Les limites de ce que l’on peut calculer

Enseignement et vulgarisation

La forme du savoir de notre savoir

La forme du savoir de nos élèves

L'explosion des mathématiques au XX^{e} siècle

Une grande diversité de théories

Les mathématiques et les autres sciences

La physique

vitesse / dérivée

loi de Newton / eq. différentielle

observable / application linéaire

De nombreuses applications des mathématiques à la physique

De nombreux pb. mathématiques issus de la physique

Et la biologie ?

Les mathématiques et l'informatique

1. Les mathématiques fournissent des outils à l'informatique

2. L'informatique fournit des instruments aux mathématiques

3. L'informatique fournit des instruments et des outils conceptuel (alternatifs aux outils mathématiques) à toutes les sciences

Un résultat négatif

Un problème de la forme : existe-t-il un objet qui vérifie une certaine propriété ?

On répond non

Quelques exemples

Existe-t-il un couple d'entiers tels que $(n/p)^2 = 2$? (pb. et solution -V^e)

Existe-t-il une construction à la règle et au compas d'un disque et d'un carré de même aire ? (pb. -V^e, solution XIX^e)

Existe-t-il une démonstration de l'axiome des parallèles ? (pb. -III^e, solution XIX^e)

Existe-t-il une bijection entre $\mathbb{N}$ et $\mathbb{R}$ (pb. et solution XIX^e)

Une notion un peu floue

Existe-t-il deux nombres a et b tels que

$$(a+b)^2 \neq a^2 + 2ab + b^2$$

?

Non

Existe-t-il un programme (un algorithme) qui ... ?

Non (1936)

Un résultat tardif

On sait répondre ≪ oui ≫ à ce genre de questions depuis 4500 ans (algorithme de l'addition, de la multiplication, ...)

On sait répondre ≪ non ≫ depuis 70 ans

Demande une définition précise de la notion d'algorithme

Les années 30

L’époque des premières définitions de la notion d’algorithme

... des premiers langages de programmation

10 ans avant les premiers ordinateurs

Un langage de programmation

Un programme = un ensemble de règles

$$|X + Y \longrightarrow X + |Y$$

$$+Y \longrightarrow Y$$

Un langage de programmation

Un programme = un ensemble de règles

$$|X + Y \longrightarrow X + |Y$$

$$+Y \longrightarrow Y$$

$$|| + ||$$

Un langage de programmation

Un programme = un ensemble de règles

$$|X + Y \longrightarrow X + |Y$$

$$+Y \longrightarrow Y$$

$$| + ||| $$

Un langage de programmation

Un programme = un ensemble de règles

$$|X + Y \longrightarrow X + |Y$$

$$+Y \longrightarrow Y$$

$$+||||$$

Un langage de programmation

Un programme = un ensemble de règles

$$|X + Y \longrightarrow X + |Y$$

$$+Y \longrightarrow Y$$

Un autre programme

$$f\ |X \longrightarrow g\ X$$

$$g\ |X \longrightarrow f\ |||X$$

Un autre programme

$$f\ |X \longrightarrow g\ X$$

$$g\ |X \longrightarrow f\ |||X$$

$$f\ ||$$

Un autre programme

$$f\ |X \longrightarrow g\ X$$

$$g\ |X \longrightarrow f\ |||X$$

$$g\ |$$

Un autre programme

$$f\ |X \longrightarrow g\ X$$

$$g\ |X \longrightarrow f\ |||X$$

$$f\ |||$$

Un autre programme

$$f\ |X \longrightarrow g\ X$$

$$g\ |X \longrightarrow f\ |||X$$

$$g\ ||$$

Un autre programme

$$f\ |X \longrightarrow g\ X$$

$$g\ |X \longrightarrow f\ |||X$$

$$f\ ||||$$

Un autre programme

$$f\ |X \longrightarrow g\ X$$

$$g\ |X \longrightarrow f\ |||X$$

...

