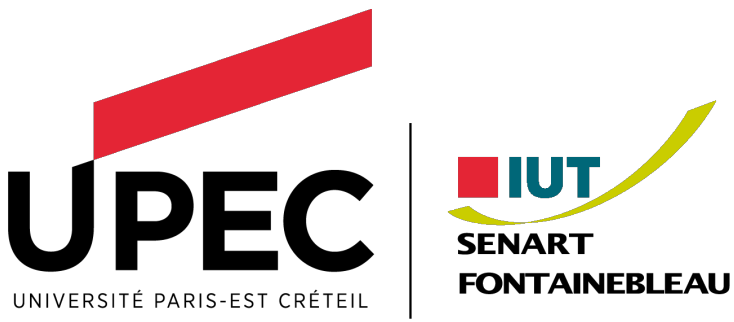
Modèles de calculs

Machine de Turing

Florent Madelaine

Fondements de l'informatique



Plan

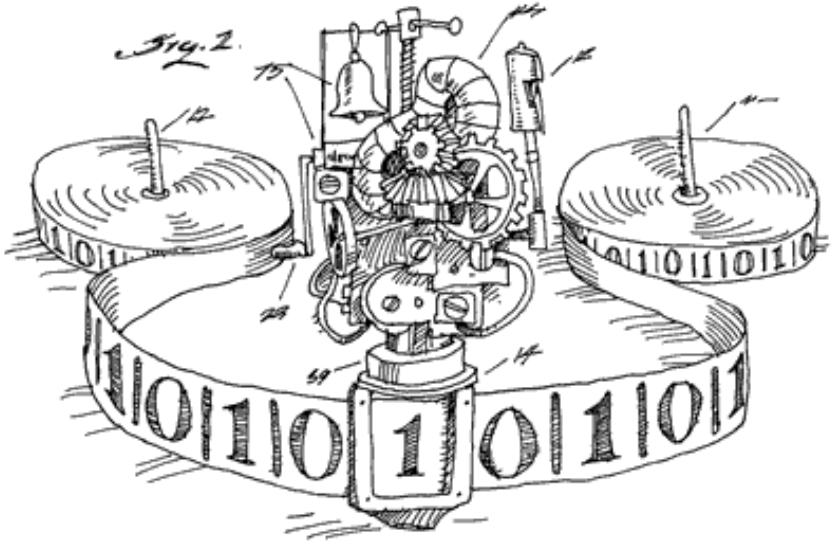
Introduction

Nous avons vu brièvement le modèle des automates finis qui permet de décider les langages réguliers.

Nous avons vu que certains langages ne sont pas réguliers, par exemple le langage $\{a^k b^k$ avec k un entier$\}$ ou encore le langage des palindromes à l'aide du lemme de la pompe.

Nous voyons maintenant le modèle, plus expressif, de la machine de Turing, car il permet de décider ces langages.

