

# Neumann, Turing, Shannon : des pionniers de l'informatique (partie 2)

SÉBASTIEN VEREL

Laboratoire d'Informatique, Signal et Image de la Côte d'opale (LISIC)  
Université du Littoral Côte d'Opale, Calais, France  
<http://www-lisic.univ-littoral.fr/~verel/>

Juin, 2021



# Plan général

## Programme

- Historique de l'invention de l'informatique au tournant du milieu du XXeme siècle
- Principe des machines à calculer
- Introduction sur la notion de calculabilité
- Introduction de la notion d'information

## Compétences visées

- Connaître les principes fondamentaux de l'informatique
- Connaître les notions d'information
- Connaître les notions de calculabilité

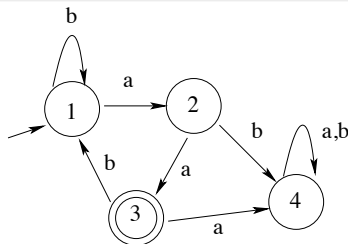
# Machine abstraite : Automate fini

But : calculer si un mot est accepté ou non (mots d'un langage)

## Automate fini

- Etats : mémoire finie,
- Lecture des symboles,
- Programme : fonction de transition d'états

états	a	b
→ 1	2	1
2	3	4
3	4	1
4	4	4



a	a	b	a
---	---	---	---

∧

1

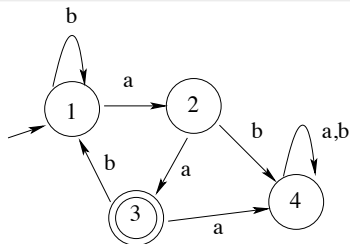
# Machine abstraite : Automate fini

But : calculer si un mot est accepté ou non (mots d'un langage)

## Automate fini

- Etats : mémoire finie,
- Lecture des symboles,
- Programme : fonction de transition d'états

états	a	b
→ 1	2	1
2	3	4
3	4	1
4	4	4



a	a	b	a
---	---	---	---

∧

2

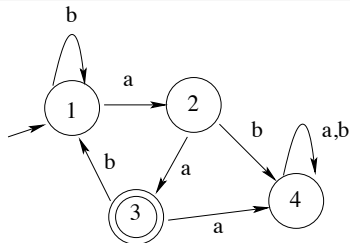
# Machine abstraite : Automate fini

But : calculer si un mot est accepté ou non (mots d'un langage)

## Automate fini

- Etats : mémoire finie,
- Lecture des symboles,
- Programme : fonction de transition d'états

états	a	b
→ 1	2	1
2	3	4
3	4	1
4	4	4



a	a	b	a
---	---	---	---

∧

3

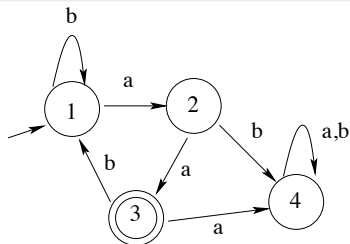
# Machine abstraite : Automate fini

But : calculer si un mot est accepté ou non (mots d'un langage)

## Automate fini

- Etats : mémoire finie,
- Lecture des symboles,
- Programme : fonction de transition d'états

états	a	b
→ 1	2	1
2	3	4
3	4	1
4	4	4



a	a	b	a
---	---	---	---

∧

1

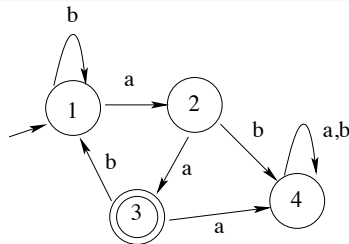
## Machine abstraite : Automate fini

But : calculer si un mot est accepté ou non (mots d'un langage)

### Automate fini

- Etats : mémoire finie,
- Lecture des symboles,
- Programme : fonction de transition d'états

<i>états</i>	a	b
→ 1	2	1
2	3	4
3	4	1
4	4	4

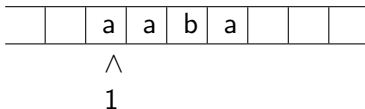


a a b a

∧  
2

# Caractéristiques d'une machine de Turing

Support illimité de l'information : Ruban



## Machine de Turing

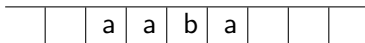
- Etats : mémoire finie,
- Lecture des symboles du ruban **infini**,
- **Ecriture** sur le ruban
- Programme :  
fonction de transition d'états et de déplacement et d'écriture



# Premier exemple

## Fonction de transition

Ancien état	Symbole lu	Symbole écrit	Mouv.	Nouvel état
1	□	□	→	arrêt
	a	b	→	1
	b	a	→	1



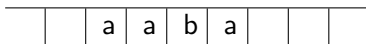
^

1

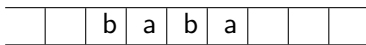
## Premier exemple

## Fonction de transition

Ancien état	Symbole lu	Symbole écrit	Mouv.	Nouvel état
1	□	□	→	arrêt
	a	b	→	1
	b	a	→	1


 $\wedge$ 

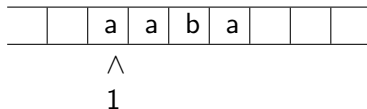
1


 $\wedge$ 

1

# Transition : Tableau à double entrée

	a	b	□
→1	b, →, 1	a, →, 1	□, →, arrêt



## Exo

Exécutez la machine de Turing ci-dessus et décrivez sa fonction de calcul.

