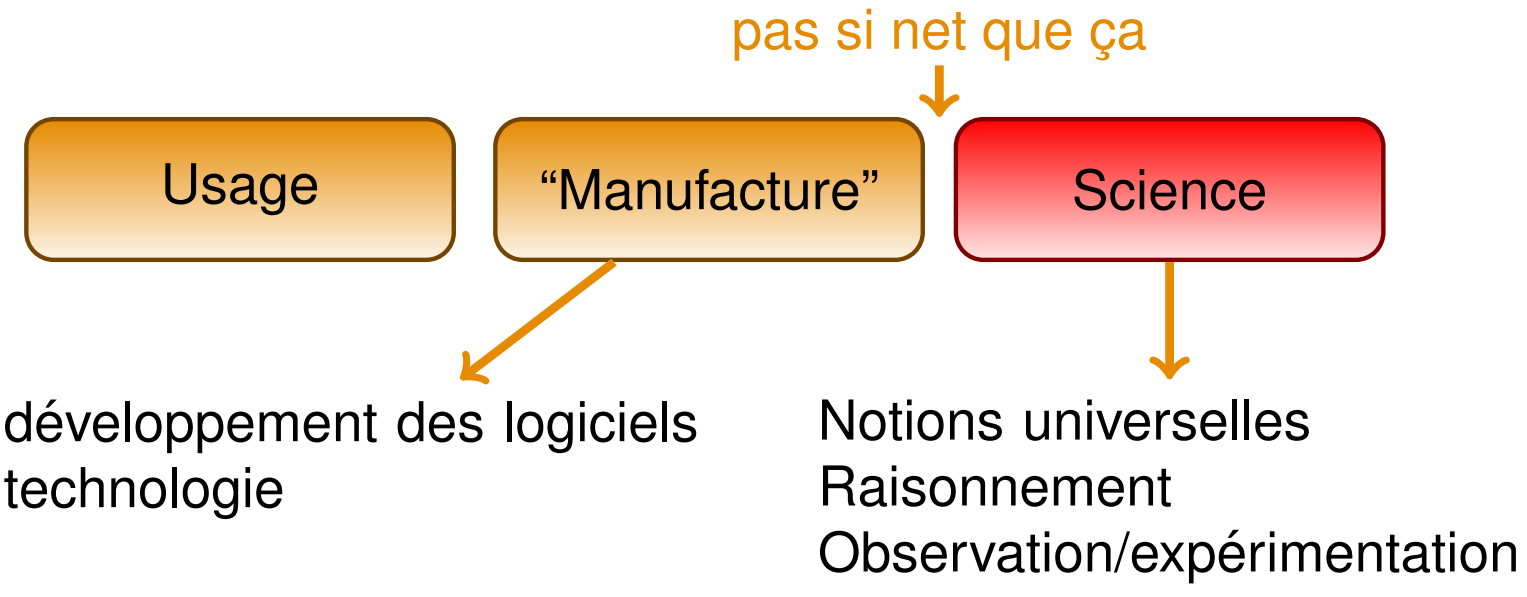
changements éventuels seront communiqués sur la page et par la liste de diffusion

C'est quoi l'informatique ?

informatique /ɛ̃.fɔʁ.ma.tik/ féminin

Science du traitement automatique et rationnel de l'information en tant que support des connaissances et des communications ; ensemble des applications de cette science, mettant en œuvre des matériels (ordinateurs) et des programmes (logiciels).

(Larousse Pratique)



La science informatique

- Ce cours verte sur l'informatique théorique :

Fondements logiques et mathématiques de la science informatique

- À part ça :

architecture (hardware)	réseaux
systèmes d'exploitation	sécurité
langages de prog	compilation
bases de données	intelligence artificielle
interface homme/machine	.....

C’est quoi l’information ?

