

Langages et Automates

Introduction

Engel Lefaucieux

Prépas des INP

Organisation du cours

- 8 CM de 1h30
- 5 TD en demi-groupe
- 1 DM (30% de la note finale)
- 1 Examen (70% de la note finale)
- L'ensemble des slides se trouve sur ma page web
<https://elefauch.github.io/>

Objectifs du cours

- Méthodologie et approche scientifique :
 - La modélisation mathématique de problèmes informatiques.
 - L'analyse des modèles mathématiques.
- Connaissances spécifiques :
 - Plusieurs formalismes de modélisation (expression régulière, automates, ...)
 - Les capacités de ces modèles, ainsi que leurs limitations.

Plan

- 1 Qu'est-ce qu'un langage ?
- 2 Langages et machine
- 3 Qu'est-ce qu'un automate ?
- 4 L'ordinateur, automate à états déterministe
- 5 Quelques notions de décidabilité (algorithmique)

Outline

- 1 Qu'est-ce qu'un langage ?
- 2 Langages et machine
- 3 Qu'est-ce qu'un automate ?
- 4 L'ordinateur, automate à états déterministe
- 5 Quelques notions de décidabilité (algorithmique)

Qu'est-ce qu'un langage ?

- Un nombre incroyable de langage
 - Français, anglais, chinois, Russe,...
 - Braille, SMS, Morse,...
 - Pascal, Ocaml, C++, Python,...
- Origine du langage
 - Parlé \approx -50000 ?
 - Écrit \approx -6000 (écriture cunéiforme)
 - Informatique : 1951, A0
- Un langage est *structuré*:
 - Symboles (lettre, hiéroglyphe, chiffre,...)
 - Mots
 - Ordonnancement (phrase, structure "if, then, else",...)

L'importance des règles

- Sans règle, tout est un langage
- L'ensemble des nombres premiers.
- Les programmes Python qui compilent correctement.
- L'ensemble des théorèmes mathématiquement vrai.

Résoudre un problème, c'est identifier un langage.

Entrée : un système \mathcal{A}

Question : \mathcal{A} satisfait-il la propriété P ?

Entrée : un mot \mathcal{A}

Question : est-ce que \mathcal{A} appartient au langage L_P ?

→ Certains langages ne sont pas étudiables !

Qu'est-ce qu'un langage ?

Langages et machine

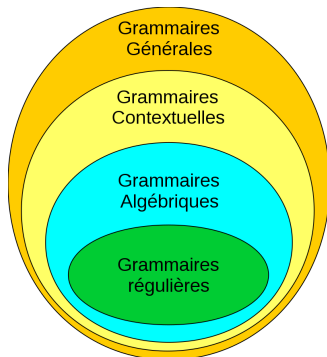
Qu'est-ce qu'un automate ?

L'ordinateur, automate à états déterministe

Quelques notions de décidabilité (algorithmique)

Hierarchie des langages (Chomsky, 1956)

Chomsky propose des limitations pour différentes grammaires, définissant ainsi des classes de langages formels



Les grammaires Générales ne sont pas toute puissantes.

→ Machine de Turing

Exemple d'un langage

Comment modéliser les exécutions d'un système simple ?

- Attribuer un symbole ou mot à chaque action du système
- N'autoriser que les ordres correspondant au système.

Exemple d'un langage

Comment modéliser les exécutions d'un système simple ?

- Attribuer un symbole ou mot à chaque action du système
- N'autoriser que les ordres correspondant au système.

Comment représenter l'achat d'une baguette ou d'un croissant dans une boulangerie ?

- *entrer, commander_pain, commander_croissant, payer, sortir*
- *entrer commander_pain payer sortir*
ou
entrer commander_croissant payer sortir

Exemple d'un langage (2)

Nous voulons modéliser une machine à café possédant les propriétés suivantes :

- Si à l'arrêt, on peut cliquer sur un bouton pour l'activer.
- En cours de fonctionnement, la machine fait du bruit pendant une durée aléatoire.
- Si la machine a été activé, elle va éventuellement produire du café et s'éteindre.

