

Informatique, informatiques

Pourquoi l'informatique
et pas les informatiques?

Lionel Vaux

Institut de Mathématiques de Marseille (CNRS) & IREM d'Aix-Marseille
Université d'Aix-Marseille

Fête de la science au lycée Val de Durance
10 octobre 2019

Un ordinateur c'est...

Un ordinateur c'est...



domaine public, tiré de Wikimedia Commons

Un ordinateur c'est...



© Kristian Thy, CC-BY-2.0, tiré de Wikimedia Commons

Un ordinateur c'est...

ce qui affiche cette présentation

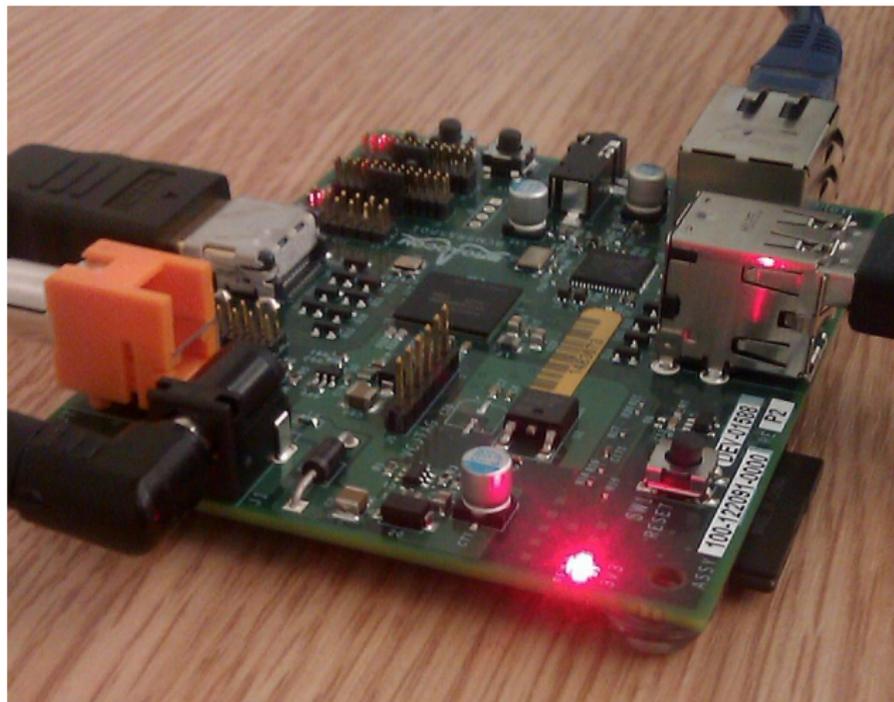
Un ordinateur c'est aussi...

plus petit (mini-PC) :



Un ordinateur c'est aussi...

vraiment tout petit (Raspberry Pi) :



Un ordinateur c'est aussi...
très gros (IBM Blue Gene/P) :



Un ordinateur c'est aussi...

mon téléphone
et le vôtre aussi

Un ordinateur c'est aussi...

mon téléphone
et le vôtre aussi

et je le prouve

Un ordinateur c'est aussi...



© ChtiTux, CC-BY-SA-2.0, tiré de Wikimedia Commons

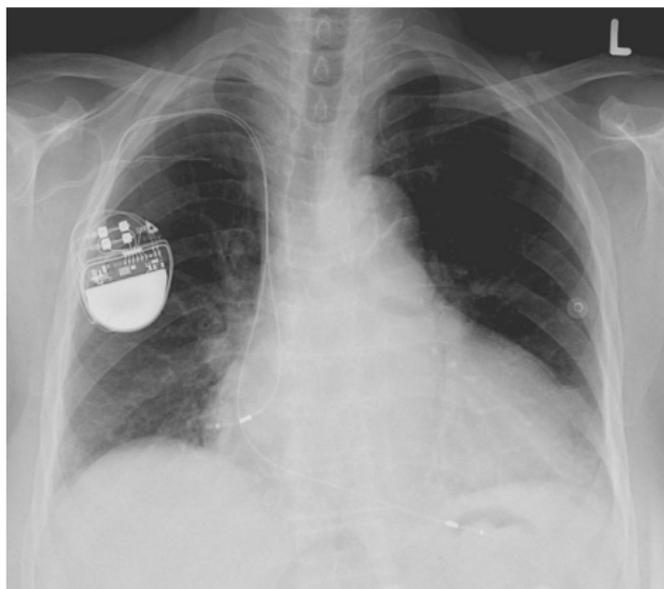
Un ordinateur c'est aussi...



© Jon 'ShakataGaNai' Davis, CC-BY-SA-3.0, tiré de Wikimedia Commons

Mais encore...

Ceci n'est pas un frigo :



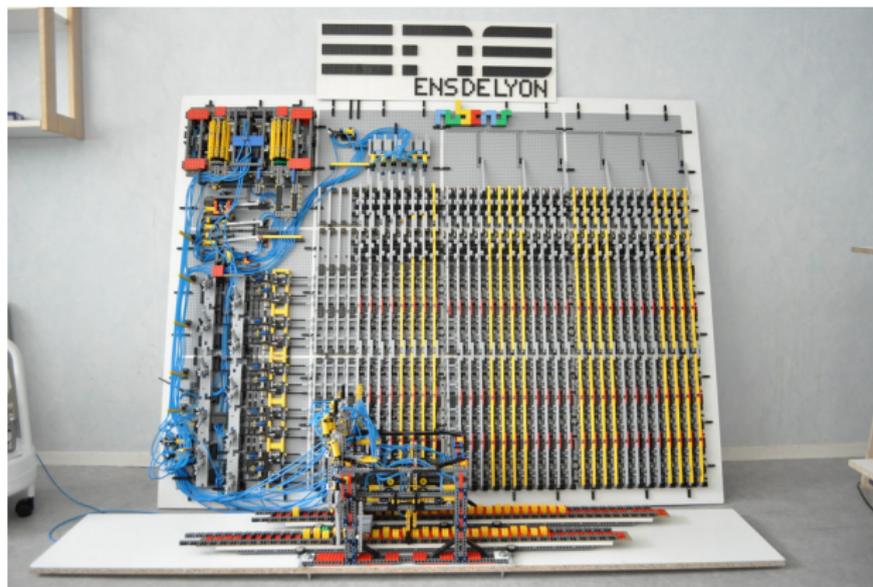
© Lucien Monfils, CC-BY-SA-3.0, tiré de Wikimedia Commons