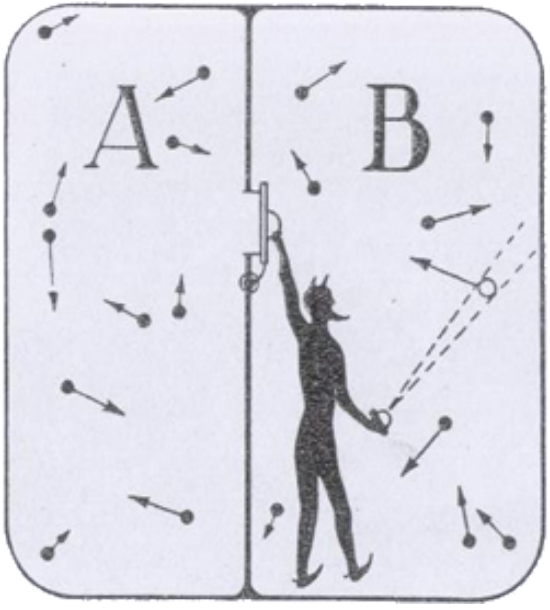
- En termes générales : ce qui permet de distinguer une chose d’une autre, ou un état d’un système d’un autre, sa description totale
- Techniquement, on va régarder un information comme une séquence de symboles.
  Pour exemple :

nombres :	`1768842863`
bits :	`01101001011011100110011001101111`
mots :	`info`
formules :	`x^2+1=0`

$$\text{alphabet fini } \Sigma \rightsquigarrow \Sigma^* = \{ \text{ mots engendrés par } \Sigma \}$$

À propos d'information...

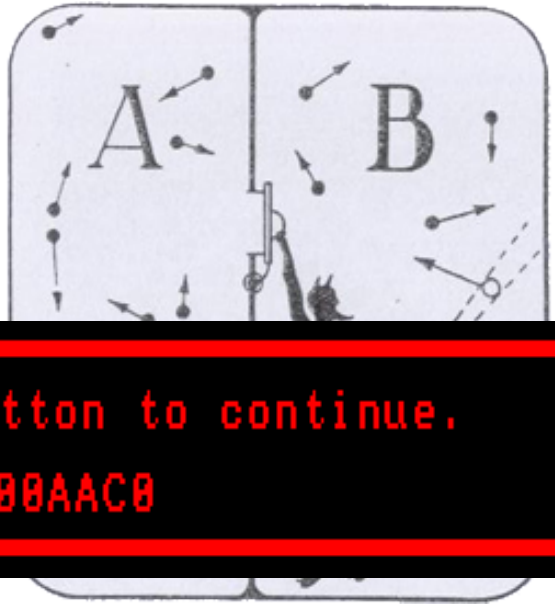
- L'information a aussi une connotation physique
- En 1867 Maxwell décrivit une expérience imaginaire qui contredirait le $2^{ème}$ principe de la thermodynamique, à l'aide d'un démon (décrit comme un être fini)
- 1929, Szilárd (et plus tard Brillouin) : no, mesurer les molécules augmente l'entropie

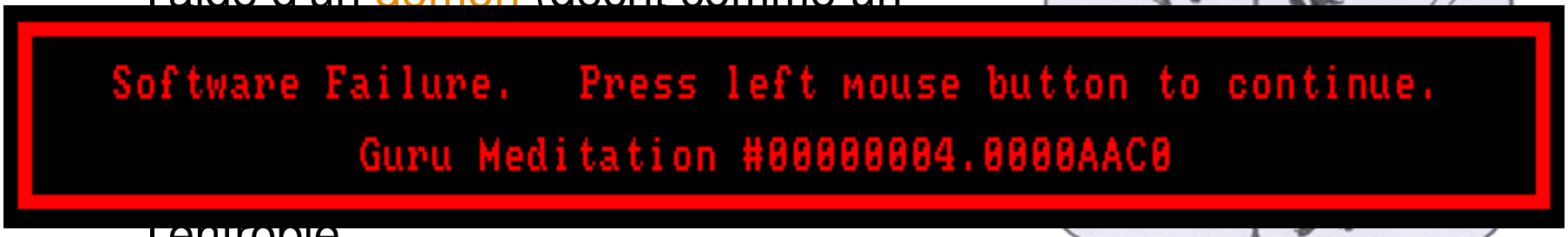


- 1960, Landauer : si, il y a de mesures thermodynamiquement réversibles
- 1982, Bennet : no, le démon devrait garder l'information qu'il utilise (un bit par molécule), car c'est la destruction d'information qui augmente l'entropie.
  Le démon, en tant que fini, va épuiser sa mémoire !