Exemple d'un langage (2)

Nous voulons modéliser une machine à café possédant les propriétés suivantes :

- Si à l'arrêt, on peut cliquer sur un bouton pour l'activer.
- En cours de fonctionnement, la machine fait du bruit pendant une durée aléatoire.
- Si la machine a été activé, elle va éventuellement produire du café et s'éteindre.

Quels symboles / mots pour ce modèle ?

Exemple d'un langage (2)

Nous voulons modéliser une machine à café possédant les propriétés suivantes :

- Si à l'arrêt, on peut cliquer sur un bouton pour l'activer.
- En cours de fonctionnement, la machine fait du bruit pendant une durée aléatoire.
- Si la machine a été activé, elle va éventuellement produire du café et s'éteindre.

Quels symboles / mots pour ce modèle ?

Quelles phrases représentent un bon fonctionnement du modèle ?

Qu'est-ce qu'un langage ?

Langages et machine

Qu'est-ce qu'un automate ?

L'ordinateur, automate à états déterministe

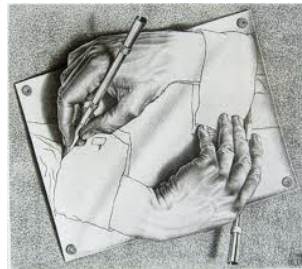
Quelques notions de décidabilité (algorithmique)

Outline

- 1 Qu'est-ce qu'un langage ?
- 2 **Langages et machine**
- 3 Qu'est-ce qu'un automate ?
- 4 L'ordinateur, automate à états déterministe
- 5 Quelques notions de décidabilité (algorithmique)

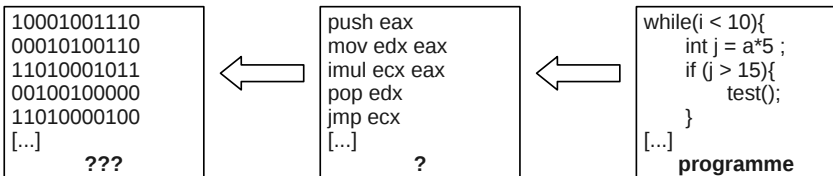
Communication homme-machine

- Un langage permet de communiquer entre :
 - Humain et humain
 - Humain et animal (?)
 - Animal et animal (???)
 - **Humain et machine**
 - ...
- Cette communication s'appuie sur :
 - Le dialogue homme-machine
 - Les interfaces homme-machine
 - \Rightarrow la programmation, par des informaticiens, de logiciels afin que les machines puissent comprendre d'autres hommes



De langages en langages

- Initialement, la programmation d'automates (ordinateurs) se faisait par utilisation de **codes**
- La correspondance entre **instructions** (plus ou moins basiques) et **langage machine** se fait par un processus (**automatique**) de **compilation**



Identification automatique d'un langage

- L'**automate** (ordinateur) « comprend » un langage que l'ingénieur lui soumet :
 - Comme une **commande** (« ordre »)
 - Il « **reconnaît** » ou « **accepte** » un langage
 - Il ne « reconnaît pas » ou « n'accepte pas » les autres langages
- Ceci a besoin d'être formalisé pour :
 - Créer de **nouveaux langages**, sous certaines contraintes :
 - **Efficacité** : rapidité des traitements
 - **Expressivité** : capacité à formuler des « choses »
 - **Précision** : degré de contrôle sur ce que fait la machine
 - Savoir à l'avance **si un message** / code **sera reconnu** ou non par l'automate (sinon : erreurs)

Outline

- 1 Qu'est-ce qu'un langage ?
- 2 Langages et machine
- 3 Qu'est-ce qu'un automate ?
- 4 L'ordinateur, automate à états déterministe
- 5 Quelques notions de décidabilité (algorithmique)

Qu'est-ce qu'un automate ?