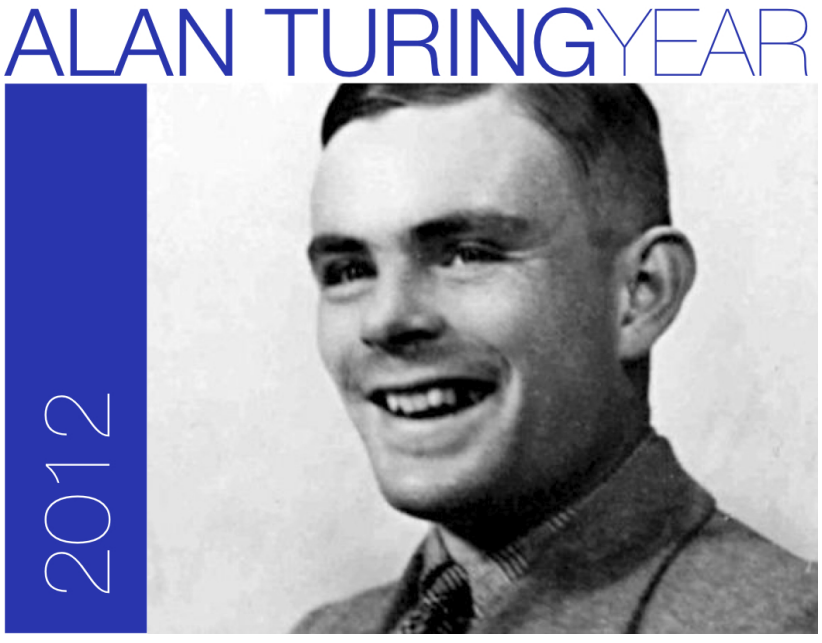
Différence avec les automates



Ressemble à un automate (états, alphabet etc) avec deux différences notables :

- on peut écrire ; et,
- on peut revenir en arrière.

Alan Turing



Il s'agit d'un mathématicien né en 1912, décédé en 1954 qui a marqué l'informatique.

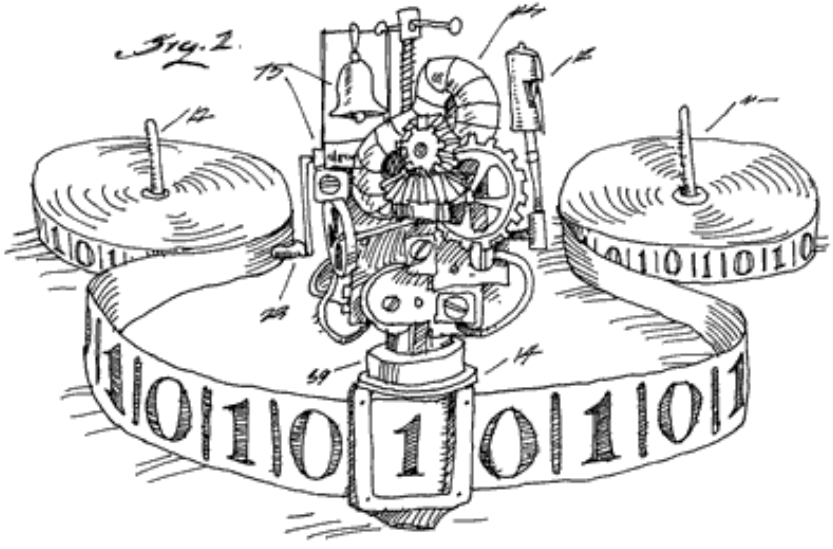
Une loi britannique porte son nom.

https://en.wikipedia.org/wiki/Alan_Turing_law

Vous pouvez par exemple regarder cette vidéo de Bernard Chazelle qui replace la vie de Turing et ses contributions

https://www.youtube.com/watch?v=BYaWoLoEkIQ

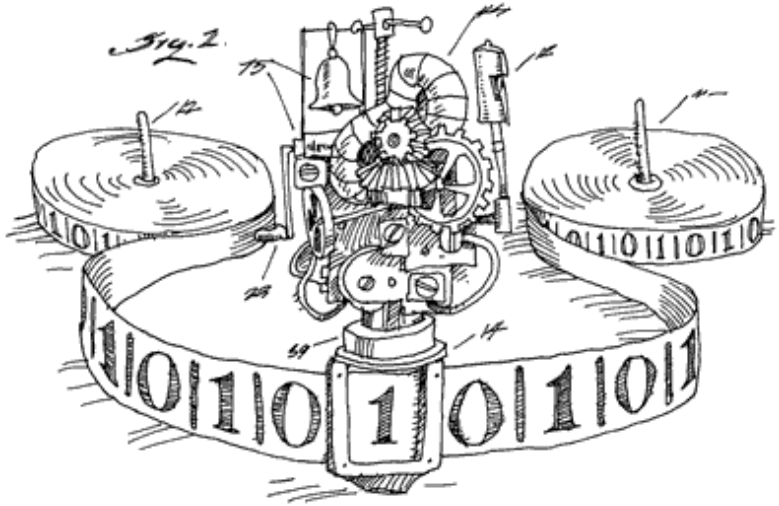
Mémoire non bornée



La machine se présente comme un ruban infini de cellules vides sauf immédiatement à droite de la tête d'écriture et de lecture qui contient le mot d'entrée.