À propos d'information...

- L'information a aussi une connotation physique
- En 1867 Maxwell décrivit une expérience imaginaire qui contredirait le 2ème principe de la thermodynamique, à l'aide d'un démon (décrit comme un





l'entropie
- 1960, Landauer : si, il y a de mesures thermodynamiquement réversibles
- 1982, Bennet : no, le démon devrait garder l'information qu'il utilise (un bit par molécule), car c'est la destruction d'information qui augmente l'entropie.
  Le démon, en tant que fini, va épuiser sa mémoire !

Traitement automatique de l'information

c'est à dire calcul !

Q : que veut dire calculer ?

À part l'acte de calculer une instance spécifique, c'est plus importante la procédure, ou algorithme, c.-à-d. les instructions

Q : c'est quoi une procédure ?

Traitement automatique de l'information

c'est à dire calcul !

Q : que veut dire calculer ?

À part l'acte de calculer une instance spécifique, c'est plus importante la procédure, ou algorithme, c.-à-d. les instructions

Q : c'est quoi une procédure ?

R : Un objet fini qui décrit complétement l'ensemble de calculs qui à partir d'un ensemble (en général infini) des données en entrée mènent aux solutions attendues.

Les détails généraux sont fixés par un certain modèle de calcul.

On a commencé à penser rigoureusement à tout ça il n'y a pas longtemps. . .

Le dixième problème de Hilbert

- 1900, Congres International des Mathématiciens à Paris : 23 problèmes ouvertes qui ont marqué les maths du XX$^{\text{ème}}$ siècle
- 10$^{\text{ème}}$ problème : Trouver une procédure déterminant si une équation diophantienne a des solutions. P.ex.

existent-ils $X, Y \in \mathbb{Z}$ t.q. $X^5 + 5XY^3 - 21Y^2 = 0$? $\overset{\text{procédure}}{\rightsquigarrow}$ oui/non

Le programme de Hilbert

1920, programme de recherche : unifier les mathématiques tout en ayant (démontrables)

- Complétude : tous énoncés valides sont démontrables
- Consistance : on ne peut pas prouver une chose et son contraire
- Décidabilité : il y a une procédure qui étant donnée un énoncé décide si c'est valide ou non

Et encore on ne savait pas qu'est-ce que ça voulait dire précisément !

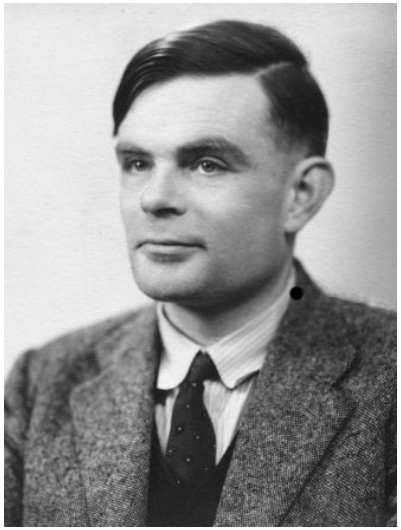
Attention ! Valide vs démontrable

- On fixe des axiomes
- Un énoncé est valide si il est satisfait par tous le modèles (c.-à-d. les “universes” qui satisfont les axiomes)
- Un énoncé est démontrable s’il en existe une preuve, c.-à-d. une séquence qui enchaine des règles de déduction et qui parte des axiomes
- Démontrable $\Rightarrow$ valide (le système de déduction ne dit pas n’importe quoi)
- Théorie complète si valide $\Rightarrow$ démontrable aussi