voir l'histoire de Karen Sandler

<http://www.framablog.org/index.php/post/2012/11/26/un-coeur-gros-comme-ca>
et les travaux de Barnaby Jack sur le piratage de dispositifs médicaux
https://en.wikipedia.org/wiki/Barnaby_Jack#Pacemakers

Ou encore...

Ceci n'est pas un ordinateur nucléaire :



© Projet Rubens, ÉNS de Lyon, CC-BY

voir le site du projet <http://rubens.ens-lyon.fr/>

Parle-t-on vraiment de la même
chose ?

Parle-t-on vraiment de la même
chose ?

un peu d'histoire s'impose

Un ordinateur c'était...

Apple II



Un ordinateur c'était...

ZX Spectrum



domaine public, tiré de Wikimedia Commons

Le premier ordinateur (?)

EDVAC (1949–1961) : électronique, binaire et programmable.

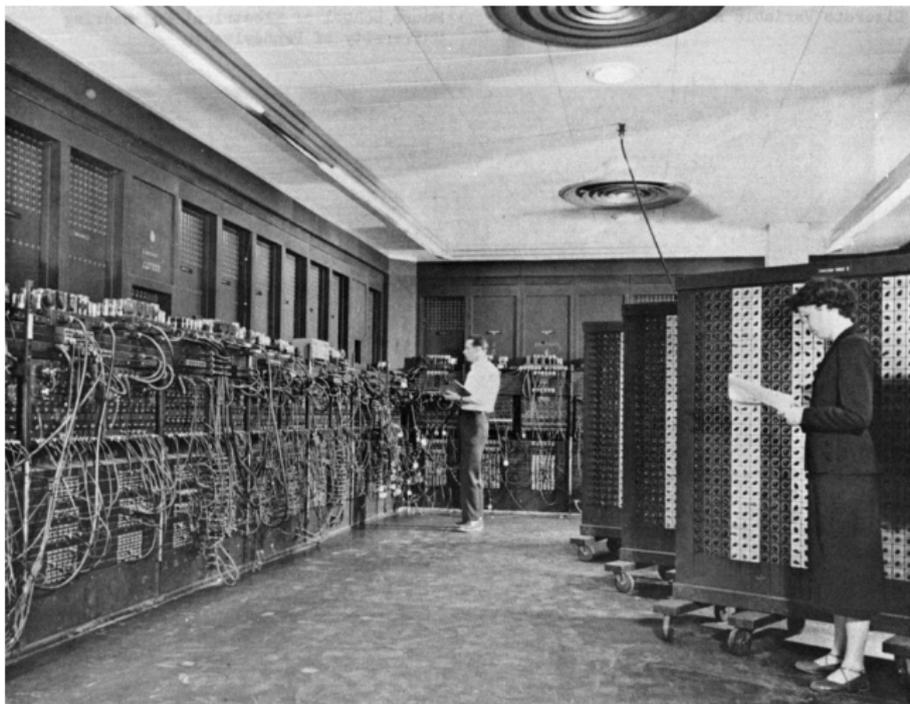


Sur la photo : John von Neumann et Robert Oppenheimer

domaine public, tiré de Wikimedia Commons

Le premier ordinateur (?)

ENIAC (1946–1955) : électronique, décimal et « reconfigurable ».



domaine public, tiré de Wikimedia Commons

Tentative de définition

Un ordinateur, c'est une machine
qu'on peut programmer.

Questions

Qu'est-ce que « programmer » ?

Qu'est-ce que « programmer » ?

un exemple : parité en Scratch et en Python

Questions

Et qu'est-ce qu'une « machine » ?

Questions

Et qu'est-ce qu'une « machine » ?

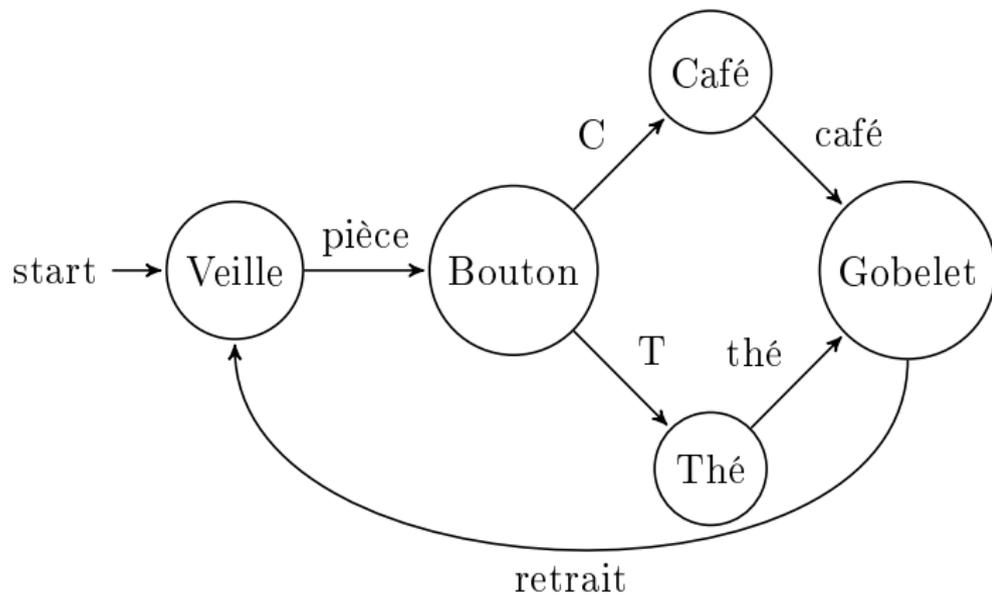
Une machine fonctionne de manière automatique.

