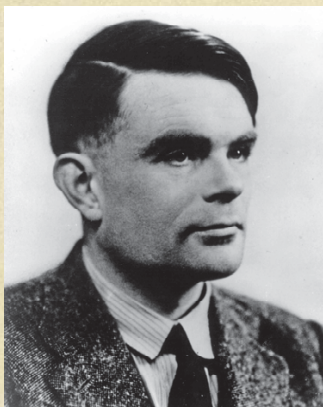


Machine de Turing

Profs. Peña & Perez-Uribe & Mosqueron

Basé sur la leçon du Prof. A. Perez-Uribe

Alan Turing (1912-1954)



- Mathématicien britannique, considéré comme le père de l'informatique moderne.
- Dans son article "On Computable Numbers with an Application to the Entscheidungsproblem" (1936), il présenta sa "machine de Turing", le premier calculateur universel programmable.
- Pendant la Seconde guerre mondiale, il travaille sur les codes secrets générés par la machine Enigma utilisée par les Nazis.

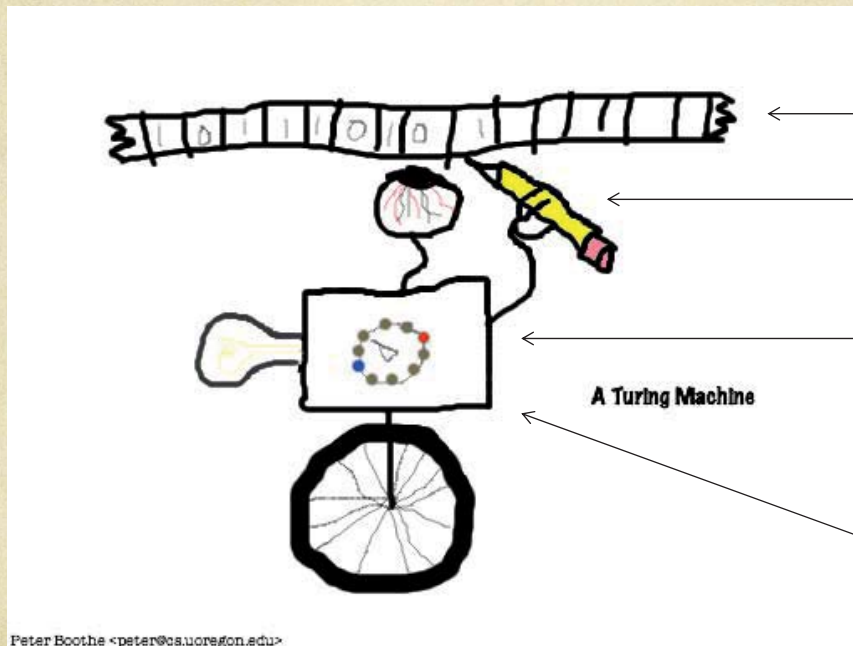
Ordinateurs à programme stocké

- Après la guerre, il travaille sur un des tous premiers ordinateurs, l'ACE (Automatic Computing Engine) qu'il décrit dans un rapport en 1945. La particularité de cette machine est que son comportement est contrôlé par un *programme stocké* dans l'ordinateur lui même.
- La plupart des ordinateurs sont des ordinateurs à *programme stocké en mémoire*, aussi appelés *ordinateurs de von Neumann*, d'après le mathématicien John von Neumann, qui décrit cette architecture d'ordinateurs dans le "*First Draft of a Report on the EDVAC*" (1945).

"Grand-père" de l' IA

- Alan Turing est aussi considéré comme le "grand-père" de l'Intelligence artificielle. Il publia un article en 1950 intitulé "*Computing machinery and intelligence*" et posa la question "*une machine, peut-elle penser ?*"
- Il définit un test, connu sous le nom *Test de Turing* qui doit déterminer si une machine peut être considérée comme intelligente.
(<http://www.conishiwa.org/zones/projets/ia/turing.htm>)
- Le père de l'IA est John McCarthy, dans le sens que c'est lui qui inventa le nom lors d'un meeting avec d'autres chercheurs, tels que Marvin Minsky et Claude Shannon entre autres.