# Exemple plus sophistiqué

	a	b	□
→0	a, ←, 0	b, ←, 0	□, →, 1
1	b, →, 1	a, →, 1	□, →, arrêt

		a	a	b	a	b	a	
--	--	---	---	---	---	---	---	--

# Langage décidé par une machine de Turing

	a	b	□
→0	□ , →, 1	refusé	accepté
1	a , →, 1	b , →, 1	□, ←, 2
2	refusé	□ , ←, 3	
3	a , ←, 3	b , ←, 3	□, →, 0

	a	a	a	b	b	b		
--	---	---	---	---	---	---	--	--

	a	a	a	b	a	b		
--	---	---	---	---	---	---	--	--

## Exo

Exécutez la machine de Turing sur les mots ci-dessus et décrivez le langage reconnu.

# Fonction calculée par une Machine de Turing (MT)

- Entrée d'une MT :  
mot inscrit sur le ruban initialement.
- Sortie d'une MT :  
mot inscrit sur le ruban lorsque la MT s'arrête.

## Fonction calculée

La **fonction calculée**  $f$  par une MT  $M$  est définie par :

A toute entrée  $x$  sur laquelle  $M$  s'arrête, on associe la sortie  $y$  :

$$f(x) = y$$

Aucune image n'est associée au mot  $x$  sur lequel  $M$  ne s'arrête pas.

# Fonction calculée par une machine de Turing

## Exemples

- Représentation des nombres en "unaire" (notation en bâton)
- Définir une machine de Turing qui calcule la somme
- Définir une machine de Turing qui la multiplication par 2

Nombre 1 :

	1							
--	---	--	--	--	--	--	--	--

Nombre 3 :

	1	1	1					
--	---	---	---	--	--	--	--	--

$3 + 2$  :

	1	1	1		1	1		
--	---	---	---	--	---	---	--	--

# Machines de Turing équivalentes

On peut imaginer beaucoup de variantes de MT :

- sur un "demi" ruban
- sur deux ou plusieurs rubans
- la tête de lecture peut être stationnaire
- non-déterminisme
- écrire ou non de symbole blanc
- ...

Et pourtant, elles sont toutes équivalentes (reconnaissance du même langage ou fonction calculée identique)

La machine de Turing semble bien représenter une notion de "calcul" par une "procédure effective".



# Machine de Turing universelle

## Machine de Turing universelle

Une machine de Turing universelle est capable de simuler le comportement de n'importe quelle autre machine de Turing.

Le programme est inscrit sur un ruban que la machine universelle est capable d'exécuter.

Le programme devient une donnée !

## Existence

Par exemple, en utilisant 2 rubans :

- sur un ruban le programme de la machine de Turing originale
- sur l'autre ruban le calcul de cette machine

# Fonctions calculables

## Thèse de Church-Turing

Les fonctions calculables par une procédure effective le sont par une machine de Turing.

- Modélisation de la notion de calcul et procédure effective
- Ce n'est pas un résultat que l'on peut démontrer
- Fonctions calculables par MT = fonctions définies par  $\lambda$ -calcul de Church
- Base de la théorie de la calculabilité
- Alonzo Church (1903 -1995), mathématicien, logicien américain.

⇒ **Tous** les ordinateurs sont équivalents à une machine de Turing...

# Fonctions non-calculables

## Exercice

- 1 L'ensemble des machines de Turing est-il dénombrable ?  
*i.e.* autant que de nombre entiers positifs, peut-on numérotter les MT ?
- 2 Existe-il un ensemble de fonctions non-dénombrables ?
- 3 Existe-t-il des fonctions non calculables ?

# Cardinalité de $[0, 1]$

## Théorème

L'ensemble des nombres réels entre 0 et 1 n'est pas dénombrable.

Conséquence : on ne peut pas coder l'ensemble des nombres réels de  $[0, 1]$  avec un ordinateur !

# Cardinalité de $[0, 1]$

## Théorème

L'ensemble des nombres réels entre 0 et 1 n'est pas dénombrable.

Conséquence : on ne peut pas coder l'ensemble des nombres réels de  $[0, 1]$  avec un ordinateur !