Automates

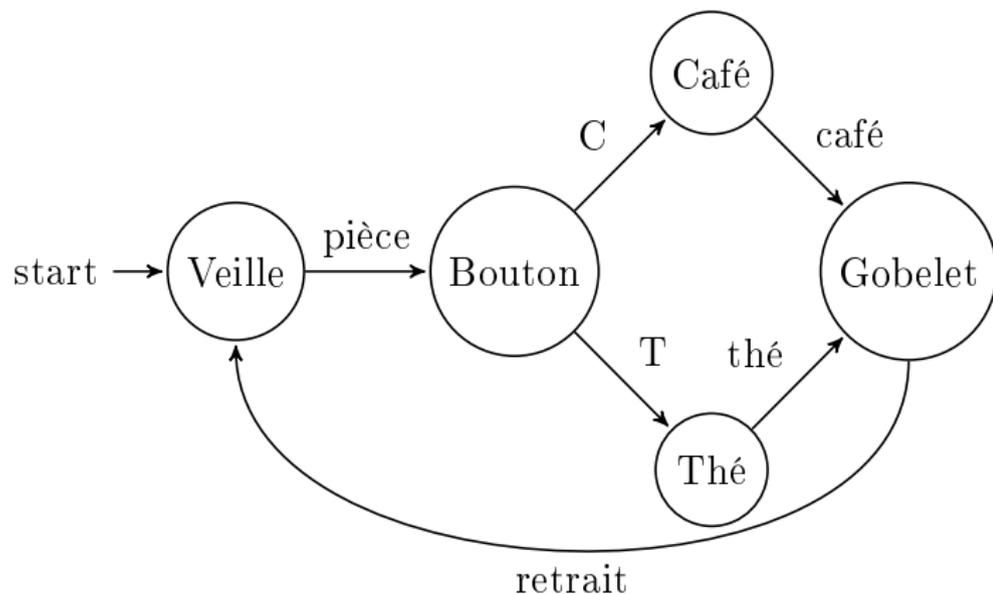


domaine public, tiré de Wikimedia Commons

Automates



Automates



- ▶ un ensemble fini d'états
- ▶ des transitions guidées par la donnée à traiter

Un ordinateur n'est pas une machine à café!

- ▶ entrées
- ▶ sorties
- ▶ mémoire

Un ordinateur n'est pas une machine à café!

- ▶ entrées
- ▶ sorties
- ▶ mémoire

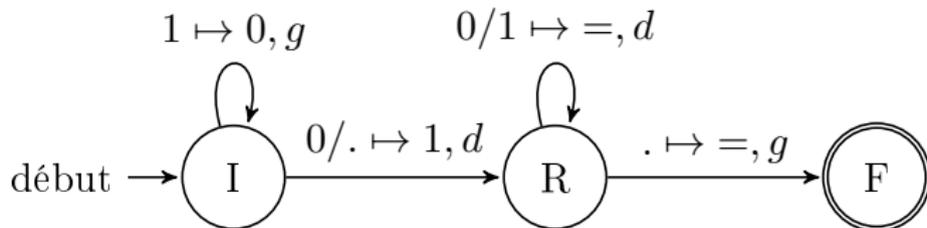
Machines de Turing



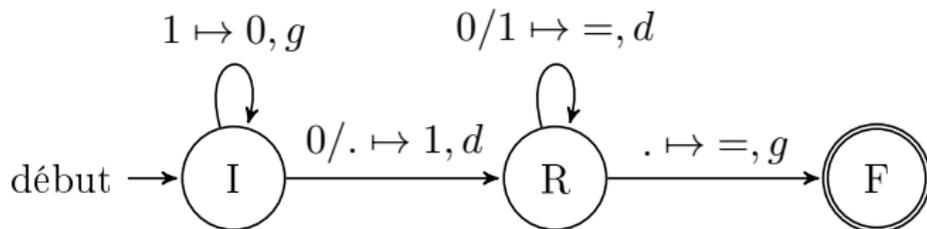
Alan Turing (1912–1954)

domaine public, tiré de Wikimedia Commons

Machines de Turing



Machines de Turing

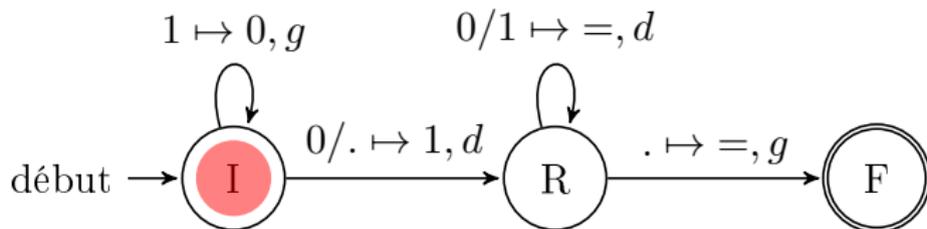


entrée :

...	.	1	0	1	<u>1</u>
-----	---	---	---	---	----------	---	-----

Mémoire : on agit sur un ruban sur lequel on lit et on écrit des caractères.

