



LYCÉE LECONTE DE LISLE

Machine de Turing

Vincent Picard

Introduction

- On veut savoir quelles fonctions sont **calculables** par un ordinateur, il faut pour cela se donner un modèle général de ce qu'est une **machine** et un **calcul**. Plusieurs modèles ont été proposés :
 - ▶ 1933 : R. Péter, K. Gödel et J. Herbrand proposent les **fonctions μ -récursives**
 - ▶ 1936 : Alonzo Church propose le **λ -calcul**
 - ▶ 1936 : Alan Turing propose la **machine de Turing**
- Church, Turing et Stephen Kleene démontrent les résultats suivants :

Une fonction est μ -récurcive ssi elle est λ -calculable ssi elle Turing-calculable

■ Thèse de Church-Turing

Toute fonction calculable par machine l'est au sens de l'un des ces 3 modèles équivalents.

Les visages de la calculabilité



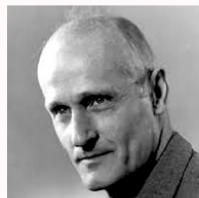
Rózsa
Péter



Kurt
Gödel



Jacques
Herbrand



Stephen
Kleene



Alonzo
Church

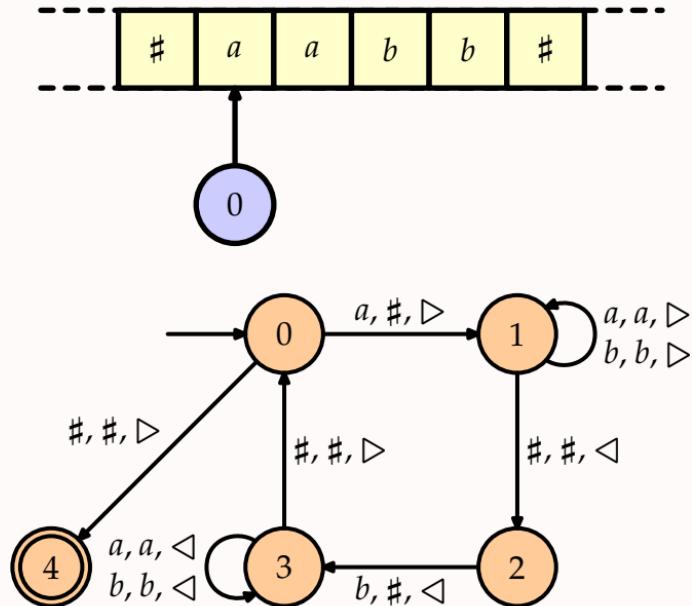


Alan Turing

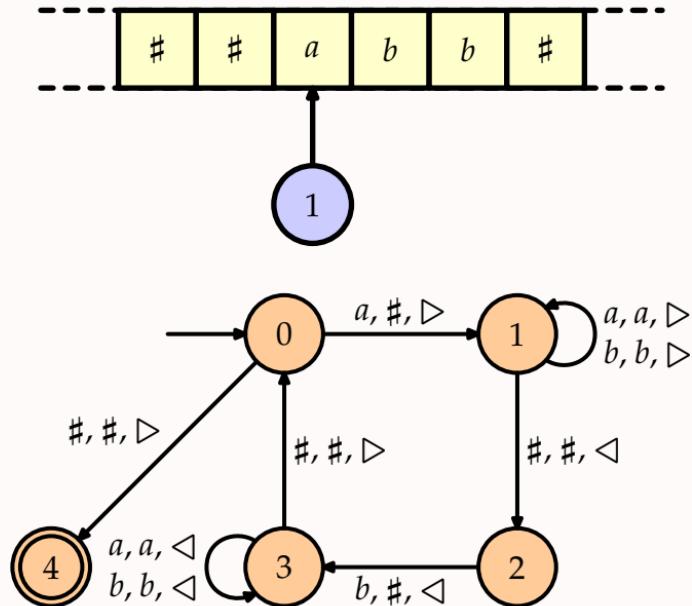
Machine de Turing : intuition

- Intuitivement une machine de Turing possède une **mémoire infinie** modélisée sous forme d'un **ruban bi-infini de cases mémoire**.
- Il existe une **tête de lecture** positionnée sur l'une des cases du ruban.
- La machine est une **machine à états** qui lorsqu'elle se situe dans un état q :
 - ▶ lit le caractère x situé sous la tête de lecture
 - ▶ selon sa **fonction de transition** :
 1. remplace le symbole x par un symbole y (éventuellement $y = x$)
 2. déplace la tête de lecture à gauche ou à droite
 3. transite vers un état q' (éventuellement $q' = q$)