**Preuve** : Procédé diagonal de Cantor, voir au tableau.



Georg Cantor, mathématicien allemand, 1845 - 1918.

Théorie des ensembles  
Ensemble bien ordonné  
"infinité d'infinis"

# Fonctions non programmables

## Théorème

L'ensemble des fonctions de  $\mathbb{N}$  dans  $\mathbb{N}$  n'est pas dénombrable.

# Fonctions non programmables

## Théorème

L'ensemble des fonctions de  $\mathbb{N}$  dans  $\mathbb{N}$  n'est pas dénombrable.

Conséquence : il existe des fonctions que l'on ne peut pas programmer !

# Fonctions non programmables

## Théorème

L'ensemble des fonctions de  $\mathbb{N}$  dans  $\mathbb{N}$  n'est pas dénombrable.

Conséquence : il existe des fonctions que l'on ne peut pas programmer !

**Preuve** : Procédé diagonal de Cantor, voir au tableau.



# Fonctions non-calculables

## Exercice

- 1 L'ensemble des machines de Turing est-il dénombrable ?
- 2 Existe-il un ensemble de fonctions non-dénombrables ?
- 3 Existe-t-il des fonctions non calculables ?

Le tout est de savoir lesquelles...

# Propriété décidable et indécidable

## Décidabilité / Indécidabilité

Intuitivement,

- propriété décidable :
  - on peut savoir (démontrer)
  - si pour tout  $x$ ,  $P(x)$  est vrai ou faux.
- propriété indécidable :
  - on ne peut pas savoir (démontrer)
  - si pour tout  $x$ ,  $P(x)$  est vrai ou faux.

# Une phrase indécidable

Les phrases célèbres d'Alain

Alain dit : " Je mens."

# Une phrase indécidable

## Les phrases célèbres d'Alain

Alain dit : " Je mens."

De 2 choses l'une :

- Soit Alain dit vrai,  
et donc il ment et la phrase est donc fausse....
- Soit Alain dit faux,  
et donc il ne ment pas, et la phrase est vraie...

# Une phrase indécidable

## Les phrases célèbres d'Alain

Alain dit : " Je mens."

De 2 choses l'une :

- Soit Alain dit vrai,  
et donc il ment et la phrase est donc fausse....
- Soit Alain dit faux,  
et donc il ne ment pas, et la phrase est vraie...

A faire retourner tous les logiciens grecs dans leur tombe !

Exemple de phrase indécidable :

on ne peut pas démontrer que cette phrase est vraie ou fausse !

# Décidabilité

## Définition

Une famille dénombrable de propriétés  $P(x)$  est **décidable** si sa fonction caractéristiques  $f_P$  est **calculable**.

$$f_P(x) = \begin{cases} 1 & \text{si } P(x) \text{ est vrai,} \\ 0 & \text{si } P(x) \text{ est faux.} \end{cases}$$

## Problème de l'arrêt : fonction non calculable

Fonction qui associe l'arrêt d'une machine de Turing :

$$A(M, e) = \begin{cases} 1 & \text{si la machine de Turing } M \text{ s'arrête sur l'entrée } e \\ 0 & \text{sinon.} \end{cases}$$

### Problème de l'arrêt

La fonction  $A$ , qui associe l'arrêt d'une machine de Turing, est non calculable.

Preuve : argument diagonal

- Supposons qu'il existe une MT  $MA$  qui calcule la fonction  $A$   
 $MA(m, e)$  calcule si la machine  $m$  s'arrête sur l'entrée  $e$  :  
 $MA(m, e) = 0$  lorsque  $m$  s'arrête sur  $e$ ,  $MA(m, e) = 1$  sinon.
- soit la machine :

$$MD(e) = \begin{cases} 1 & \text{si } MA(e, e) = 0 \text{ alors boucle infinie} \\ 0 & \text{si } MA(e, e) = 1 \text{ alors terminer} \end{cases}$$

- Contradiction en analysant  $MA(MD, MD)$

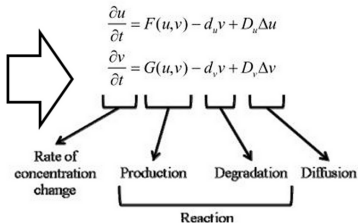
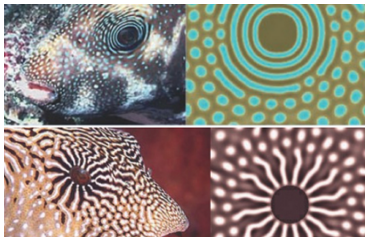
# Fin de la procédure effective de cet exposé

## Travaux de Turing

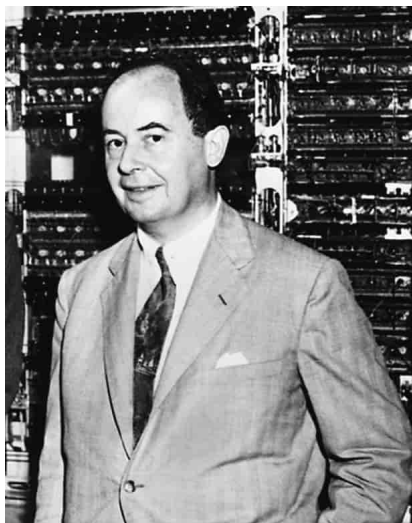
Définition de la notion de calcul, nombres et fonctions calculables, et aussi il montre qu'il y a au moins une fonction non calculable.