Machines de Turing

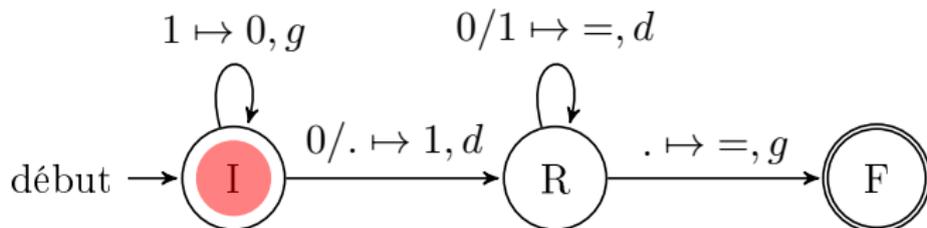


entrée :

...	.	1	0	1	<u>1</u>
-----	---	---	---	---	----------	---	-----

Mémoire : on agit sur un ruban sur lequel on lit et on écrit des caractères.

Machines de Turing



entrée :

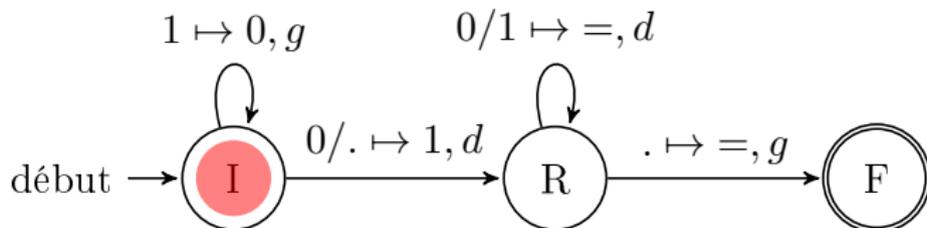
...	.	1	0	1	<u>1</u>
-----	---	---	---	---	----------	---	-----

étape 1 :

...	.	1	0	<u>1</u>	0
-----	---	---	---	----------	---	---	-----

Mémoire : on agit sur un ruban sur lequel on lit et on écrit des caractères.

Machines de Turing



entrée :

...	.	1	0	1	<u>1</u>
-----	---	---	---	---	----------	---	-----

étape 1 :

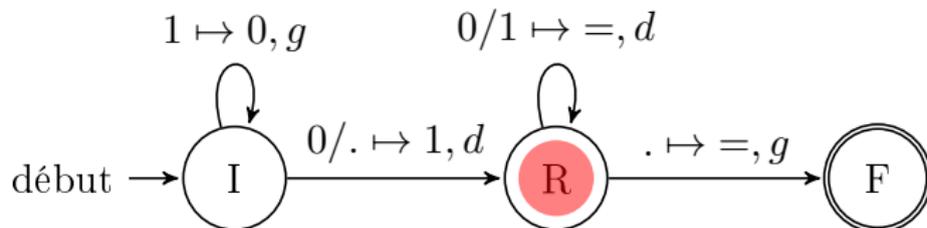
...	.	1	0	<u>1</u>	0
-----	---	---	---	----------	---	---	-----

étape 2 :

...	.	1	<u>0</u>	0	0
-----	---	---	----------	---	---	---	-----

Mémoire : on agit sur un ruban sur lequel on lit et on écrit des caractères.

Machines de Turing



entrée :

...	.	1	0	1	<u>1</u>
-----	---	---	---	---	----------	---	-----

étape 1 :

...	.	1	0	<u>1</u>	0
-----	---	---	---	----------	---	---	-----

étape 2 :

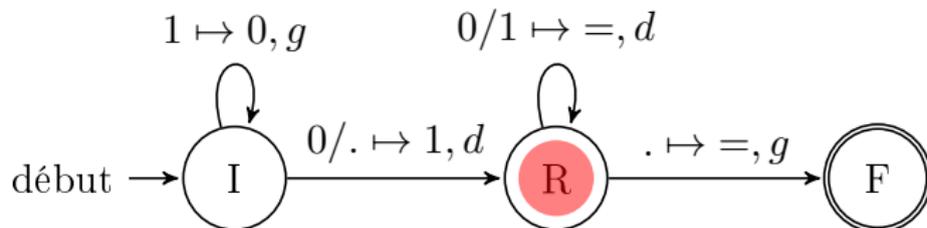
...	.	1	<u>0</u>	0	0
-----	---	---	----------	---	---	---	-----

étape 3 :

...	.	1	1	<u>0</u>	0
-----	---	---	---	----------	---	---	-----

Mémoire : on agit sur un ruban sur lequel on lit et on écrit des caractères.

Machines de Turing



entrée :

...	.	1	0	1	<u>1</u>
-----	---	---	---	---	----------	---	-----

étape 1 :

...	.	1	0	<u>1</u>	0
-----	---	---	---	----------	---	---	-----

étape 2 :

...	.	1	<u>0</u>	0	0
-----	---	---	----------	---	---	---	-----

étape 3 :

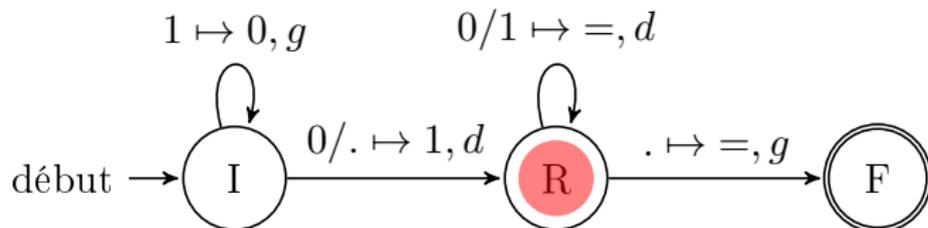
...	.	1	1	<u>0</u>	0
-----	---	---	---	----------	---	---	-----

étape 4 :

...	.	1	1	0	<u>0</u>
-----	---	---	---	---	----------	---	-----

Mémoire : on agit sur un ruban sur lequel on lit et on écrit des caractères.