Il est important de noter que puisqu'on peut écrire sur le ruban, on dispose d'une mémoire arbitrairement grande.

Programme



Le programme de la machine de Turing ressemble à celui d'un automate. C'est une fonction de transition.

La différence notable c'est qu'on peut écrire et que la tête peut bouger autrement que vers la droite du mot. Quand on lit un symbole dans un certain état, il faut non seulement indiquer le futur état (comme pour l'automate) mais en plus indiquer le symbole qu'on va écrire et le déplacement de la tête.

La fonction de transition associe donc à une paire (état, lettre lue) un triplet (nouvel état, lettre écrite, déplacement).

Le programme est de taille fini

La fonction de transition associe à une paire (état, lettre lue) un triplet (nouvel état, lettre écrite, déplacement).

Comme les choix possibles sont finis (on a un nombre fini d'états, de symboles, et seulement 2 mouvements possibles), la fonction de transition est forcément de taille finie.

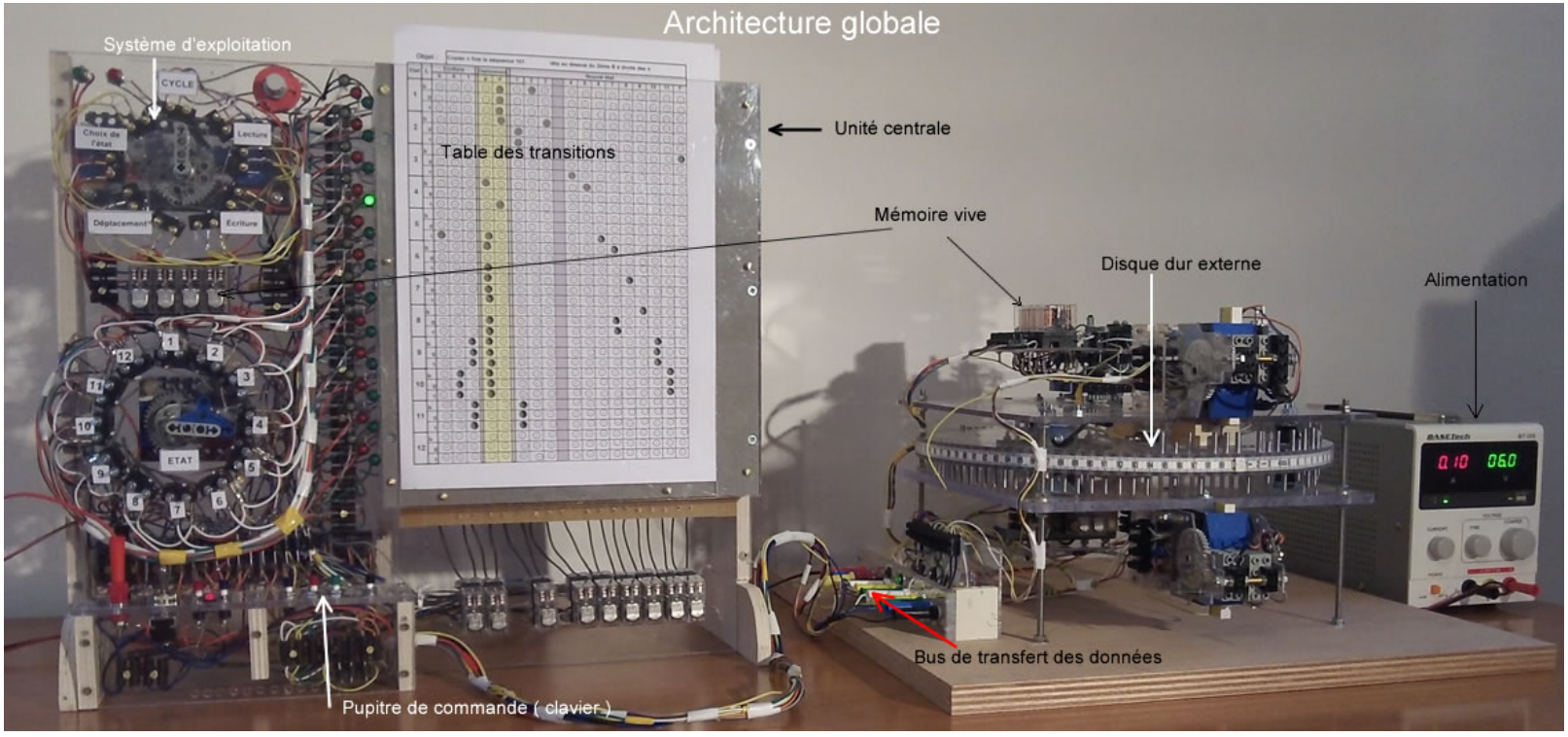
Machines de Turing construites pour de vrai