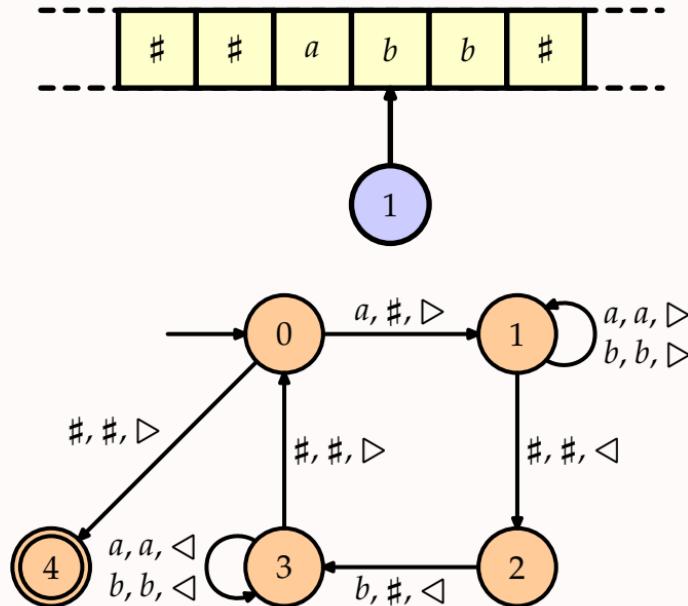
Machine de Turing : exemple



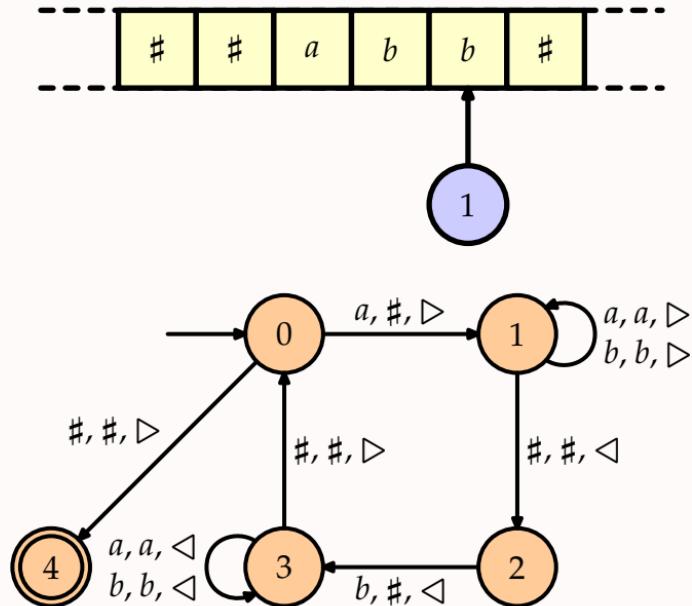
Machine de Turing : exemple



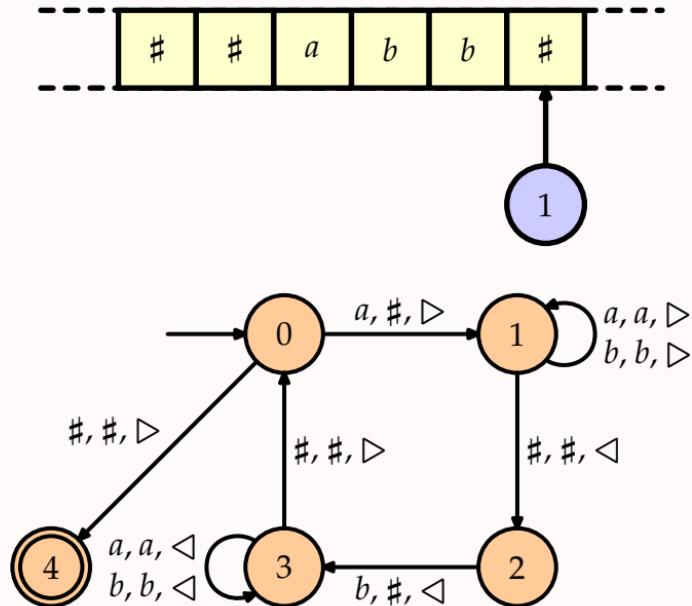
Machine de Turing : exemple



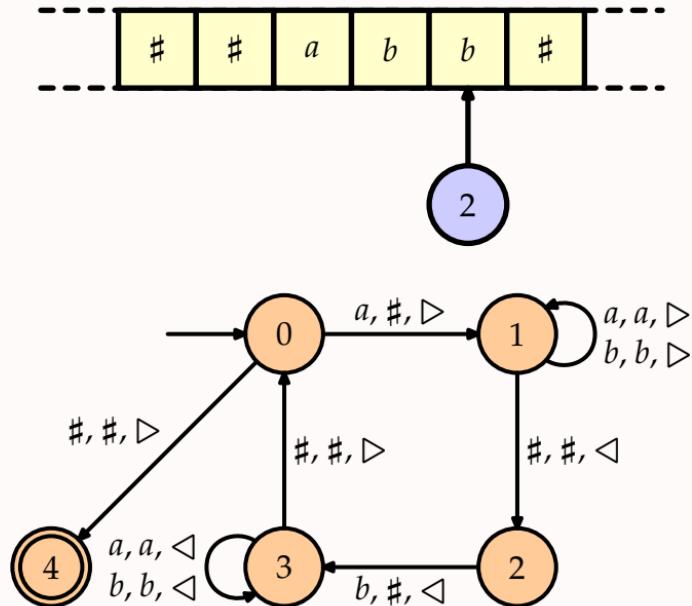
Machine de Turing : exemple



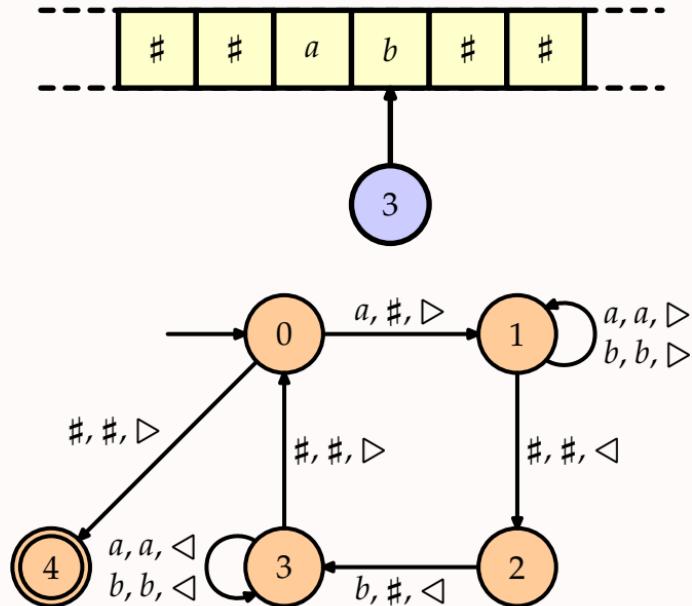
Machine de Turing : exemple



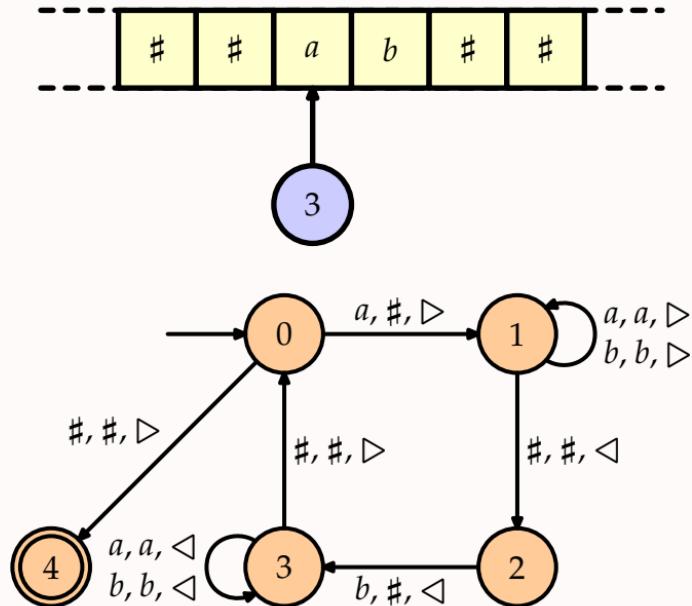
Machine de Turing : exemple



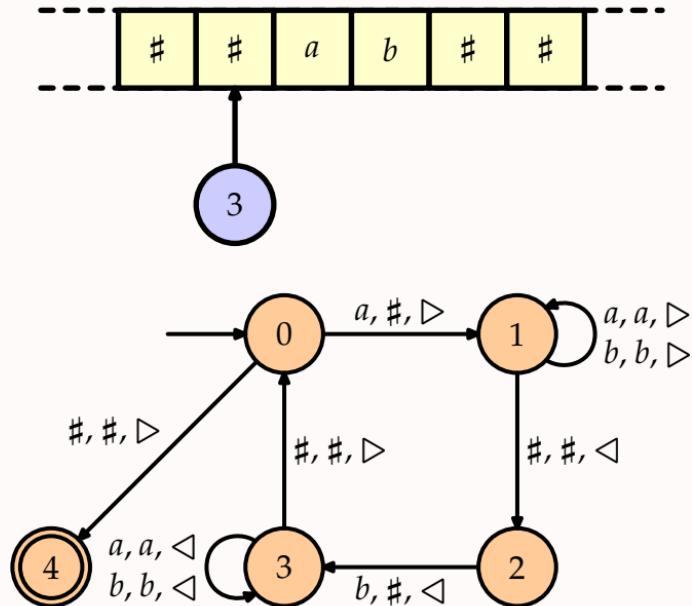
Machine de Turing : exemple



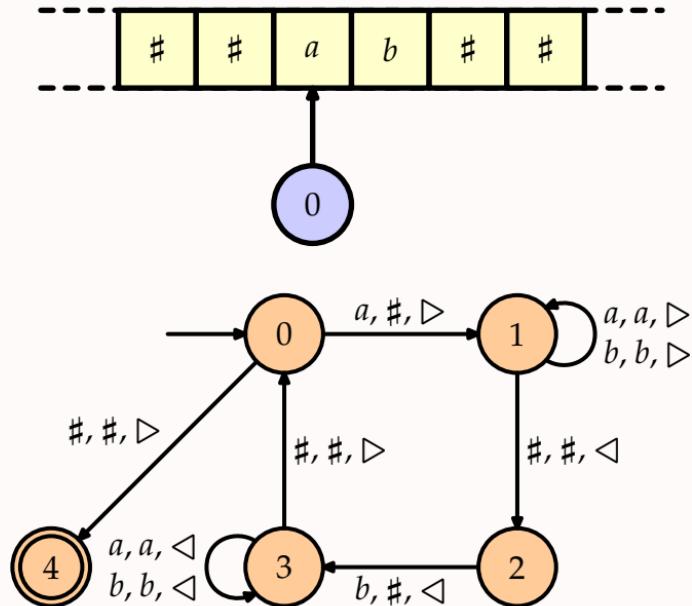
Machine de Turing : exemple



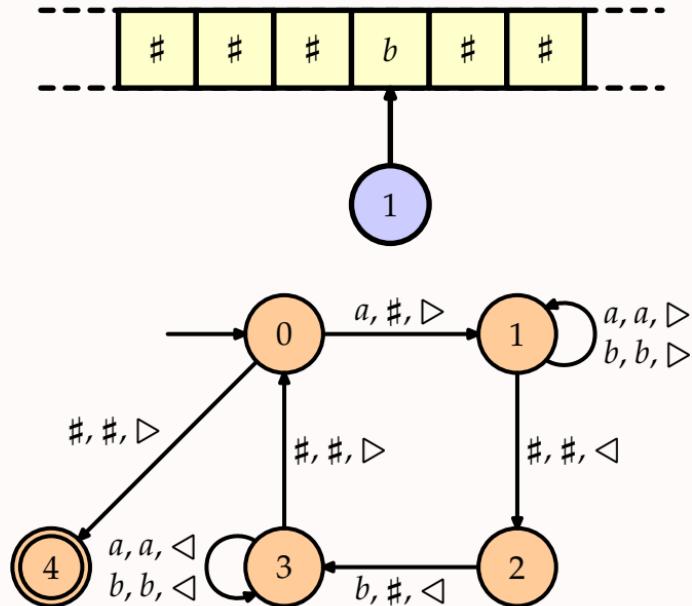
Machine de Turing : exemple



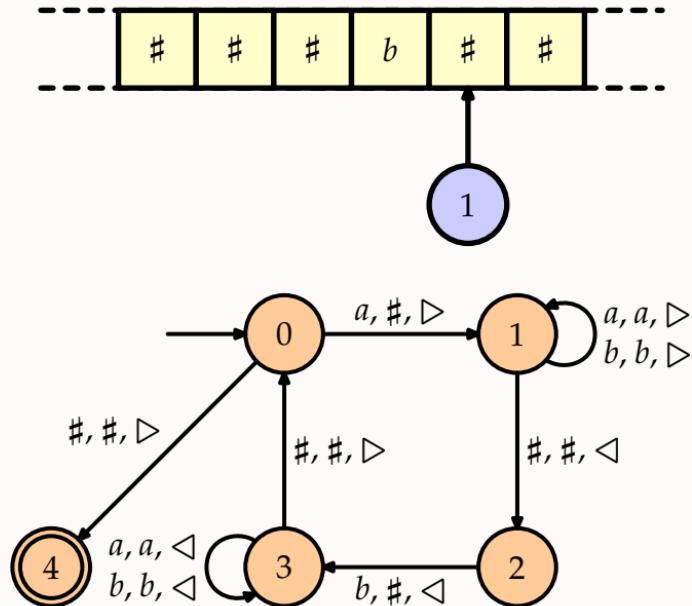
Machine de Turing : exemple



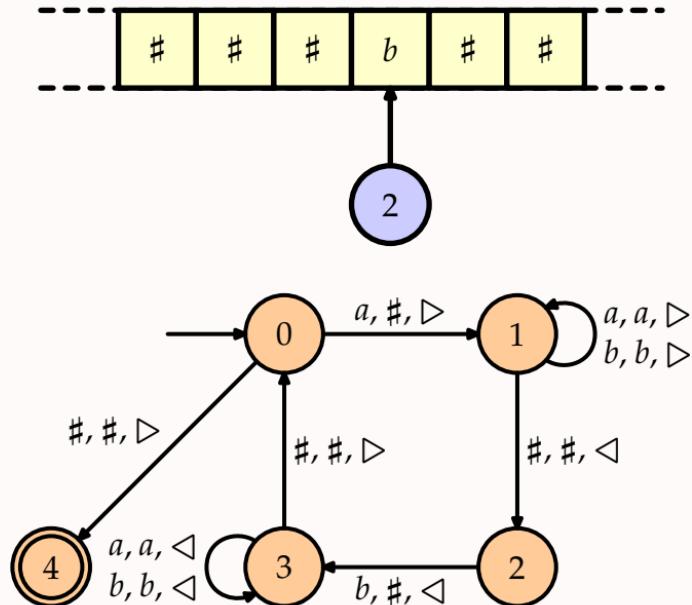
Machine de Turing : exemple



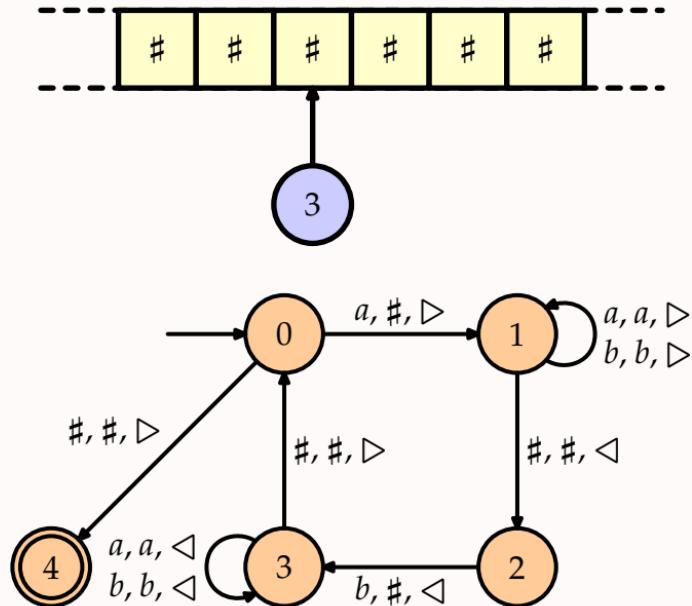
Machine de Turing : exemple