- Du grec "*αυτοματος*" (automatos) : qui agit de sa propre volonté, de son propre esprit.
→ Devenu "qui se meut soi-même"
- C'est un dispositif reproduisant en autonomie une séquence d'actions prédéterminées sans l'intervention humaine.
- Pas nécessairement un robot.



Automate en informatique

Défini par des états et des transitions, une entrée et une sortie

- **États** : décrit la situation interne actuelle du système
- **Transition** : évolutions possibles du système depuis un état.



Les transitions peuvent être enrichies par des probabilités, des actions, des poids, des temps, . . .

- **Entrée** : ce que l'automate "consomme"
- **Sortie** : un verdict / une décision

Par exemple, l'acceptation du mot fourni en entrée.

Qu'est-ce qu'un langage ?

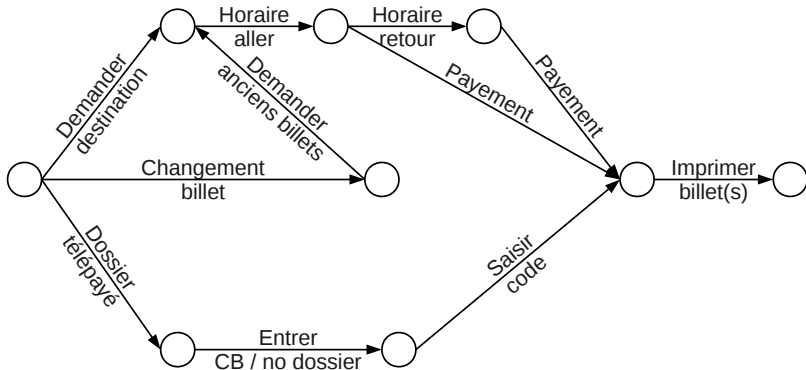
Langages et machine

Qu'est-ce qu'un automate ?

L'ordinateur, automate à états déterministe

Quelques notions de décidabilité (algorithmique)

Un distributeur automatique de billet



Accepte toute interaction "normale" avec le distributeur

Qu'est-ce qu'un langage ?

Langages et machine

Qu'est-ce qu'un automate ?

L'ordinateur, automate à états déterministe

Quelques notions de décidabilité (algorithmique)

Un automate pour la machine à café

Comment représenteriez-vous une machine à café ?

Qu'est-ce qu'un langage ?

Langages et machine

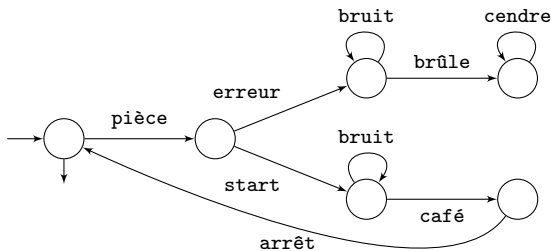
Qu'est-ce qu'un automate ?

L'ordinateur, automate à états déterministe

Quelques notions de décidabilité (algorithmique)

Un automate pour la machine à café

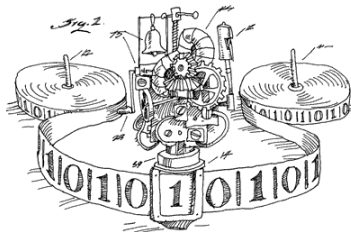
Comment représenteriez-vous une machine à café ?



La machine de Turing, un automate avancé

Inventé par Alan Turing en 1936

- entrée = un ruban de symboles infini
- une tête de lecture indique quel symbole du ruban est actuellement lu
- une transition peut écrire sur le ruban et se déplacer



Outline

- 1 Qu'est-ce qu'un langage ?
- 2 Langages et machine
- 3 Qu'est-ce qu'un automate ?
- 4 L'ordinateur, automate à états déterministe
- 5 Quelques notions de décidabilité (algorithmique)

Qu'est-ce qu'un langage ?