On verra que n'importe quel ordinateur est essentiellement une machine de Turing.

Plus prosaïquement, de nombreuses personnes ont fabriqué des machines qui fonctionnent exactement comme celle de Turing. En particulier, le père d'un collègue de Clermont Ferrand Marc Raynaud s'est attaché à montrer que Turing aurait pu vraiment construire sa machine en 1936.

https://machinedeturing.com/videos.php?page=17

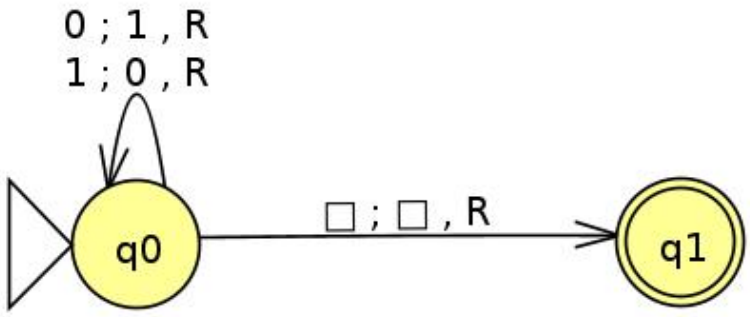
Machine de Raynaud



Écrire

On peut écrire le mot d'entrée en inversant les 0 et les 1.

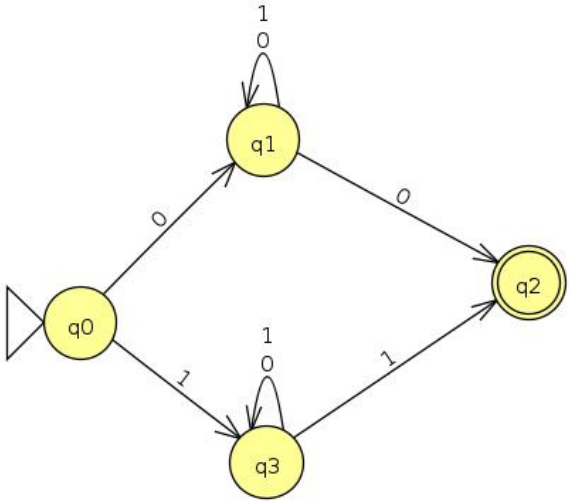
Un automate qui peut écrire – on parle de transducteur – pourrait faire la même chose.



Faire comme un automate

On peut toujours faire comme un automate. Il y a juste un truc nécessaire pour détecter la fin du mot.