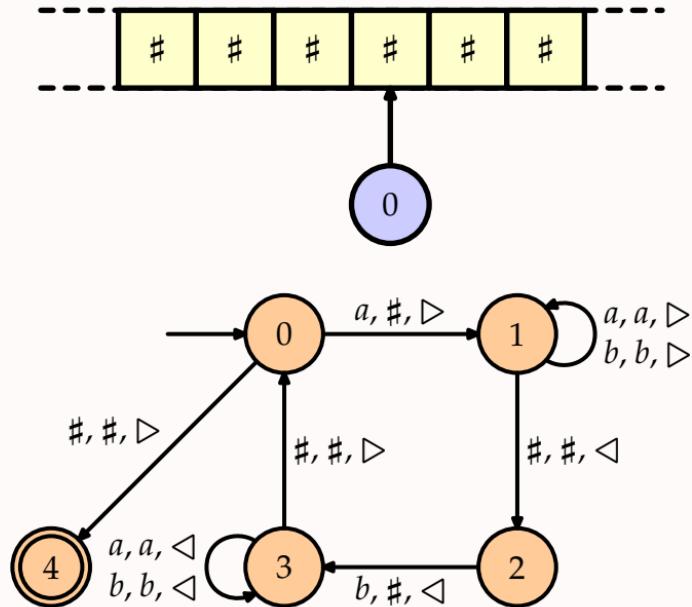
Machine de Turing : exemple



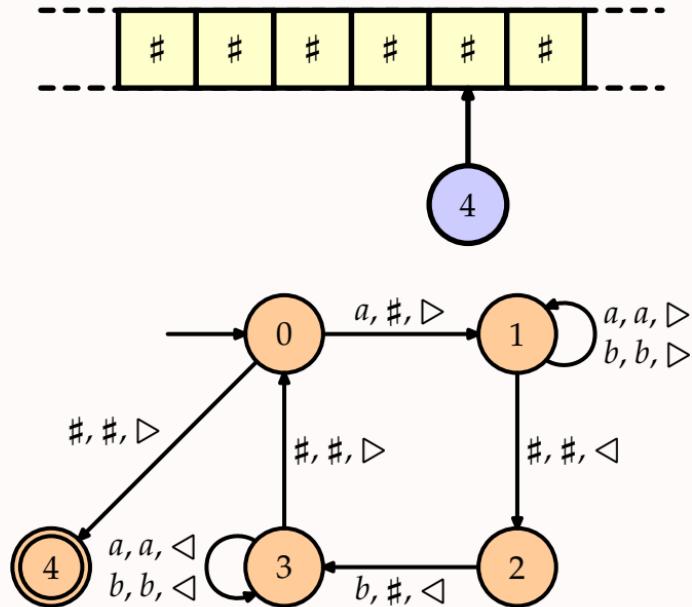
Machine de Turing : exemple



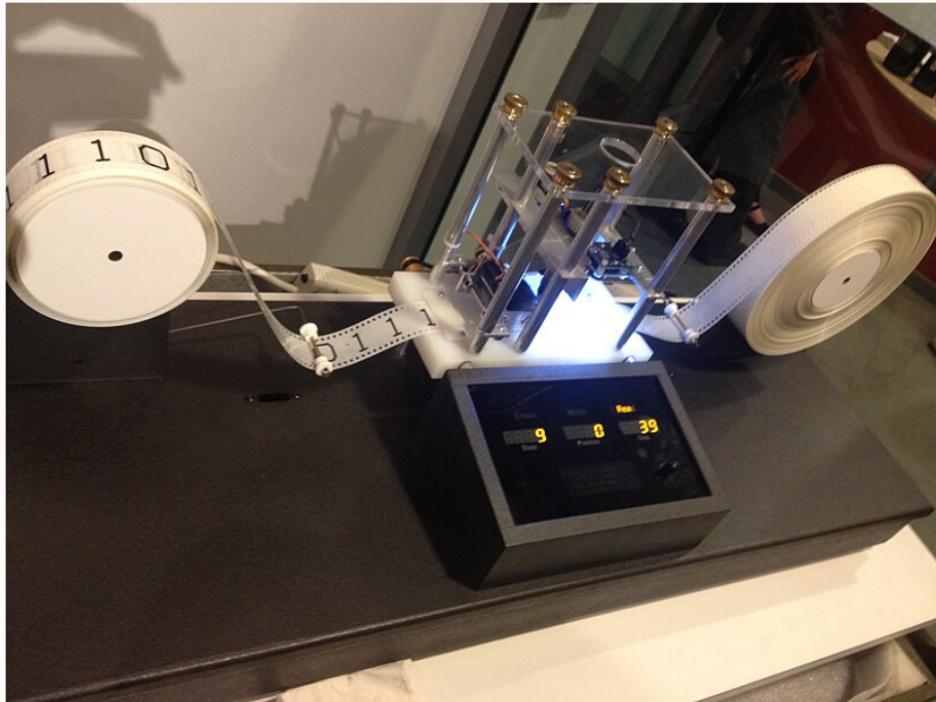
Machine de Turing : exemple



Machine de Turing : exemple

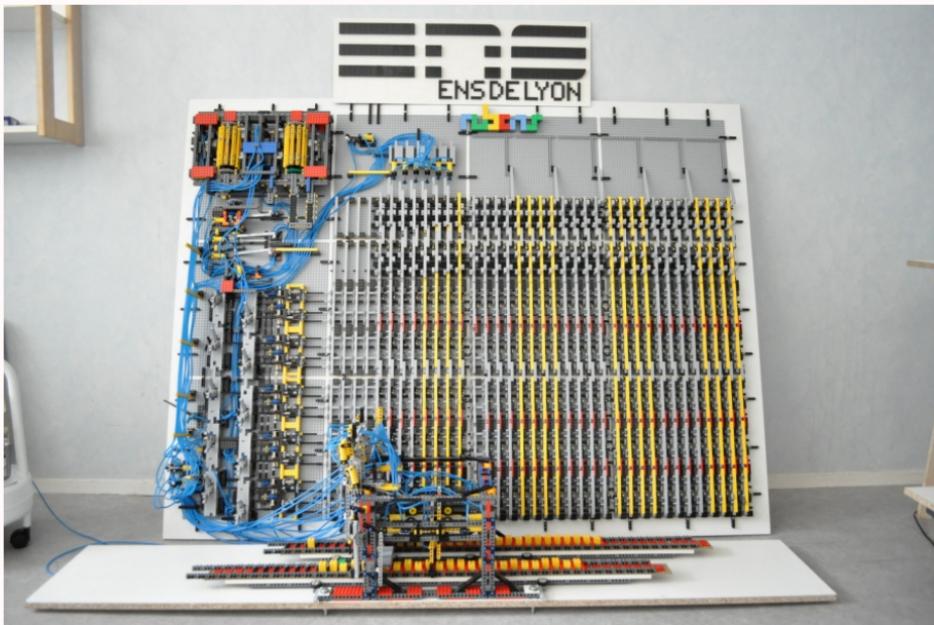


Machine de Turing réelles



Harvard Collection of Historical Scientific Instruments (Wiki commons)

Machine de Turing réelles



Machine de Turing en Lego de l'ENS Lyon (Wiki commons)

Machine de Turing

- Une **machine de Turing déterministe** est un 7-uplet $(Q, \Gamma, \#, \Sigma, \delta, q_0, F)$ où :
 - ▶ Q est un ensemble fini d'**états**
 - ▶ Γ est l'**alphabet de ruban**
 - ▶ $\# \in \Gamma$ est un symbole appelé **symbole blanc**
 - ▶ $\Sigma \subset \Gamma \setminus \{\#\}$ est l'**alphabet d'entrée**