L'ouvrage de Gödel

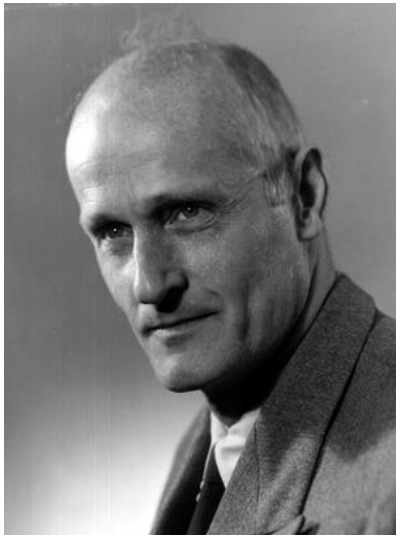
- 1929 : complétude de la logique du premier ordre
- 1931 : incomplétude de l'arithmétique !
- Une notion de fonction calculable apparaît dans la preuve

1936 : quelle année !



Turing

Church



Kleene

Machines de Turing

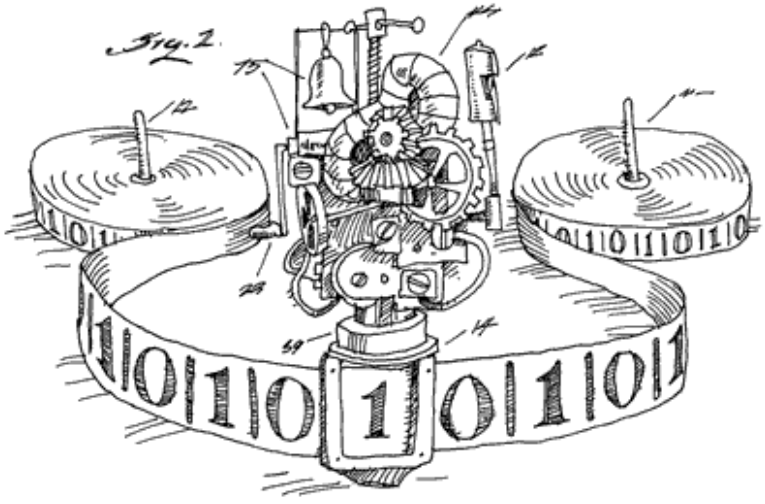
λ-calcul

fonctions μ-récursives

Tout d'un coup, trois définitions différentes de ce qui est calculable

naissance de l'informatique (sans aucun ordinateur)

Machines de Turing



Une machine abstraite qui peut écrire et se déplacer sur une bande potentiellement infini, mais qui se base ses coups sur une quantité fini d'information (contenu de la case et son état entre n possibles et fixés)

Calculable ssi calculé par une machine de Turing (Turing-calculable)

L'arrêt des machines de Turing exprimables en logique du premier ordre mais indécidable

λ-calcul

$$(\lambda x(xx))(\lambda x(xx)) \qquad \lambda x \lambda y\, y$$

Langage où les objets de base sont les fonctions, avec rien d'autre que leur définition et leur application comme opérations

Calculable ssi représentable par un λ-terme (λ-définissable)

Le λ-calcul est le cœur des langages fonctionnels (p.ex. LISP, ML, OCaml, Haskell)

Fonctions μ-récursives

Toutes les fonctions sur les naturels à valeurs naturels qui sont obtenues à partir de certains fonctions élémentaires (constants, successeur, projections) qu'en utilisant :

- composition,
- récursion, p.ex. $\begin{cases} f(0,x) = 0 \\ f(n+1,x) = x + f(n, x+2) \end{cases}$
- l'operateur μf, qui trouve le plus petit argument qui donne 0

Calculable ssi définissable comme fonction μ-récursive

Qui gagne ? Tous ! La thèse de Church

D'abord :