Machines de Turing



entrée :

...	.	1	0	1	<u>1</u>
-----	---	---	---	---	----------	---	-----

étape 1 :

...	.	1	0	<u>1</u>	0
-----	---	---	---	----------	---	---	-----

étape 2 :

...	.	1	<u>0</u>	0	0
-----	---	---	----------	---	---	---	-----

étape 3 :

...	.	1	1	<u>0</u>	0
-----	---	---	---	----------	---	---	-----

étape 4 :

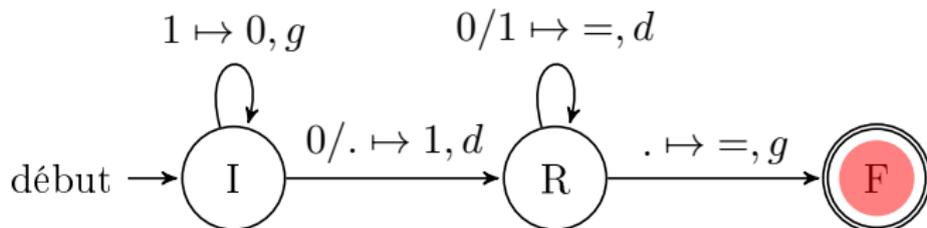
...	.	1	1	0	<u>0</u>
-----	---	---	---	---	----------	---	-----

étape 5 :

...	.	1	1	0	0	<u>.</u>	...
-----	---	---	---	---	---	----------	-----

Mémoire : on agit sur un ruban sur lequel on lit et on écrit des caractères.

Machines de Turing



entrée :

...	.	1	0	1	<u>1</u>
-----	---	---	---	---	----------	---	-----

étape 1 :

...	.	1	0	<u>1</u>	0
-----	---	---	---	----------	---	---	-----

étape 2 :

...	.	1	<u>0</u>	0	0
-----	---	---	----------	---	---	---	-----

étape 3 :

...	.	1	1	<u>0</u>	0
-----	---	---	---	----------	---	---	-----

étape 4 :

...	.	1	1	0	<u>0</u>
-----	---	---	---	---	----------	---	-----

étape 5 :

...	.	1	1	0	0	<u>.</u>	...
-----	---	---	---	---	---	----------	-----

étape 6 :

...	.	1	1	0	<u>0</u>
-----	---	---	---	---	----------	---	-----

Mémoire : on agit sur un ruban sur lequel on lit et on écrit des caractères.

Pourquoi seulement des 0 et des 1 ?

On pourrait travailler avec un ensemble de symboles quelconques :

- ▶ les chiffres décimaux : 0, 1, ... 9 ;
- ▶ tous les caractères disponibles sur un clavier ;
- ▶ un seul symbole (trou sur une carte) ;
- ▶ ...

Pourquoi seulement des 0 et des 1 ?

On pourrait travailler avec un ensemble de symboles quelconques :

- ▶ les chiffres décimaux : 0, 1, ... 9 ;
- ▶ tous les caractères disponibles sur un clavier ;
- ▶ un seul symbole (trou sur une carte) ;
- ▶ ...

Est-ce que ça change quelque chose ?

Pourquoi seulement des 0 et des 1 ?

On pourrait travailler avec un ensemble de symboles quelconques :

- ▶ les chiffres décimaux : 0, 1, ... 9 ;
- ▶ tous les caractères disponibles sur un clavier ;
- ▶ un seul symbole (trou sur une carte) ;
- ▶ ...

Est-ce que ça change quelque chose ?

Pas vraiment : on peut tout coder en binaire.

Exemple

0 \mapsto 0000, 1 \mapsto 0001, 2 \mapsto 0010, 3 \mapsto 0011, 4 \mapsto 0100, 5 \mapsto 0101,
6 \mapsto 0110, 7 \mapsto 0111, 8 \mapsto 1000, 9 \mapsto 1001.

Pourquoi seulement des 0 et des 1 ?

On pourrait travailler avec un ensemble de symboles quelconques :

- ▶ les chiffres décimaux : 0, 1, ... 9 ;
- ▶ tous les caractères disponibles sur un clavier ;
- ▶ un seul symbole (trou sur une carte) ;
- ▶ ...

Est-ce que ça change quelque chose ?

Pas vraiment : on peut tout coder en binaire.

Exemple

0 \mapsto 0000, 1 \mapsto 0001, 2 \mapsto 0010, 3 \mapsto 0011, 4 \mapsto 0100, 5 \mapsto 0101,
6 \mapsto 0110, 7 \mapsto 0111, 8 \mapsto 1000, 9 \mapsto 1001.

mais ça va un peu moins vite...

Pourquoi seulement un ruban ?

On pourrait travailler sur plusieurs rubans simultanément.

Pourquoi seulement un ruban ?

On pourrait travailler sur plusieurs rubans simultanément.

Est-ce que ça change quelque chose ?

Pourquoi seulement un ruban ?

On pourrait travailler sur plusieurs rubans simultanément.

Est-ce que ça change quelque chose ?

Pas vraiment : ça revient à changer les symboles.