 - ▶ $\delta : (Q \setminus F) \times \Gamma \rightarrow Q \times \Gamma \times \{\triangleleft, \triangleright\}$ est la **fonction de transition**
 - ▶ $q_0 \in Q$ est l'**état initial**
 - ▶ $F \subset Q$ est l'ensemble des **états finaux**
- La notion de **blocage** existe lorsque δ n'est pas définie. Il y a toujours blocage sur un état final¹.

¹ ce n'est pas indispensable mais ça ne change rien à la théorie de le supposer

Définitions alternatives

- On trouve dans la littérature de nombreuses définitions alternatives qui ne changent en rien la puissance de calcul de la machine.
- On peut par exemple :
 - ▶ Considérer que le ruban n'est infini que d'un côté.
 - ▶ Ajouter l'action ∇ qui correspond à ne pas déplacer la tête de lecture.
 - ▶ Avoir une machine avec plusieurs rubans ayant chacun sa tête de lecture.
- Ces définitions alternatives peuvent faciliter la conception de machines de Turing réalisant une tâche donnée.

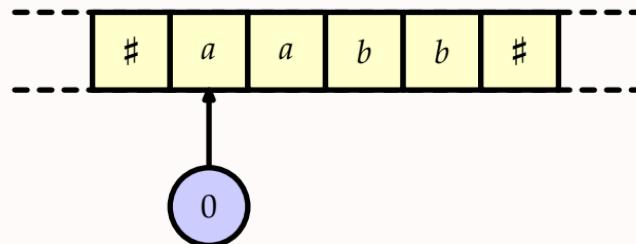
Configurations

- La **configuration** d'une machine de Turing désigne l'ensemble de tous ses paramètres courants :
 - ▶ Le contenu du ruban
 - ▶ La position de la tête de lecture
 - ▶ L'état actuel
- Formellement une configuration est un triplet (u, q, v) dans :
$$\{\varepsilon\} \cup (\Gamma \setminus \{\#\})\Gamma^* \quad \times \quad Q \quad \times \quad \{\varepsilon\} \cup \Gamma^*(\Gamma \setminus \{\#\})$$
 - ▶ u est le mot strictement à gauche de la tête de lecture
 - ▶ v est le mot à droite de la tête de lecture
 - ▶ la tête de lecture est placée sur la première lettre de v (si $v \neq \varepsilon$)
- Initialement on place le mot d'entrée $w \in \Sigma^*$ sur le ruban et la tête de lecture sur le premier symbole de ce mot la configuration initiale est donc $t_w = (\varepsilon, q_0, w)$.