Langages et machine

Qu'est-ce qu'un automate ?

L'ordinateur, automate à états déterministe

Quelques notions de décidabilité (algorithmique)

Quel modèle pour l'ordinateur ?

Pour chaque système, il faut choisir un modèle adapté.

Est-ce qu'une machine de Turing représente correctement un ordinateur ?

Qu'est-ce qu'un langage ?

Langages et machine

Qu'est-ce qu'un automate ?

L'ordinateur, automate à états déterministe

Quelques notions de décidabilité (algorithmique)

Quel modèle pour l'ordinateur ?

Pour chaque système, il faut choisir un modèle adapté.

Est-ce qu'une machine de Turing représente correctement un ordinateur ?

Ruban / mémoire infinie \rightarrow plus puissant qu'un ordinateur

Quel modèle pour l'ordinateur ?

Pour chaque système, il faut choisir un modèle adapté.

Est-ce qu'une machine de Turing représente correctement un ordinateur ?

Ruban / mémoire infinie \rightarrow plus puissant qu'un ordinateur

De quelles limitations et quelles capacités avons-nous besoin ?

Un système communiquant

Un ordinateur, des systèmes

- L'ordinateur reçoit un **langage** : les périphériques d'entrée traduisent sons, images, et clics en signaux binaires pour le processeur
- L'ordinateur émet un **langage** : les signaux binaires produits par le processeur sont traduits en son, image et messages par les périphériques de sortie
- L'ordinateur est idiot. Il ne fait que des actions **programmées**.

Un système fini

De nombreuses données peuvent décrire l'**état** actuel du système

- État du processeur
- Variables (registres) du processeur
- Mémoire volatile (RAM)
- Mémoire centrale (disque dur)
- État des périphériques
- État du réseau
- ...

Une représentation complète est onéreuse.

Souvent, on voudra une représentation partielle et **finie**.

Un système déterministe

- L'ordinateur "sait" à tout moment dans quel état il se trouve. Il n'y a pas de "peut-être" (on oublie l'informatique quantique pour le moment).
- L'univers est aléatoire et l'aléatoire sert en informatique, mais on ne peut pas réellement produire de comportement proprement aléatoire dans un ordinateur.

Automate dans ce cours : **fini, déterministe, à information parfaite** et traitant un **langage**.

Qu'est-ce qu'un langage ?

Langages et machine

Qu'est-ce qu'un automate ?

L'ordinateur, automate à états déterministe

Quelques notions de décidabilité (algorithmique)

Outline

- 1 Qu'est-ce qu'un langage ?
- 2 Langages et machine
- 3 Qu'est-ce qu'un automate ?
- 4 L'ordinateur, automate à états déterministe
- 5 Quelques notions de décidabilité (algorithmique)

Algorithme

Un algorithme consiste en un ensemble fini d'instructions simples et précises.

- il doit produire un résultat en un nombre fini d'étapes.
- un humain équipé d'un papier et d'un crayon devrait pouvoir le réaliser.
- aucune intelligence hormis la compréhension et l'exécution des instructions n'est nécessaire.

Algorithme

Un algorithme consiste en un ensemble fini d'instructions simples et précises.

- il doit produire un résultat en un nombre fini d'étapes.
- un humain équipé d'un papier et d'un crayon devrait pouvoir le réaliser.
- aucune intelligence hormis la compréhension et l'exécution des instructions n'est nécessaire.

Qu'est-ce qu'on considère simple ?

Algorithme

Un algorithme consiste en un ensemble fini d'instructions simples et précises.

- il doit produire un résultat en un nombre fini d'étapes.
- un humain équipé d'un papier et d'un crayon devrait pouvoir le réaliser.
- aucune intelligence hormis la compréhension et l'exécution des instructions n'est nécessaire.

Qu'est-ce qu'on considère simple ?