...	.	1	<u>0</u>	0
...	.	.	0	1
...	.	.	0	<u>0</u>	0
...	.	1	<u>0</u>	.	0

↦

...	.	1	<u>0</u>	0
...	.	.	0	1
...	.	.	0	<u>0</u>	0
...	.	1	<u>0</u>	.	0

Pourquoi seulement un ruban ?

On pourrait travailler sur plusieurs rubans simultanément.

Est-ce que ça change quelque chose ?

Pas vraiment : ça revient à changer les symboles.

...	.	1	<u>0</u>	0
...	.	.	0	1
...	.	.	0	<u>0</u>	0
...	.	1	<u>0</u>	.	0

↔

...	.	1	<u>0</u>	0
...	.	.	0	1
...	.	.	0	<u>0</u>	0
...	.	1	<u>0</u>	.	0

mais ça va beaucoup moins vite...

Plus généralement

On peut trouver énormément de variantes des machines de Turing...

Plus généralement

On peut trouver énormément de variantes des machines de Turing... mais on peut chaque fois se ramener au cas binaire sur un ruban.

Que peut-on calculer avec une machine de Turing ?

Définition

On dit que la fonction f est calculée par une machine de Turing si :

- ▶ on écrit les nombres n_1, \dots, n_k en binaire sur k rubans ;
- ▶ on fait tourner la machine ;
- ▶ $f(n_1, \dots, n_k)$ est écrit en binaire sur le dernier ruban.

Théorème

Les opérations usuelles sont calculables : somme, multiplication, puissance, etc.

Que peut-on calculer avec une machine de Turing ?

Définition

On dit que la fonction f est calculée par une machine de Turing si :

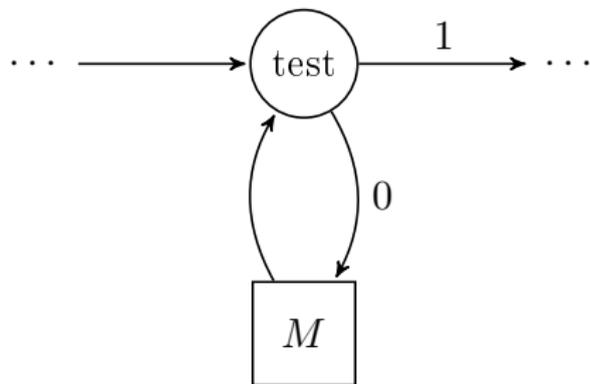
- ▶ on écrit les nombres n_1, \dots, n_k en binaire sur k rubans ;
- ▶ on fait tourner la machine ;
- ▶ $f(n_1, \dots, n_k)$ est écrit en binaire sur le dernier ruban.

Théorème

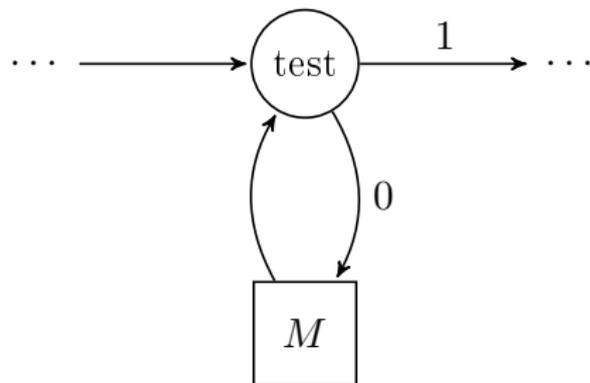
Les opérations usuelles sont calculables : somme, multiplication, puissance, etc.

Mais un ordinateur n'est pas qu'une calculette !

Boucles



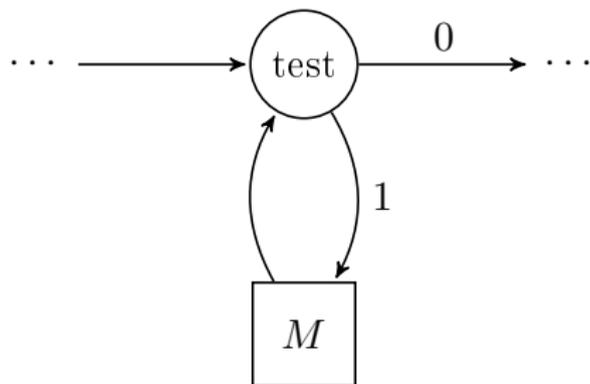
Boucles



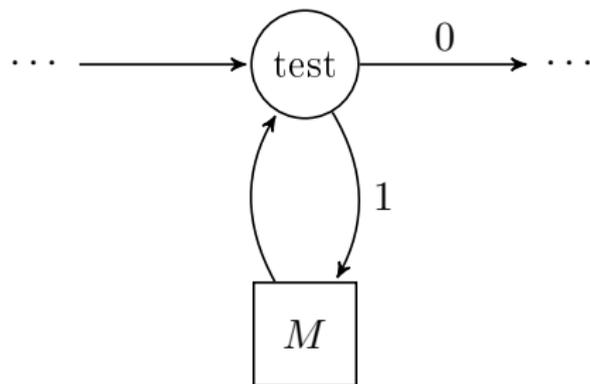
C'est la « boucle » :



Boucles



Boucles



C'est la « boucle » :

```
while test:  
    M
```

Codage d'une machine

On peut décrire une machine de Turing sur un ruban en utilisant les symboles :