Implications nombreuses et fécondes de son approche des nombres calculables

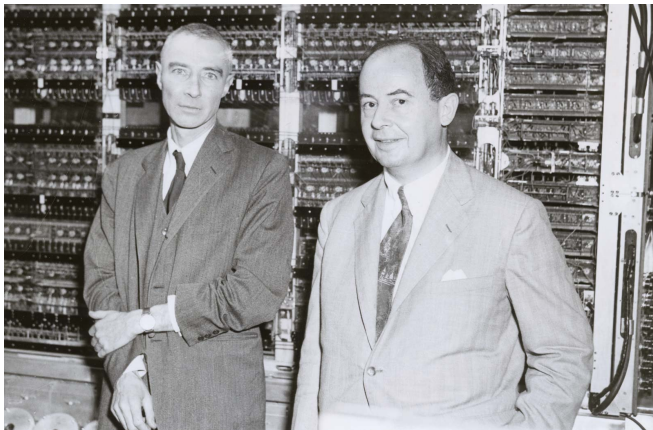
Malheureusement, sans doute parti trop tôt comme le montre ses derniers travaux publiés en 1952 :







Devant l'ordinateur IAS (construction 1945 - 51, Institute for Advanced Study, Princeton, US.)



Devant l'ordinateur IAS (construction 1945 - 51, Institute for Advanced Study, Princeton, US.)



Devant l'ordinateur IAS (construction 1945 - 51, Institute for Advanced Study, Princeton, US.)



- Né en 1903 à Budapest mort en 1957 à Washington
- Mathématicien, physicien
- à 2 ans, sait lire ; à 6 ans, parle grec avec son père et division à 8 chiffres ; à 8 ans, 46 vol. de l'histoire universelle ; Mémoire photographique hors du commun, vitesse de calcul énorme
- 1925, doctorat de mathématiques (axiomatization de la théorie des ensembles de Cantor), univ. Budapest ; en parallèle génie chimique à Zurich

- 1925, doctorat de mathématiques (axiomatization de la théorie des ensembles de Cantor), univ. Budapest ; en parallèle génie chimique à Zurich
- entre 1926 et 1930, bourse Rockefeller pour travailler à l'université de Göttingen, Allemagne, direction D. Hilbert. Rencontre R. Oppenheimer, W. Heisenberg et K. Gödel.
- 1930 : professeur invité à l'université de Princeton, US.
- 1933 - 1957 : Institute for Advanced Study, Princeton. Einstein, Gödel, Dirac, Turing
- Pendant la seconde guerre mondiale, il travaille au Manhattan Project pour produire la première bombe atomique
- 1957 : mort d'un cancer

# Principales publications

- 1923 On the introduction of transfinite numbers
- 1925 An axiomatization of set theory
- 1932 Mathematical Foundations of Quantum Mechanics
- 1937 Continuous geometries with a transition probability
- 1944 Theory of Games and Economic Behavior (with Morgenstern)
- 1945 First Draft of a Report on the EDVAC
- 1946 The Principles of Large-Scale Computing Machines
- 1948 The general and logical theory of automata
- 1960 Continuous geometry
- 1966 Theory of Self-Reproducing Automata

## Logique mathématique

Théorie des ensembles (Cantor) fin du XIXème siècle. Mais paradoxe de Russell.

Zermelo-Fraenkel (ZFC) : axiomatique, mais  $x \in x$  possible...

Refondation axiomatique : NBG (Neumann, Bernays, Gödel)  
un ensemble non vide ne peut pas appartenir à lui-même.