Calcul d'une machine de Turing

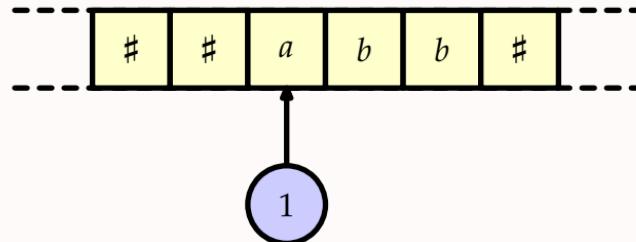
- Une **étape de calcul** d'une machine de Turing est une paire de configurations (C, C') notée $C \rightarrow C'$ telle que
 - ▶ $C = (u, q, av)$ et $C' = (ub, q', v)$ si $\delta(q, a) = (q', b, \triangleright)$
 - ▶ $C = (uc, q, av)$ et $C' = (u, q', cbv)$ si $\delta(q, a) = (q', b, \triangleleft)$
- On ajoute aussi les cas limites :
 - ▶ $C = (u, q, \varepsilon)$ et $C' = (ub, q', \varepsilon)$ si $\delta(q, \#) = (q', b, \triangleright)$
 - ▶ $C = (uc, q, \varepsilon)$ et $C' = (u, q', cb)$ si $\delta(q, \#) = (q', b, \triangleleft)$
 - ▶ $C = (\varepsilon, q, \varepsilon)$ et $C' = (\varepsilon, q', \#b)$ si $\delta(q, \#) = (q', b, \triangleleft)$
- Remarque : si dans la configuration $C' = (s, q', t)$ d'arrivée s commence par des $\#$ on les supprime, de même si t termine par des $\#$.

Calcul : exemple



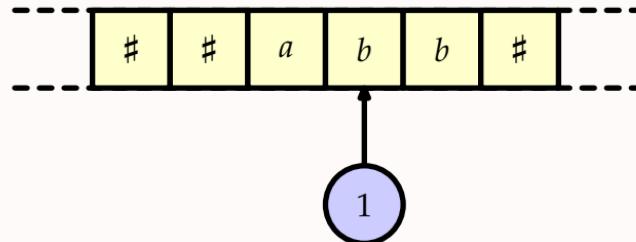
$(\varepsilon, 0, aabb)$

Calcul : exemple



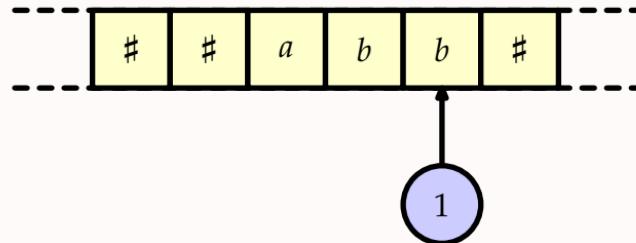
$$(\varepsilon, 0, aabb) \rightarrow (\varepsilon, 1, aab)$$

Calcul : exemple



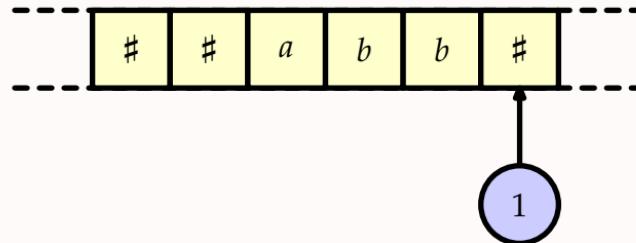
$(\varepsilon, 0, aabb) \rightarrow (\varepsilon, 1, aab) \rightarrow (a, 1, ab)$

Calcul : exemple



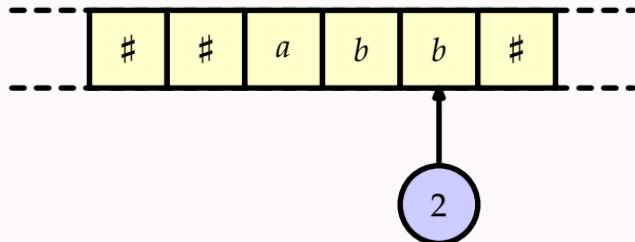
$(\varepsilon, 0, aabb) \rightarrow (\varepsilon, 1, aab) \rightarrow (a, 1, ab) \rightarrow (ab, 1, b)$

Calcul : exemple



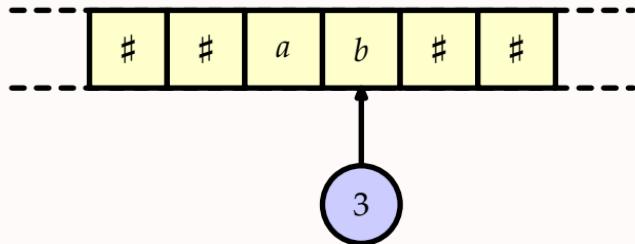
$(\varepsilon, 0, aabb) \rightarrow (\varepsilon, 1, aab) \rightarrow (a, 1, ab) \rightarrow (ab, 1, b) \rightarrow (abb, 1, \varepsilon)$

Calcul : exemple



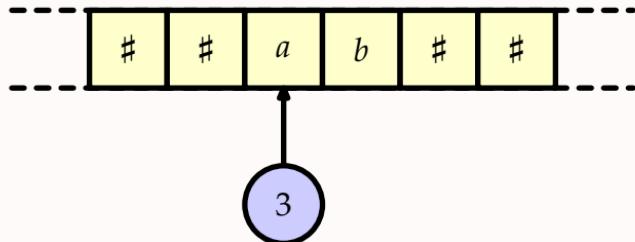
$(\varepsilon, 0, aabb) \rightarrow (\varepsilon, 1, aab) \rightarrow (a, 1, ab) \rightarrow (ab, 1, b) \rightarrow (abb, 1, \varepsilon)$
 $\rightarrow (ab, 2, b)$

Calcul : exemple



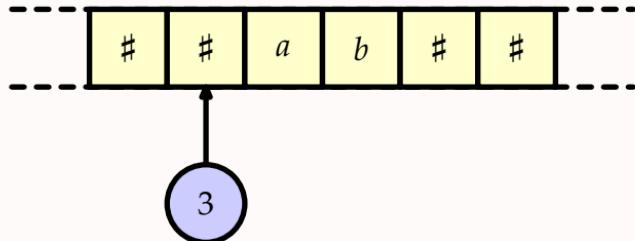
$(\varepsilon, 0, aabb) \rightarrow (\varepsilon, 1, aab) \rightarrow (a, 1, ab) \rightarrow (ab, 1, b) \rightarrow (abb, 1, \varepsilon)$
 $\rightarrow (ab, 2, b)$
 $\rightarrow (a, 3, b)$