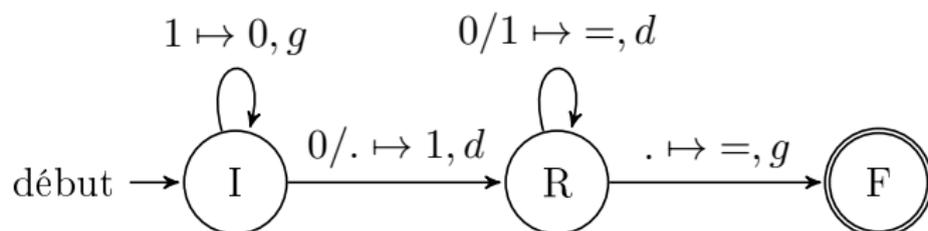
, 0 1 . g d

Codage d'une machine

On peut décrire une machine de Turing sur un ruban en utilisant les symboles :

, 0 1 . g d

Exemple

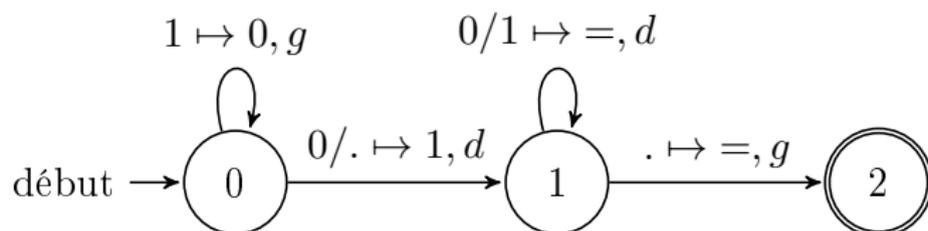


Codage d'une machine

On peut décrire une machine de Turing sur un ruban en utilisant les symboles :

, 0 1 . g d

Exemple

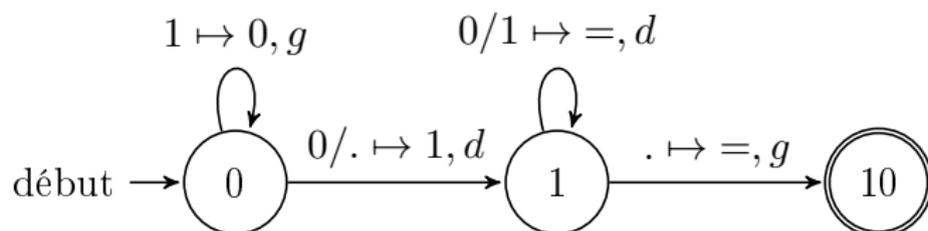


Codage d'une machine

On peut décrire une machine de Turing sur un ruban en utilisant les symboles :

, 0 1 . g d

Exemple

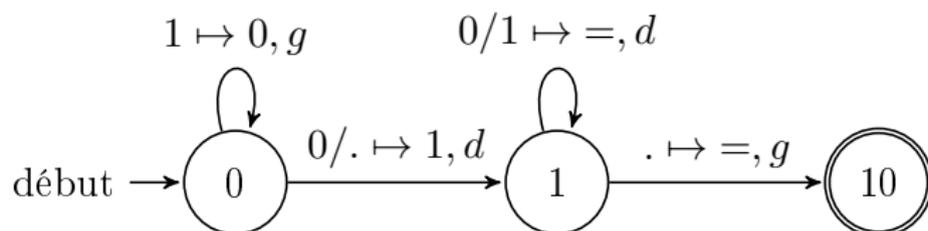


Codage d'une machine

On peut décrire une machine de Turing sur un ruban en utilisant les symboles :

, 0 1 . g d

Exemple



0,.,1,1,d,0,0,1,1,d,0,1,0,0,g,1,.,10,.,g,1,0,1,0,d,1,1,1,1,d,

c'est la table de transitions :

état courant,symbole lu,état suivant,symbole écrit,déplacement

Machine universelle

Théorème

Il existe une machine de Turing U qui calcule la fonction

(code de la machine M , code du ruban d'entrée) \mapsto
résultat de l'exécution de la machine M

Cette machine U **simule** M en suivant sa description.

Machine universelle

Théorème

Il existe une machine de Turing U qui calcule la fonction

(code de la machine M , code du ruban d'entrée) \mapsto
résultat de l'exécution de la machine M

Cette machine U **simule** M en suivant sa description.

En d'autres mots : U **exécute le programme** M .

Machine universelle

Théorème

Il existe une machine de Turing U qui calcule la fonction

*(code de la machine M , code du ruban d'entrée) \mapsto
résultat de l'exécution de la machine M*

Cette machine U **simule** M en suivant sa description.

En d'autres mots : U **exécute le programme** M .

une machine M = un programme pour U

Qu'est-ce qu'un ordinateur ?

Un ordinateur, c'est une machine
qu'on peut programmer.

Qu'est-ce qu'un ordinateur ?

Un ordinateur, c'est une machine
qu'on peut programmer.

Un ordinateur, c'est une machine de
Turing universelle.

Qu'est-ce qu'un ordinateur ?

Un ordinateur, c'est une machine
qu'on peut programmer.

Un ordinateur, c'est une machine de
Turing universelle.

ou quelque chose qui y ressemble

D'autres modèles de calcul

- ▶ les fonctions récursives (Kurt Gödel)
- ▶ le λ -calcul (Alonzo Church)
- ▶ machines RAM (modèle de nos ordinateurs, Von Neumann)

D'autres modèles de calcul

- ▶ les fonctions récursives (Kurt Gödel)
- ▶ le λ -calcul (Alonzo Church)
- ▶ machines RAM (modèle de nos ordinateurs, Von Neumann)
- ▶ les automates cellulaires (ex. : jeu de la vie de Conway)

démo : golly

D'autres modèles de calcul

- ▶ les fonctions récursives (Kurt Gödel)
- ▶ le λ -calcul (Alonzo Church)
- ▶ machines RAM (modèle de nos ordinateurs, Von Neumann)
- ▶ les automates cellulaires (ex. : jeu de la vie de Conway)

démo : golly

- ▶ la **redstone** dans Minecraft
- ▶ ...