Machine de Turing



ruban
tête de lecture/écriture

registre d'état

table d'actions:
- symbole à écrire
- bouger à G/D
- état futur

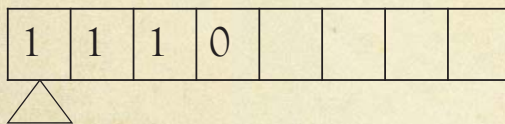
Peter Boothe <peter@cs.uoregon.edu>

ARO1 - 2016 - APE & CPN & RMQ

5

turingmachinesimulator.com/ Exemple d'une Machine de Turing

Cette Machine de Turing lit une séquence de '1 et de 0' dans son ruban, et teste si le nombre de 0 est paire. La machine comporte 2 états q_0 et q_1 .



state, read char
new state, write char, dir

- = halt
_ = don't care

$q_0, 0$
 $q_1, 0, >$

$q_1, 0$
 $q_0, 0, >$

$q_0, 1$
 $q_0, 1, >$

$q_1, 1$
 $q_1, 1, >$

$q_0, _$
 $q_{Accept}, _, -$

ARO1 - 2016 - APE & CPN & RMQ

6

State: q0 Steps: 0

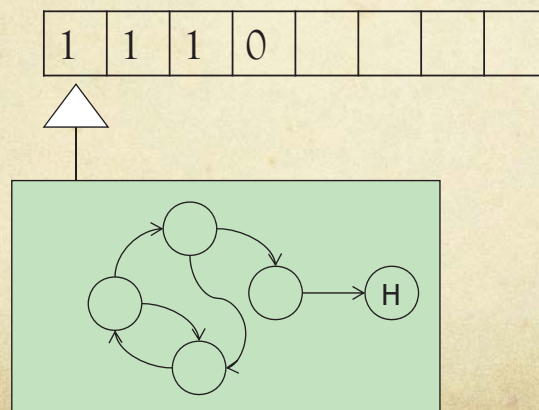
1 0 1 1 0 1

input: 101101 Load ▶ || ■ ▶▶ Speed:

Introduction	MACHINE	Examples
What is a Turing machine?	<pre> 10 // States 11 // q0 amount of 0s mod2 == 0 12 // q1 amount of 0s mod2 == 1 13 // qAccept - accepting state 14 //----- 15 16 name: Even amount of zeros 17 init: q0 18 accept: qAccept 19 20 q0,0 21 q1,0,> 22 23 q1,0 24 q0,0,> 25 26 q0,1 27 q0,1,> 28 29 q1,1 30 q1,1,> 31 32 q0,_ 33 qAccept,_,- </pre>	1 tape 2 tapes 3 tapes
Tutorial		Options
Programming example		Number of visible cells <input type="text" value="25"/>
Multiple tapes		More
The Theory		About
Formal Languages		Contact me!
Decision Problems		Log in to save online! Share link
Why are TM's important?		COMPILE
Computational Complexity		
Formalization		

ARO1 - 2016 - APE & CPN & RMQ

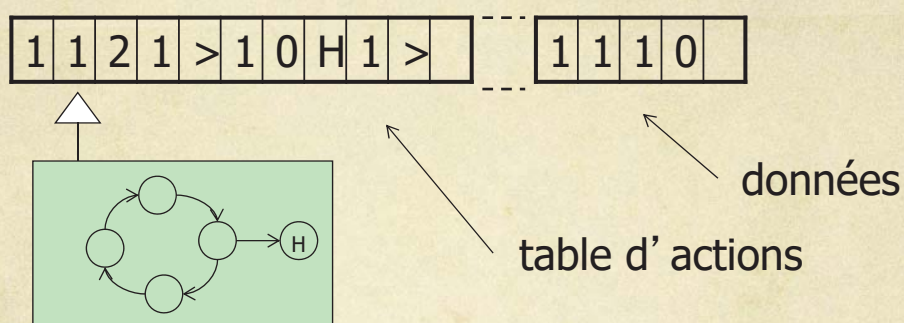
- Le registre d'état et la table d'états peuvent être réalisés par une machine à états finis



* H = halt 8

- Toute machine de Turing calcule le résultat d'une fonction partielle sur des chaînes de caractères composées des caractères de son alphabet.
- En ce sens, une machine de Turing se comporte comme un ordinateur avec un programme déterminé.
- Turing démontra que la table d'actions d'une machine de Turing peut être codé sous la forme d'une chaîne de caractères.