Théorème

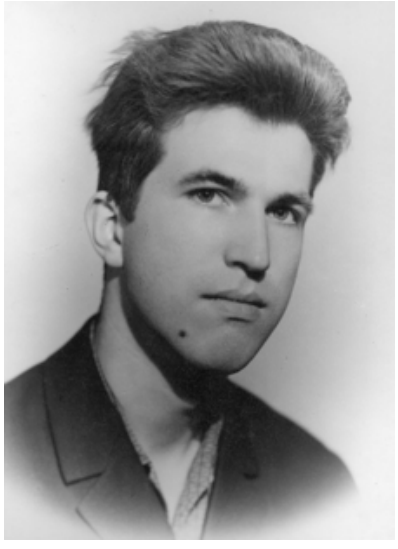
Turing-calculable $\Leftrightarrow$ λ*-définissable* $\Leftrightarrow$ μ*-récursive*

Et puis, plus philosophiquement :

Une formalisation quelconque de la notion intuitive du calcul sera toujours équivalente à celles ci-dessus

Pour exemple, on verra les automates cellulaires

En passant



Q : Et le 10ème problème de Hilbert ?

R : 1970, théorème de Matiyasevich $\Rightarrow$ indécidable !

Bon, on sait calculer, mais combien ça coûte ?

- Au début on s'est concentré sur la question

  "Est telle fonction f calculable si on a assez de ressources ?"

- Pour exemple, on suppose une bande infinie pour les machines de Turing

- Principalement on va discuter de deux ressources :

Temps
nombre d'étapes élémentaires

Espace
dimension de la mémoire utilisée

- De la calculabilité à la complexité

Encore une fois, machines de Turing

- Turing était intéressé qu'à l'aspect calculabilité
- Néanmoins, les machines de Turing se sont montré adaptes aussi comme base théorique de la complexité

Temps
nombre d'étapes afin que la machine s'arrête

Espace
dimension maximale de la portion écrite de la bande

- Ce qui intéresse est la dépendance de ces mesures par rapport à la taille des entrées (= quantité d'information en entrée)
- Attention à distinguer le coût d'une certaine procédure spécifique et le coût d'un problème/fonction calculable (en gros, le coût minimale entre tous les procédures qui le calculent)

P et NP

- Pour exemple

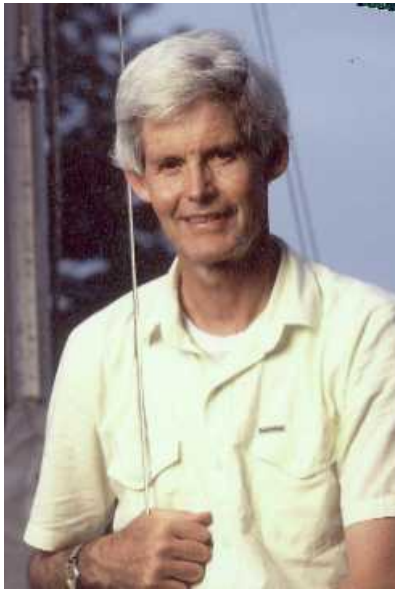
  P : la classe des problèmes dont pour chacun il existe une procédure le résolvant en temps borné par n^d

  où n est la taille de l'entrée et d ne dépende que du problème.
- Peut être considéré comme la classe des problème *traitables*

  NP : la classe des problèmes où vérifier une solution candidate est dans P
- Pour les problèmes NP on peut essayer à deviner une solution jusqu'à trouver la bonne

$$P \subseteq NP$$

P vs NP



Cook

Levin

Cook et Levin indépendemment introduisent en 1970 :

Conjecture

$P \neq NP$?

C'est un des problèmes du prix du millénaire : l'Institut de Mathématiques Clay paie $ 1000000 pour la solution

En résumant

Qu'est-ce qu'on verra en cours ?

- Notions de modèles de calcul, notamment automates finis, puis machines de Turing, fonctions récursives et plus tard automates cellulaires
- Calculabilité et indécidabilité
- Complexité : évaluation de la complexité des algorithmes, puis hiérarchie des classes de complexité, et sa collection de conjectures ouvertes