La non-terminaison

$$f\ |X \longrightarrow g\ X$$

$$g\ |X \longrightarrow f\ |||X$$

Le calcul de f en $||$ ne termine pas

Des chiffres et des lettres

$$c \ll aX \gg \longrightarrow |c \ll X \gg$$

$$c \ll bX \gg \longrightarrow c \ll X \gg$$

$$c \ll \ \gg \longrightarrow$$

Des chiffres et des lettres

$$c \ll aX \gg \longrightarrow |c \ll X \gg$$

$$c \ll bX \gg \longrightarrow c \ll X \gg$$

$$c \ll \ \gg \longrightarrow$$

$$c \ll aabbaaba \gg$$

Des chiffres et des lettres

$$c \ll aX \gg \longrightarrow |c \ll X \gg$$

$$c \ll bX \gg \longrightarrow c \ll X \gg$$

$$c \ll \ \gg \longrightarrow$$

$$|c \ll abbaaba \gg$$

Des chiffres et des lettres

$$c \ll aX \gg \longrightarrow |c \ll X \gg$$

$$c \ll bX \gg \longrightarrow c \ll X \gg$$

$$c \ll \ \gg \longrightarrow$$

$$||c \ll bbaaba \gg$$

Des chiffres et des lettres

$$c \ll aX \gg \longrightarrow |c \ll X \gg$$

$$c \ll bX \gg \longrightarrow c \ll X \gg$$

$$c \ll \ \gg \longrightarrow$$

$$||c \ll baaba \gg$$

Des chiffres et des lettres

$$c \ll aX \gg \longrightarrow |c \ll X \gg$$

$$c \ll bX \gg \longrightarrow c \ll X \gg$$

$$c \ll \ \gg \longrightarrow$$

$$||c \ll aaba \gg$$

Des chiffres et des lettres

$$c \ll aX \gg \longrightarrow |c \ll X \gg$$

$$c \ll bX \gg \longrightarrow c \ll X \gg$$

$$c \ll \ \gg \longrightarrow$$

$$|||c \ll aba \gg$$

Des chiffres et des lettres

$$c \ll aX \gg \longrightarrow |c \ll X \gg$$

$$c \ll bX \gg \longrightarrow c \ll X \gg$$

$$c \ll \ \gg \longrightarrow$$

$$||||c \ll ba \gg$$

Des chiffres et des lettres

$$c \ll aX \gg \longrightarrow |c \ll X \gg$$

$$c \ll bX \gg \longrightarrow c \ll X \gg$$

$$c \ll \ \gg \longrightarrow$$

$$||||c \ll a \gg$$

Des chiffres et des lettres

$$c \ll aX \gg \longrightarrow |c \ll X \gg$$

$$c \ll bX \gg \longrightarrow c \ll X \gg$$

$$c \ll \ \gg \longrightarrow$$

$$|||||c \ll \ \gg$$

Des chiffres et des lettres

$$c \ll aX \gg \longrightarrow |c \ll X \gg$$

$$c \ll bX \gg \longrightarrow c \ll X \gg$$

$$c \ll \ \gg \longrightarrow$$

Un autre exemple

Deux mots sont identiques

Un autre exemple

Deux mots sont identiques

$$f \ll aX \gg \ll aY \gg \longrightarrow f \ll X \gg \ll Y \gg$$

$$f \ll bX \gg \ll bY \gg \longrightarrow f \ll X \gg \ll Y \gg$$

$$f \ll \gg \ll \gg \longrightarrow 1$$

$$f \ll aX \gg \ll bY \gg \longrightarrow 0$$

$$f \ll bX \gg \ll aY \gg \longrightarrow 0$$

$$f \ll aX \gg \ll \ \gg \longrightarrow 0$$

$$f \ll bX \gg \ll \ \gg \longrightarrow 0$$

$$f \ll \ \gg \ll aX \gg \longrightarrow 0$$

$$f \ll \ \gg \ll bX \gg \longrightarrow 0$$

Et si on peut traiter des lettres ...

... pourquoi ne pas traiter des programmes

$$c \ll |x + y \longrightarrow x + |y, +y \longrightarrow y \gg \longrightarrow \ldots$$

pourquoi des minuscules ?

Des exemples de programmes qui traite des programmes

Vérifier que les variables à droite et à gauche sont les mêmes

Vérifier que chaque règle à un nombre de symboles à gauche qui est $>$ au nombre de symboles à droite

Que peut-on dire d’un programme tel que dans chaque règle

– les variables à droite et à gauche sont les mêmes
– le nombre de symboles à gauche est $>$ au nombre de symboles à droite ?

De manière plus générale

Peut-on écrire un programme qui vérifie qu'un programme termine ?

(Alan Turing, Alonzo Church-Stephen Kleene 1936)

Depuis 4500 ans

On avait toujours répondu ≪ oui ≫ aux questions de cette forme

Peut-on trouver un algorithme qui ajoute deux nombres ?

Peut-on trouver un algorithme qui multiplie deux deux nombres ?

Peut-on trouver algorithme qui calcule les coefficients binomiaux ?

Peut-on trouver algorithme qui résout des équations linéaires ?