Calcul : exemple



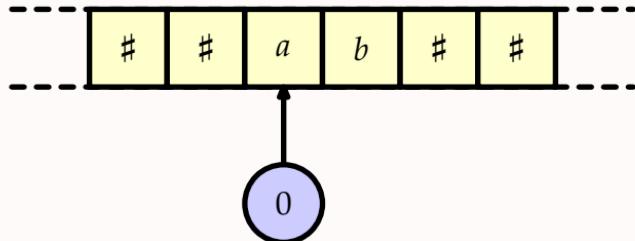
$(\varepsilon, 0, aabb) \rightarrow (\varepsilon, 1, aab) \rightarrow (a, 1, ab) \rightarrow (ab, 1, b) \rightarrow (abb, 1, \varepsilon)$
 $\rightarrow (ab, 2, b)$
 $\rightarrow (a, 3, b) \rightarrow (\varepsilon, 3, ab)$

Calcul : exemple



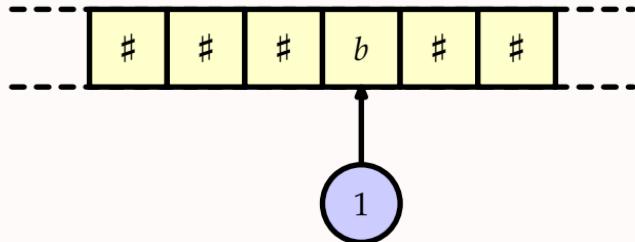
$(\varepsilon, 0, aabb) \rightarrow (\varepsilon, 1, aab) \rightarrow (a, 1, ab) \rightarrow (ab, 1, b) \rightarrow (abb, 1, \varepsilon)$
 $\rightarrow (ab, 2, b)$
 $\rightarrow (a, 3, b) \rightarrow (\varepsilon, 3, ab) \rightarrow (\varepsilon, 3, \#ab)$

Calcul : exemple



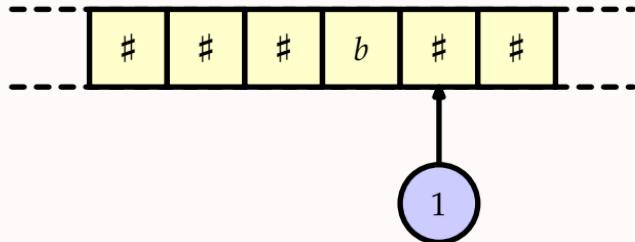
$(\varepsilon, 0, aabb) \rightarrow (\varepsilon, 1, aab) \rightarrow (a, 1, ab) \rightarrow (ab, 1, b) \rightarrow (abb, 1, \varepsilon)$
 $\rightarrow (ab, 2, b)$
 $\rightarrow (a, 3, b) \rightarrow (\varepsilon, 3, ab) \rightarrow (\varepsilon, 3, \#ab)$
 $\rightarrow (\varepsilon, 0, ab)$

Calcul : exemple



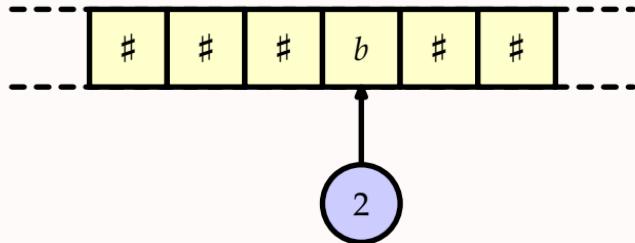
$(\varepsilon, 0, aabb) \rightarrow (\varepsilon, 1, aab) \rightarrow (a, 1, ab) \rightarrow (ab, 1, b) \rightarrow (abb, 1, \varepsilon)$
 $\rightarrow (ab, 2, b)$
 $\rightarrow (a, 3, b) \rightarrow (\varepsilon, 3, ab) \rightarrow (\varepsilon, 3, \#ab)$
 $\rightarrow (\varepsilon, 0, ab)$
 $\rightarrow (\varepsilon, 1, b)$

Calcul : exemple



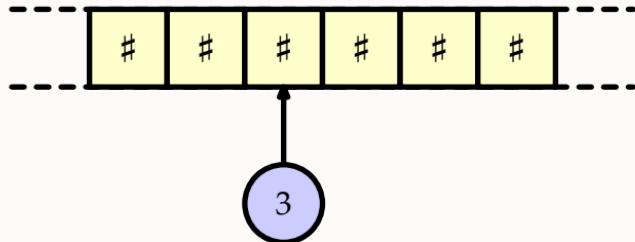
$(\varepsilon, 0, aabb) \rightarrow (\varepsilon, 1, aab) \rightarrow (a, 1, ab) \rightarrow (ab, 1, b) \rightarrow (abb, 1, \varepsilon)$
 $\rightarrow (ab, 2, b)$
 $\rightarrow (a, 3, b) \rightarrow (\varepsilon, 3, ab) \rightarrow (\varepsilon, 3, \#ab)$
 $\rightarrow (\varepsilon, 0, ab)$
 $\rightarrow (\varepsilon, 1, b) \rightarrow (b, 1, \varepsilon)$

Calcul : exemple



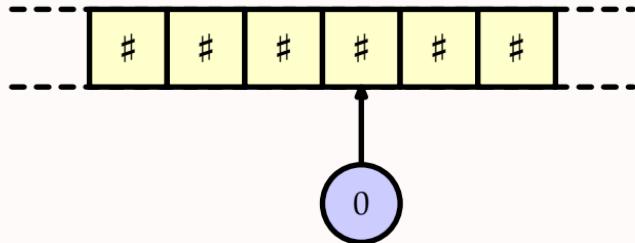
$(\varepsilon, 0, aabb) \rightarrow (\varepsilon, 1, aab) \rightarrow (a, 1, ab) \rightarrow (ab, 1, b) \rightarrow (abb, 1, \varepsilon)$
 $\rightarrow (ab, 2, b)$
 $\rightarrow (a, 3, b) \rightarrow (\varepsilon, 3, ab) \rightarrow (\varepsilon, 3, \#ab)$
 $\rightarrow (\varepsilon, 0, ab)$
 $\rightarrow (\varepsilon, 1, b) \rightarrow (b, 1, \varepsilon)$
 $\rightarrow (\varepsilon, 2, b)$