*définition, noyaux astucieux pour lever un paradoxe, et un calcul difficile*

## Physique quantique

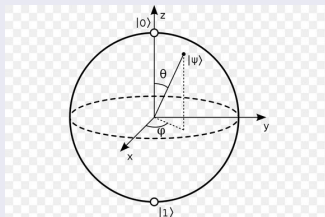
Les 23 problèmes de D. Hilbert en 1900, dont l'un sur l'axiomatisation de la physique

Formalisation matricielle de Heisenberg, et équation diff. ondulatoire de Schrödinger

Réunification : systèmes quantique = vecteurs de l'espace de Hilbert, quantité physique = opérateurs linéaires

(anachronisme mais bon) qubit :

$$|\psi\rangle = \alpha|0\rangle + \beta|1\rangle$$





## Economie

Volonté de définir un langage et une méthode scientifique  
(axiomatique)

Théorie des jeux (28) :

th. minmax : jeu à somme nulle, information parfaite, une stratégie optimale existe

jeux avec asymétrie d'information (44) :

Theory of Games and Economic Behavior avec Morgenstern

## Dilemme des prisonniers

Notre 'jeux' pour 2 joueurs (J1 et J2) :

	Silence	Rock
Silence	3	0
Rock	5	1

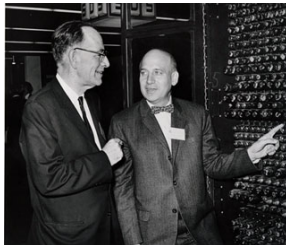
- Si  $J1 = S$  et  $J2 = S$  alors J1 gagne 3 points  
*J1 et J2 gagne par la tranquillité de l'environnement*
- Si  $J1 = S$  et  $J2 = R$  alors J1 gagne 0 point  
*J1 doit supporter son voisin, aucun gain*
- Si  $J1 = R$  et  $J2 = S$  alors J1 gagne 5 points  
*J1 gagne parce qu'il est content de déranger son voisin sans en subir les conséquences*
- Si  $J1 = R$  et  $J2 = R$  alors J1 gagne 1 point  
*J1 gagne peu parce qu'il subit aussi des conséquences de sa nuisance*

# Architecture de von Neumann

<https://interstices.info/le-modele-darchitecture-de-von-neumann/>

Deux tendances : mathématiciens et logiciens / Ingénieurs

- 1943-44 : calculateur Colossus à Bletchley Park, UK, Turing.  
mais une seule tâche
- 1943 - 1946 : Univ. de Pennsylvanie  
par J. Presper Eckert et John Mauchly :  
construction d'un grand calculateur électronique, l'ENIAC
- 1944 : Von Neumann, poste de consultant pour EDVAC



# Architecture de von Neumann

## "First Draft of a Report on the EDVAC", 1945 (vraiment un draft)

### CONTENTS

#### 1.0 DEFINITIONS

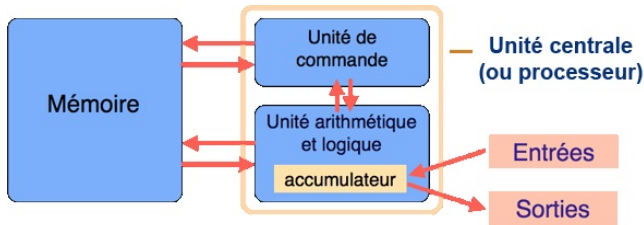
1.1	Automatic digital computing systems .....	1
1.2	Exact description of the functions of such a system .....	1
1.3	Distinctions within the numerical material produced by such a system .....	1
1.4	Checking and correcting malfunctions (errors), automatic possibilities .....	1

#### 2.0 MAIN SUBDIVISIONS OF THE SYSTEM

2.1	Need for subdivisions .....	1
2.2	First: Central arithmetic part: CA .....	1
2.3	Second: Central control part: CC .....	2
2.4	Third: Various forms of memory required: (a)-(h) .....	2
2.5	Third: (Cont.) Memory: M .....	3
2.6	CC, CA (together: C), M are together the associative part. Afferent and efferent parts: Input and output, mediating the contact with the outside. Outside recording medium: R	3
2.7	Fourth: Input: I .....	3
2.8	Fifth: Output: O .....	3
2.9	Comparison of M and R, considering (a)-(h) in 2.4 .....	3

#### 3.0 PROCEDURE OF DISCUSSION

# Architecture de von Neumann



## Innovations

- 1 • Séparation unité de commande et unité arithmétique
- 2 • Programme enregistré dans la mémoire elle-même (et non pas sur support externe)

Mémoire contient données et instruction

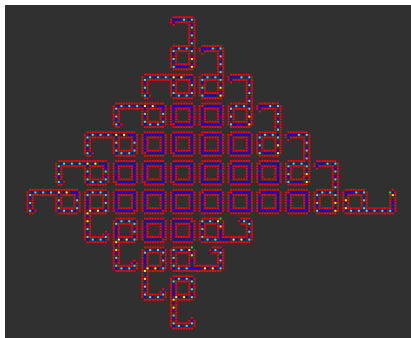
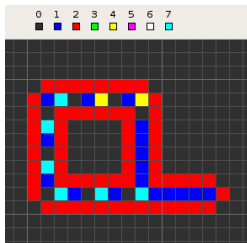
Un programme peut être considéré comme une donnée :  
**autoréférence**