D'autres modèles de calcul

- ▶ les fonctions récursives (Kurt Gödel)
- ▶ le λ -calcul (Alonzo Church)
- ▶ machines RAM (modèle de nos ordinateurs, Von Neumann)
- ▶ les automates cellulaires (ex. : jeu de la vie de Conway)

démo : golly

- ▶ la **redstone** dans Minecraft
- ▶ ...
- ▶ **tous les langages de programmation**

D'autres modèles de calcul

- ▶ les fonctions récursives (Kurt Gödel)
- ▶ le λ -calcul (Alonzo Church)
- ▶ machines RAM (modèle de nos ordinateurs, Von Neumann)
- ▶ les automates cellulaires (ex. : jeu de la vie de Conway)

démo : golly

- ▶ la **redstone** dans Minecraft
- ▶ ...
- ▶ **tous les langages de programmation**

un langage de programmation =
la description d'un programme universel

La programmation avant les ordinateurs

années 1940 premiers calculateurs universels en fonctionnement

La programmation avant les ordinateurs

années 1940 premiers calculateurs universels en fonctionnement

années 1930 travaux de Church, Gödel, Turing, *etc.*

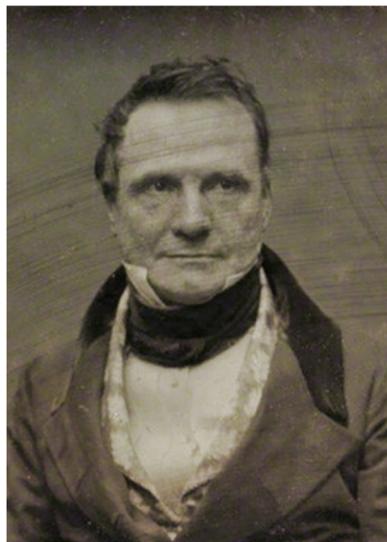
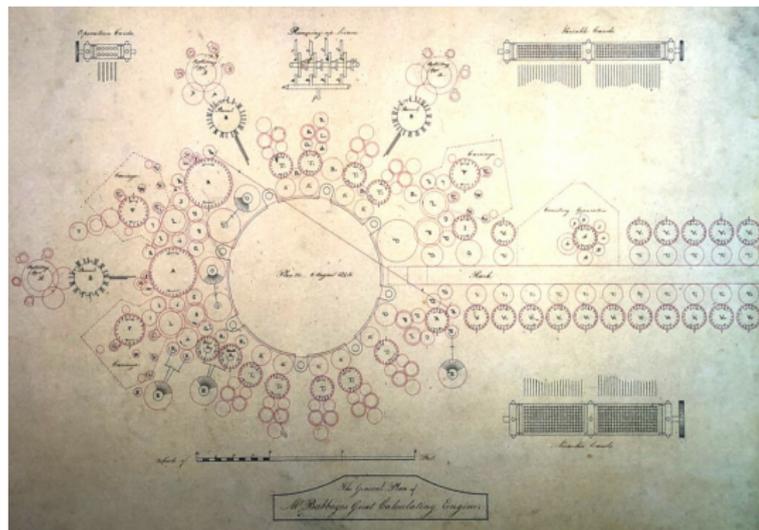
La programmation avant les ordinateurs

années 1940 premiers calculateurs universels en fonctionnement

années 1930 travaux de Church, Gödel, Turing, *etc.*

1843 premier programme informatique : la note G

La machine analytique de Charles Babbage (1791–1871)



Le premier concept (jamais réalisé) d'une machine universelle

gauche : © ArnoldReinhold, CC-BY-4.0, tiré de Wikimedia Commons

droite : domaine public, tiré de Wikimedia Commons

C'est toujours la même chose

Ces modèles décrivent la même notion de calcul!

C'est toujours la même chose

Ces modèles décrivent la même notion de calcul!

démo : parité sur micro:bit

C'est toujours la même chose

Ces modèles décrivent la même notion de calcul!

démo : [parité sur micro:bit](#)

Dans chacun de ces modèles il y a un programme universel
et un programme universel peut en simuler un autre...

démo : [machines virtuelles](#)

Thèse de Church

« C'est la seule bonne notion de calcul. »



Alonzo Church (1903–1995)

I am a Mac / I am a PC...



I am a Mac / I am a PC...



... je suis un téléphone mobile

I am a Mac / I am a PC...



... je suis un téléphone mobile

... je suis un réfrigérateur très haut de gamme

I am a Mac / I am a PC...



- ... je suis un téléphone mobile
- ... je suis un réfrigérateur très haut de gamme
- ... je suis une machine programmable universelle!

Bonus : le problème de l'arrêt

Le problème de l'arrêt

Le problème de l'arrêt pour un modèle de calcul est la fonction :
 $A : (P, E) \mapsto$

$$\begin{cases} 1 & \text{si } P \text{ est un programme dont le calcul sur l'entrée } E \text{ termine} \\ 0 & \text{sinon} \end{cases}$$

Par exemple, en Python, avec :

```
def P(n):  
    s = 0  
    i = 0  
    while i < n:  
        s = i + s  
    return s
```

Que vaut $A(P, 0)$? Que vaut $A(P, 42)$?

Bug?

Théorème

Le problème de l'arrêt n'est pas calculable.

Supposons qu'on ait un programme **A** pour le calculer.

On définit un nouveau programme **P** comme suit :

```
def P(f):  
    if A(f,f):  
        while True:  
            pass  
    else:  
        return 0
```

Alors que calcule **P(P)** ?