Et pour la première fois

on a répondu « NON »

Un raisonnement par l'absurde

Supposons qu'il existe des règles R telles que

$$h \ll P \gg \ll A \gg$$

se calcule en 1

si A termine quand on le calcule avec les règles P

(et en 0 sinon)

Par exemple

$$h \ll |x + y \longrightarrow x + |y, +y \longrightarrow y \gg \ll || + || \gg$$

se calcule en 1

mais

$$h \ll f\ |x \longrightarrow g\ x, g\ |x \longrightarrow f\ |||x \gg \ll f|| \gg$$

se calcule en 0

On ajoute ...

$$kXY \longrightarrow b\,hXY$$

$$b1 \longrightarrow b1$$

$$b0 \longrightarrow 0$$

Si A termine quand on le calcule avec les règles P alors

$k \ll P \gg\ll A \gg \longrightarrow b\,h \ll P \gg\ll A \gg \longrightarrow b1 \longrightarrow \ldots$

$k \ll P \gg\ll A \gg$ ne termine pas

Et si A ne termine pas quand on le calcule avec les règles P alors $k \ll P \gg\ll A \gg$ termine

« je me brosse les dents »

Ajoutons la règle

$$l \ll X \gg \longrightarrow k \ll X \gg \ll l \ll X \gg \gg$$

et appelons R' l'ensemble R plus les quatre règles ajoutées

$l \ll P \gg$ termine quand on le calcule avec R'

ssi

$k \ll P \gg \ll l \ll P \gg \gg$ termine quand on le calcule avec R'

ssi

$l \ll P \gg$ ne termine pas quand on le calcule avec P

Contradiction

$l \ll P \gg$ termine quand on le calcule avec R'

ssi

$l \ll P \gg$ ne termine pas quand on le calcule avec P

$l \ll R' \gg$ termine quand on le calcule avec R'

ssi

$l \ll R' \gg$ ne termine pas quand on le calcule avec R'

Contradiction ...

Donc l'hypothèse de départ était fausse

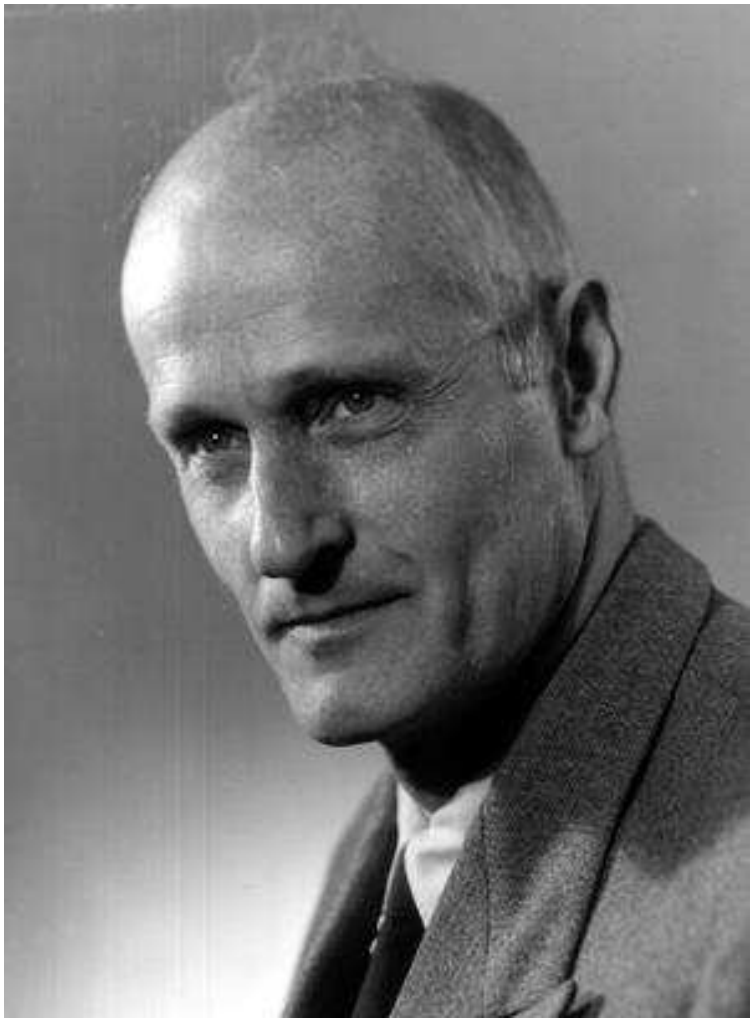
il n'existe pas de règles R telles que

$$h \ll P \gg \ll A \gg$$

se calcule en 1

si A termine quand on le calcule avec les règles P

(et en 0 sinon)



Ce qui était difficile

Trouver une définition précise de la notion d’algorithme

(Une définition vague suffisait pour les résultats positifs)

Le premier théorème d'inexistence d'un algorithme

De nombreux par la suite

En particulier : pas d'algorithme pour décider la démontrabilité

Mais aussi

Collossus, la machine de Manchester, ACE, ...

construits par un certain ... Alan Turing