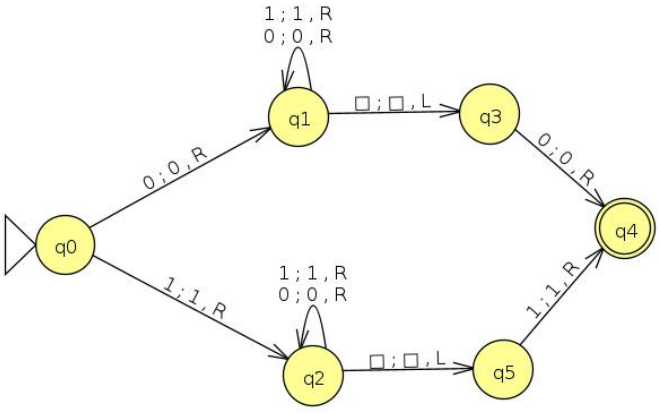
Par exemple pour les mots qui commencent et terminent par la dernière lettre.



Faire comme un automate

Par exemple pour les mots qui commencent et terminent par la dernière lettre.

Même chose avec une machine de Turing.

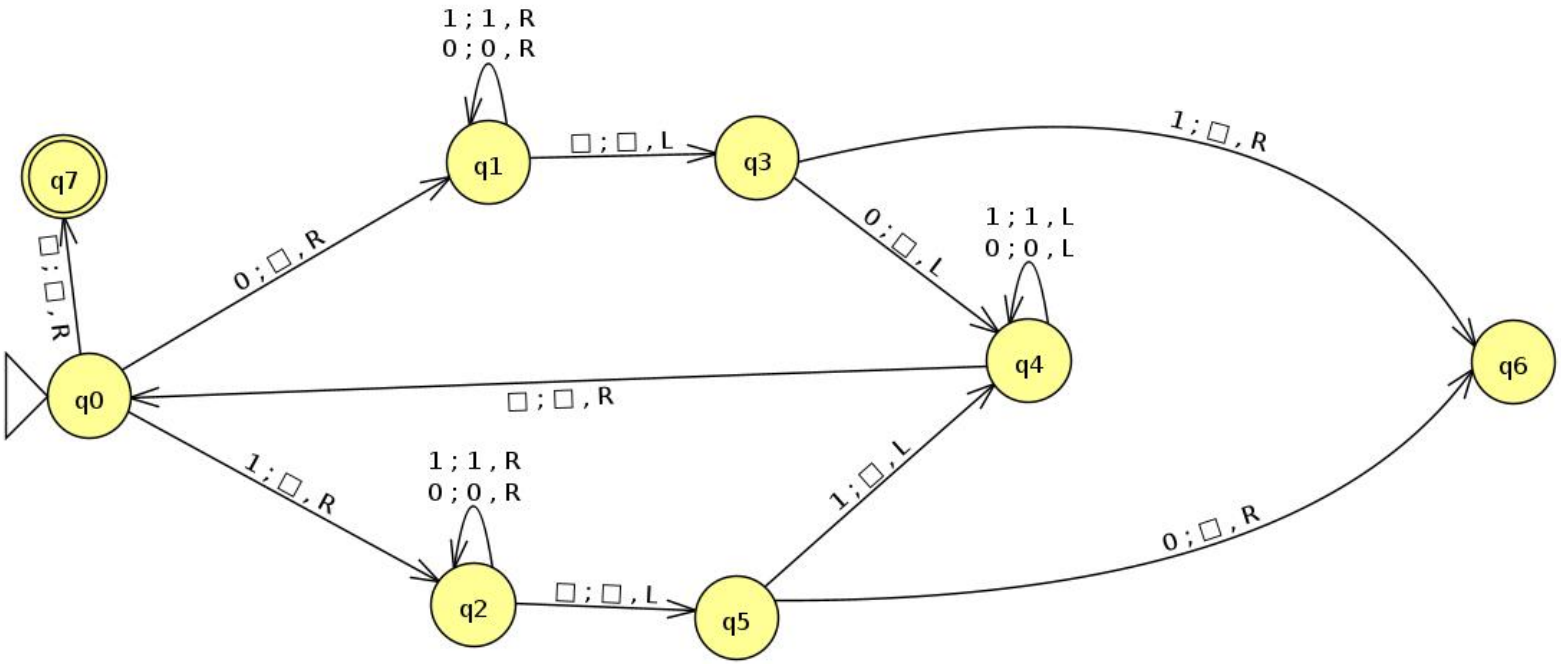


Palindrome

Par contre le modèle de Turing est plus fort si on écrit et qu'on déplace la tête vraiment vers la gauche (pas seulement pour détecter la fin du mot).

On va voir qu'on peut décider le langage des palindromes pairs, c'est-à-dire les mots qui sont de la forme un mot w suivi de w écrit à l'envers.

MdT pour palindromes pairs



Turing permet de faire quelques manipulations simples

En admettant qu'on peut simuler une machine à plusieurs rubans avec une machine à un ruban, on peut facilement avec plusieurs tubans :

- recopier un ruban sur un autre
- incrémenter la valeur écrite sur un ruban (faire +1)
- ajouter deux rubans sur un troisième ruban
- tester si un ruban est plus petit qu'un autre

On peut voir chaque ruban comme le contenu d'une variable.

Combiner des sous-programmes

On peut combiner divers sous-programmes pour fabriquer une MdT avec un plus gros programme.

Par exemple on peut faire quelque chose comme une boucle :

- Pour i=1 à n faire
- un sous programme

Turing permet de simuler n'importe quel calcul

Il existe en suite des subtilités plutôt liées à la compilation de langages de haut niveau (en particulier dans un vrai langage de programmation il n'y a pas de limite aux appels successifs qu'on peut faire).

En pratique, il est plus simple de montrer qu'on peut simuler des machines à registre avec une machine de Turing puisque un cours classique de compilation montre comment on peut compiler un langage de haut niveau en assembleur.