Calcul : exemple



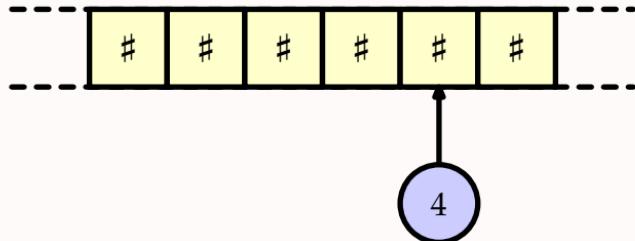
$(\varepsilon, 0, aabb) \rightarrow (\varepsilon, 1, aab) \rightarrow (a, 1, ab) \rightarrow (ab, 1, b) \rightarrow (abb, 1, \varepsilon)$
 $\rightarrow (ab, 2, b)$
 $\rightarrow (a, 3, b) \rightarrow (\varepsilon, 3, ab) \rightarrow (\varepsilon, 3, \#ab)$
 $\rightarrow (\varepsilon, 0, ab)$
 $\rightarrow (\varepsilon, 1, b) \rightarrow (b, 1, \varepsilon)$
 $\rightarrow (\varepsilon, 2, b)$
 $\rightarrow (\varepsilon, 3, \varepsilon)$

Calcul : exemple



$(\varepsilon, 0, aabb) \rightarrow (\varepsilon, 1, aab) \rightarrow (a, 1, ab) \rightarrow (ab, 1, b) \rightarrow (abb, 1, \varepsilon)$
 $\rightarrow (ab, 2, b)$
 $\rightarrow (a, 3, b) \rightarrow (\varepsilon, 3, ab) \rightarrow (\varepsilon, 3, \#ab)$
 $\rightarrow (\varepsilon, 0, ab)$
 $\rightarrow (\varepsilon, 1, b) \rightarrow (b, 1, \varepsilon)$
 $\rightarrow (\varepsilon, 2, b)$
 $\rightarrow (\varepsilon, 3, \varepsilon) \rightarrow (\varepsilon, 0, \varepsilon)$

Calcul : exemple



$(\varepsilon, 0, aabb) \rightarrow (\varepsilon, 1, aab) \rightarrow (a, 1, ab) \rightarrow (ab, 1, b) \rightarrow (abb, 1, \varepsilon)$
 $\rightarrow (ab, 2, b)$
 $\rightarrow (a, 3, b) \rightarrow (\varepsilon, 3, ab) \rightarrow (\varepsilon, 3, \#ab)$
 $\rightarrow (\varepsilon, 0, ab)$
 $\rightarrow (\varepsilon, 1, b) \rightarrow (b, 1, \varepsilon)$
 $\rightarrow (\varepsilon, 2, b)$
 $\rightarrow (\varepsilon, 3, \varepsilon) \rightarrow (\varepsilon, 0, \varepsilon) \rightarrow (\varepsilon, 4, \varepsilon)$

Langage accepté par une machine de Turing

- Un mot $u \in \Sigma^*$ est **accepté** par une machine de Turing si $(\varepsilon, q_0, u) \xrightarrow{*} (s, q_F, t)$ avec $q_F \in F$, c'est-à-dire :
 - ▶ Le calcul de la machine **termine** sur l'entrée u .
 - ▶ L'état de la configuration finale est un état final.
- Ainsi, un mot est rejeté lorsque :
 - ▶ Le calcul termine et aboutit dans un état non final.
 - ▶ Le calcul est infini.
- Le **langage accepté** par une machine de Turing M est l'ensemble des mots sur $u \in \Sigma^*$ tels que M accepte u .
- La classe des langages acceptés par machine de Turing est appelée classe des langages **récursivement énumérables**

Langage décidé par une machine de Turing

- On dit qu'une machine de Turing M **décide** un langage L sur Σ s'il existe une machine de Turing M qui
 - ▶ **termine** sur toute entrée $u \in \Sigma^*$
 - ▶ **accepte** L
- **Prop :** un langage décidable est récursivement énumérable.
- La classe des langages décidés par machine de Turing est appelée classe des langages **décidables** ou encore **récursifs**.

Lien avec le cours MPI

- Les problèmes de décision **décidables** sont ceux dont le langage des instances positives est décidable par machine de Turing.
- Le langage des instances positives du problème de correspondance de Post n'est pas décidable, par contre il est récursivement énumérable, on dit alors que le problème est **semi-décidable**.
- Les problèmes de décision de la **classe P** sont ceux dont le langage des instances positives est **décidable** par une machine de Turing **déterministe** et qui vérifie de plus que le nombre d'étapes de calcul de la machine est un $O(n^k)$ pour un certain $k \geq 0$ avec n la taille de l'instance.
- Les problèmes de décision de la **classe NP** sont ceux dont le langage des instances positives est **décidable** par une machine de Turing **non-déterministe** et dont le nombre d'étapes d'un calcul acceptant est un $O(n^k)$. Nous n'avons pas défini cette machine, mais on imagine aisément à quoi elle peut ressembler par analogie avec les automates finis.

Exercices

1. **Exemple historique de Turing** : donner une machine de Turing prenant en entrée une suite de 1 et qui construit une suite de 1 deux fois plus longue en intercalant un symbole blanc $\#$. Le format est le suivant : si l'entrée est 111 la sortie sera 111 $\#$ 111.
2. Donner une machine de Turing qui calcule $n + 1$ lorsque $n \in \mathbb{N}$ est placée sur son ruban d'entrée sous format binaire $\Sigma = \{0, 1\}$ avec le bit de poids faible à droite.
3. Démontrer que le langage des mots sur $\Sigma = \{a, b\}$ qui contiennent un nombre pair de a est décidable en temps polynomial.
4. Démontrer que tout langage régulier est décidable en temps polynomial.
5. Démontrer que le langage des mots sur $\Sigma = \{a, b\}$ qui contiennent autant de a que de b est décidable en temps polynomial. *Indication* : on pourra utiliser l'alphabet $\Gamma = \{a, b, A, B, \#\}$ pour marquer les lettres.
6. Démontrer que le langage des mots sur $\Sigma = \{a\}$ de longueur une puissance de 2 est décidable.