Machine de Turing universelle



- Une machine de Turing qui suppose l'existence sur son ruban d'une chaîne de caractères encodant une table d'actions, suivie d'une chaîne de caractères constituant les données effectives du ruban, et calcule le contenu du ruban que la machine de Turing encodée aurait calculé, est une “**machine de Turing universelle**”.

Vers la plus petite Machine de Turing Universelle

- Une machine de Turing universelle peut simuler le comportement de n'importe quelle autre machine de Turing.
- Mais, une machine de Turing universelle peut être relativement simple, et utiliser seulement quelques états et symboles. On a conçu des machines de Turing universelles de taille (symboles x états): 4x7 (M. Minsky), 4x5, 18x2, 5x2 (S. Wolfram), etc.

11

Encore un exemple

Une MT qui détecte les palindromes

Exemple:



0	1	1	0	1	0	1	1	0	=
---	---	---	---	---	---	---	---	---	---

12

Binary palindrome

State: q0 Steps: 0

0 1 1 0 1 0 1 1 0

input: 011010110 ▶ || ■ ▶▶ Speed:

Introduction	MACHINE	Examples	
What is a Turing machine?	<pre> 1 // Input: a binary number n 2 // Output: accepts if n is a palindr 3 // Example: accepts 10101 4 // 5 // Palindrome Algorithm 6 // for Turing Machine Simulator 7 // (http://martinugarte.com/turingm 8 9 name: Binary palindrome 10 init: q0 11 accept: qAccept 12 13 q0,0 14 qRight0,_,> 15 16 qRight0,0 17 qRight0,0,> 18 19 qRight0,1 20 qRight0,1,> 21 22 q0,1 23 qRight1,_,> 24 25</pre>	1 tape 2 tapes 3 tapes	
Tutorial			
Programming example			
Multiple tapes			
The Theory			
Formal Languages			
Decision Problems			
Why are TM's important?			
Computational Complexity			
Formalization			

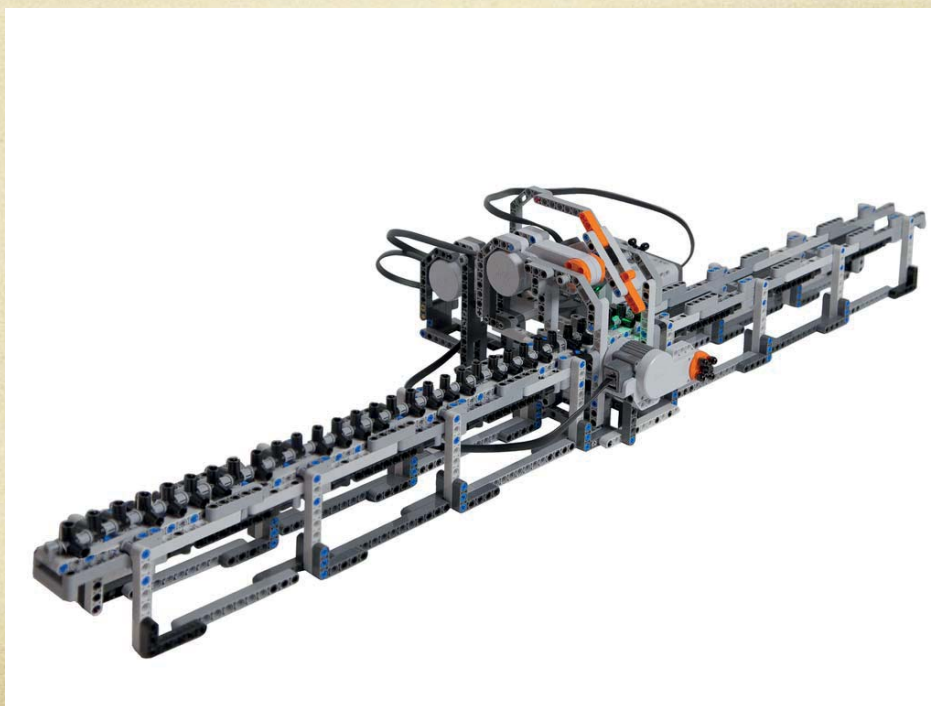
Log in to save online!

ARO1 - 2016 - APE & CPN & RMQ

3

Une machine de Turing en Lego

<http://www.legoturingmachine.org/>



14

ARO1 - 2016 - APE & CPN & RMQ