Programme vs Données

Pour l'instant notre MdT n'accède pas de la même façon à son programme (caché dans les maths de la définition) et à ses données qui elles sont sur les rubans.

Dans une machine moderne, il n'y pas fondamentalement de différence entre programme et données. En particulier, une fois « chargées » des données peuvent devenir un programme.
Cette idée existe déjà chez Turing : c'est la machine universelle.

La thèse de Church Turing

Informellement

Pour tout modèle raisonnable de calcul, on obtient la même notion de ce qui est calculable.

Arguments en faveur de cette thèse

- Théorie générales des fonctions récursives (Gödel, Herbrand 1933)
- λ-calcul (Church, 1936)
- Machines de Turing (Turing, 1936)
- Trois modèles équivalents (Church 1936, Turing 1937).
- Machines de Post (1936)
- Machines de Turing avec plusieurs rubans (Minsky)
- Machines à compteurs
- Machines à registre
- ...

La thèse de Church Turing

Informellement

Pour tout modèle raisonnable de calcul, on obtient la même notion de ce qui est calculable.

Arguments en faveur de cette thèse

- Théorie générales des fonctions récursives (Gödel, Herbrand 1933)
- λ-calcul (Church, 1936)
- Machines de Turing (Turing, 1936)
- Trois modèles équivalents (Church 1936, Turing 1937).
- Machines de Post (1936)
- Machines de Turing avec plusieurs rubans (Minsky)
- Machines à compteurs
- Machines à registre
- ...

Limite de la machine de Turing ?

Nous avons entrevu le modèles de Turing. Nous avons vu qu'il permet de décider un langage qui n'est pas régulier.

Nous avons argumenté que tout ce qui peut se calculer peut l'être avec une MdT (même si ce n'est pas pratique pour programmer en pratique évidemment).

Nous verrons les limites de ce modèle.