## Modèle auto-reproduction : automate cellulaire

Réflexion sur la structure du cerveau et des systèmes biologiques

Avec Stanislaw Ulam : automate cellulaire (1948)

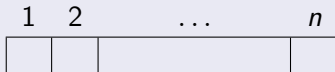
Etude des conditions pour qu'une machine se reproduire elle-même



Boucle de Langton (1984)

# Composants d'un automate cellulaire

- Une grille de  $n$  éléments appelés **cellules**



- Un ensemble d'**états**, aussi appelé alphabet,

$$A = \{0, 1\}$$

- Un voisinage

$$I = \{-1, 0, 1\}$$

- Une table appelée **règle locale**

$$f : \{0, 1\}^3 \longrightarrow \{0, 1\}$$

$\{0, 1\}^3$	000	001	010	011	100	101	110	111
$f$	1	0	1	1	0	1	1	0

# Mise à jour d'une configuration

- L'état de chaque cellule est mise à jour de manière **synchrone**, *i.e.* dans le même instant
- La mise à jour est faite de manière **locale** par la règle locale...
- ... appliquée en fonction de l'état courant de la cellule et des états des cellules voisines

$$\begin{array}{r}
 x \\
 \hline
 \dots \quad x_{i-1} \quad x_i \quad x_{i+1} \quad \dots \\
 \hline
 \underbrace{\hspace{10em}} \\
 f \\
 \hline
 x' \\
 \hline
 \dots \quad \quad \quad x'_i \quad \quad \quad \dots \\
 \hline
 \end{array}$$



## Condition aux limites

$$x = \begin{array}{cccccccccccc} & 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 & 9 & 10 & & \\ \hline & ? & | & 0 & | & 0 & | & 1 & | & 1 & | & 1 & | & 0 & | & 0 & | & 1 & | & 0 & | & 1 & | & ? \\ \hline \end{array}$$

$$x' = \begin{array}{cccccccccccc} & 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 & 9 & 10 & & \\ \hline & ? & | & 0 & | & 0 & | & 1 & | & 1 & | & 1 & | & 0 & | & 0 & | & 1 & | & 0 & | & 1 & | & ? \\ \hline \end{array}$$

### Problèmes des extrémités

Plusieurs solutions :

- Mettre les extrémités dans un état arbitraire (fixe ou non)

$$x = \begin{array}{cccccccccccc} & 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 & 9 & 10 & & \\ \hline & \mathbf{1} & | & 0 & | & 0 & | & 1 & | & 1 & | & 1 & | & 0 & | & 0 & | & 1 & | & 0 & | & 1 & | & \mathbf{0} \\ \hline \end{array}$$

- Faire "boucler" le réseau d'automates : l'extrémité gauche correspond à la cellule  $n$  et l'extrémité droite correspond à la cellule 1

$$x = \begin{array}{cccccccccccc} & 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 & 9 & 10 & & \\ \hline & x_n & | & 0 & | & 0 & | & 1 & | & 1 & | & 1 & | & 0 & | & 0 & | & 1 & | & 0 & | & 1 & | & x_1 \\ \hline \end{array}$$

# Exemple d'évolution

$\{0, 1\}^3$	000	001	010	011	100	101	110	111
$f$	0	0	0	1	1	1	0	1

$$x(0) = \begin{array}{|c|c|c|c|c|c|c|c|c|c|c|} \hline & 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 & 9 & 10 & \\ \hline 1 & 0 & 0 & 1 & 1 & 1 & 0 & 0 & 1 & 0 & 1 & 0 \\ \hline \end{array}$$

$$x(1) = \begin{array}{|c|c|c|c|c|c|c|c|c|c|c|} \hline 1 & 1 & 0 & 1 & 1 & 0 & 1 & 0 & 0 & 1 & 0 & 0 \\ \hline \end{array}$$

$$x(2) = \begin{array}{|c|c|c|c|c|c|c|c|c|c|c|} \hline 1 & 0 & 1 & 1 & 0 & 1 & 0 & 1 & 0 & 0 & 1 & 0 \\ \hline \end{array}$$

$$x(3) = \begin{array}{|c|c|c|c|c|c|c|c|c|c|c|} \hline 1 & 1 & 1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 & 0 & 0 \\ \hline \end{array}$$

$$x(4) = \begin{array}{|c|c|c|c|c|c|c|c|c|c|c|} \hline 1 & 1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 & 0 \\ \hline \end{array}$$

$$x(5) = \begin{array}{|c|c|c|c|c|c|c|c|c|c|c|} \hline 1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 \\ \hline \end{array}$$

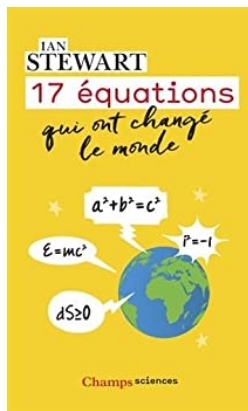
# Observation

- Etat stable ?
- Etat transitoire ?
- Signaux ?
- Frontières ?
  
- Les opinions dans une zone minoritaire disparaissent
- Les frontières sont stables



# Sources

Ian Stewart, 17 équations qui ont changé le monde, Champs Science, Flammarion, 2015.



Machine de Turing

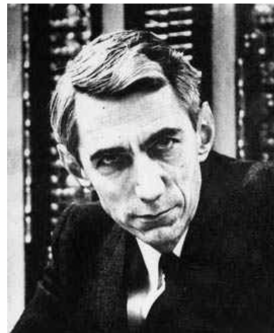
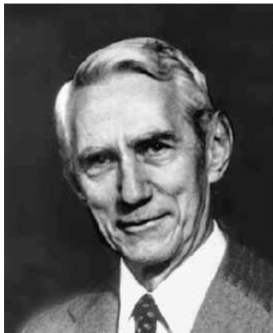
oooooooooooooooooooo

Architecture de Von Neumann

oooooooooooooooooooo

Information de Shannon

●oooooooooooooooooooo



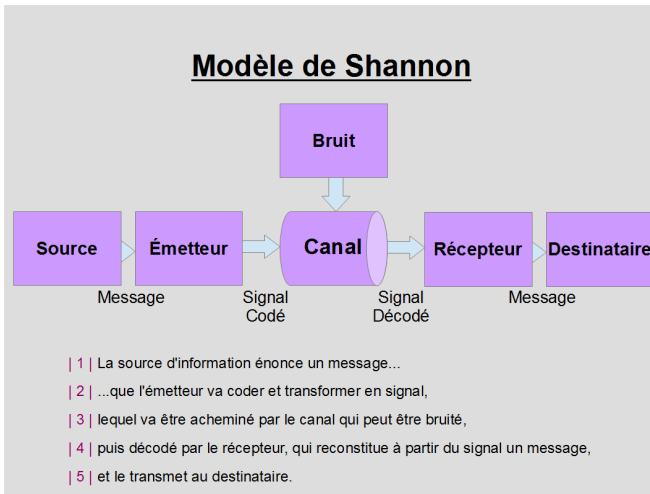
- 1916, Gaylord, Michigan, USA
- Mathématicien, ingénieur électrique
- 1932, University of Michigan : travaux de George Boole
- 1936 : bachelor génie électrique, et mathématiques  
Graduate studies au MIT en génie électrique (analyser différentiel)
- 1937, master' thesis : "A Symbolic Analysis of Relay and Switching Circuits"
- 1940, PhD thesis, MIT, "An Algebra for Theoretical Genetics"
- 1940, National Research Fellow à Institute for Advanced Study in Princeton, New Jersey

- 1940, National Research Fellow à Institute for Advanced Study in Princeton, New Jersey
- Pendant la guerre travaille en cryptographie (rencontre Turing, encodage parole), et les systèmes de contrôle d'incendie
- 1948, publication de "A Mathematical Theory of Communication"
- 1949, avec Warren Weaver : "The Mathematical Theory of Communication".
- de 1956 à 1978, MIT faculty au Research Laboratory of Electronics (RLE)
- de 1941 à 1972, travaille aux laboratoires Bell
- 2001, mort à Medford, Massachusetts, USA

# Modèle de communication

source wikipedia

Cherche un moyen de transmission efficace des messages quand le canal est sujet à des erreurs





# Théorie de l'information

1977, 2 sondes Voyager 1 et 2 sont envoyées, Voyager 1 communique encore !



grace au code de détection et de correcteurs d'erreur

Le terme "information" ne désigne plus seulement les éléments de connaissance, devient une quantité numérique mesurable

2 façons de représenter les nombres :

- par une séquence de symboles (chiffres décimaux par ex.)
- par une correspondance avec une grandeur physique (longueur bâton par ex.)

Vers 1930, calculateurs analogiques (manque de précision)

Vers 1940 émergence des ordinateurs numériques

Deux représentations pour le nombre 157 :

$$1 \times 10^2 + 5 \times 10^1 + 7 \times 10^0$$

$$1 \times 2^7 + 0 \times 2^6 + 0 \times 2^5 + 1 \times 2^4 + 1 \times 2^3 + 1 \times 2^2 + 0 \times 2^1 + 1 \times 2^0$$

Coder 2 niveaux en électronique, facile, fiable.

157 ou 10011101

# Quantité d'information

Choix du binaire pour échanger des messages  
a abouti à l'unité élémentaire d'information :  
binary digit **bit** de Turkey écrit par Shannon en 48

## Information d'une séquence chiffres binaires

Nombre de chiffres de la séquence.

10011101 contient 8 bits d'information

Mais, compter les bits pour mesurer l'information n'a d'intérêt que si 0 et 1 ont la même probabilité d'apparaître

Supposons que 0 apparaît 9 fois sur 10.

Alors on s'attend à recevoir des séquences successive de 0.

Si tel est le cas, peu d'information car on s'y attend

Par contre si un 1 apparaît, alors cela révèle plus information

## Encodage qui profite de "9 fois sur 10"

La partie de gauche s'encode par la partie de droite :

000 → 00

00 → 01

0 → 10

1 → 11

Calculer l'encodage de : 0000000000100010000001000000001

Combien de bit sont nécessaires ?

Codage grossier, il existe surement des choix plus judicieux

## Questions de Shannon

Quel degré d'efficacité peuvent atteindre les codes de ce type ?

Sachant la probabilité de chaque symbole,  
à quel point peut-on raccourcir le message au moyen d'un code approprié ?

# Entropie de Shannon

## Définition du cas binaire

$p$  probabilité que 0 survienne

$q = 1 - p$  probabilité que 1 survienne

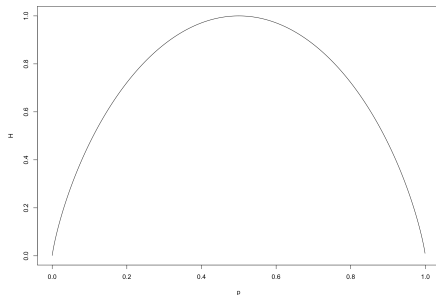
# Entropie de Shannon

## Définition du cas binaire

$p$  probabilité que 0 survienne

$q = 1 - p$  probabilité que 1 survienne

$$H = -p \log_2 p - q \log_2 q$$



Symétrique, minimum 0, maximum 1 pour  $p = 1/2$ .

# Fondement de la définition

## Principes de Shannon d'une bonne définition

Message avec  $n$  symboles avec des proba  $p_1, p_2, \dots, p_n$ .

$H$  la quantité d'information doit vérifier :

- $H$  fonction continue des  $p_i$
- Si toutes proba sont égales (à  $1/n$ ) alors  $H$  doit augmenter avec  $n$
- S'il existe un moyen naturel de décomposer un choix en deux choix successif, alors  $H$  doit être une combinaison simple des nouveaux  $H$

## Démonstration de Shannon

La seule fonction  $h$  possible (à une constante prêt) est :

$$H(p) = -p_1 \log_2 p_1 - p_2 \log_2 p_2 - \dots - p_n \log_2 p_n$$

# Une conséquence

## Quantité d'information que peut transiter un canal

Supposons un signal téléphonique se transmet à travers une ligne de capacité  $C$  bits par seconde.

Supposons que le message du contenu d'information  $H$

Peut-on encoder le signal de façon à ce que la proportion d'erreur soit aussi faible que l'on veut à travers le canal bruyant ?

- Oui si  $H \leq C$
- Non si  $H > C$ . Dans ce cas, proportion d'erreur max est  $H - C$ .

Théorème d'existence, mais il faut trouver le code optimal.  
cf. Théorie des codes pour détecter et corriger les erreur.

Utiliser partout CD, transmission numérique, cryptographie, compression mp3, jpeg, etc.



## Qualité vs. quantité

"deux plus deux font quatre" et "deux plus deux font cinq" ont la même quantité d'information

L'essentiel n'est pas l'information proprement dites, mais le sens.

## Autres domaines

- Biologie : ADN avec le code ATCG  
l'adn contient de l'information, mais l'individu n'est pas réduit à l'adn (épigénétique)
- Entropie de Boltzmann en thermodynamique  
Formules similaires mais contexte différent,  
thermo comme application de théorie de l'information

# Conférence vidéo

"Claude Shannon et l'avènement de l'ère numérique",  
cycle Un texte, un mathématicien, SMF,  
Josselin Garnier, 2016

[https://smf.emath.fr/smf-dossiers-et-ressources/  
garnier-josselin-claude-shannon-et-lavenement-de-lere-numerique](https://smf.emath.fr/smf-dossiers-et-ressources/garnier-josselin-claude-shannon-et-lavenement-de-lere-numerique)

# Et maintenant

## Informatique quantique

1935, A. Einstein intrication quantique :

deux particules peuvent être liées même séparées à grande distance

1936, A. Turing, calculabilité :

Certains problèmes ne peuvent être calculés (problème de l'arrêt)

Nouveaux liens entre intrication quantique et calcul

Quel est la complexité d'un calcul quantique ?

Utilisation des jeux non locaux (!), et preuves interactives

cf.  $MIP^* = RE$ . Zhengfeng Ji, Anand Natarajan, Thomas Vidick, John Wright, Henry Yuen, 2020.

<https://arxiv.org/abs/2001.04383>

# Informatique

Science du traitement automatique de l'information

## Merci à eux et autres pionniers