→ Réalisable par une machine de Turing
(Thèse de Church-Turing)

Décidabilité

Un problème est décidable s'il existe un algorithme qui répond par oui ou par non à la question posée par le problème.

Entrée : un entier n

Est-ce que n est un nombre premier ?

Décidabilité

Un problème est décidable s'il existe un algorithme qui répond par oui ou par non à la question posée par le problème.

Entrée : un entier n

Est-ce que n est un nombre premier ?

Décidable ✓

Décidabilité

Un problème est décidable s'il existe un algorithme qui répond par oui ou par non à la question posée par le problème.

Entrée : un entier n

Est-ce que n est un nombre premier ?

Décidable ✓

Entrée : un texte L

Est-ce que L est un programme Python qui compile correctement ?

Décidabilité

Un problème est décidable s'il existe un algorithme qui répond par oui ou par non à la question posée par le problème.

Entrée : un entier n

Est-ce que n est un nombre premier ?

Décidable ✓

Entrée : un texte L

Est-ce que L est un programme Python qui compile correctement ?

Décidable ✓

Qu'est-ce qu'un langage ?

Langages et machine

Qu'est-ce qu'un automate ?

L'ordinateur, automate à états déterministe

Quelques notions de décidabilité (algorithmique)

Problème de l'arrêt

Entrée : un programme Python / une Machine de Turing / ...

Est-ce que l'exécution du programme termine ?

Problème de l'arrêt

Entrée : un programme Python / une Machine de Turing / ...

Est-ce que l'exécution du programme termine ?

Procédure Syracuse ($i \in \mathbb{N}$)

```
while  $i \neq 1$  do  
  if  $i$  est pair then  
     $i := i/2$   
  else  
     $i := 3i + 1$   
  end if  
end while
```

Qu'est-ce qu'un langage ?

Langages et machine

Qu'est-ce qu'un automate ?

L'ordinateur, automate à états déterministe

Quelques notions de décidabilité (algorithmique)

Problème de l'arrêt (2)

Theorem

Le problème de l'arrêt est indécidable.

Problème de l'arrêt (2)

Theorem

Le problème de l'arrêt est indécidable.

Supposons l'existence de

Fonction halt (P : programme) : booléen

if P s'arrête **then**

renvoie *true*

else

renvoie *false*

end if

Problème de l'arrêt (3)

Procédure contradiction (P : programme)

$test := \text{halt}(P)$

if test **then**

while true **do**

end while

else

 print "STOP"

end if

Problème de l'arrêt (3)

Procédure contradiction (P : programme)

test := halt(P)

if test **then**

while true **do**

end while

else

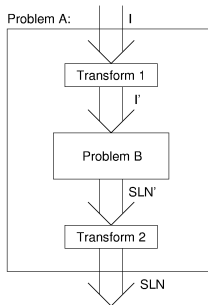
 print "STOP"

end if

Que se passe t'il si on exécute contradiction (contradiction) ?

Comment prouve t-on l'indécidabilité d'un problème ?

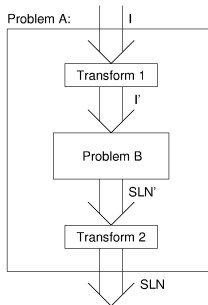
La réduction : Si on peut résoudre B, on peut résoudre A.



D'un algorithme pour B, la réduction crée un algorithme pour A.

Comment prouve t-on l'indécidabilité d'un problème ?

La réduction : Si on peut résoudre B, on peut résoudre A.



D'un algorithme pour B, la réduction crée un algorithme pour A.

Si A est indécidable, alors B est indécidable.

Qu'est-ce qu'un langage ?

Langages et machine

Qu'est-ce qu'un automate ?

L'ordinateur, automate à états déterministe

Quelques notions de décidabilité (algorithmique)

Quelques exemples

Entrée : une Machine de Turing M et un mot w

Est-ce que M accepte le mot w ?

Quelques exemples

Entrée : une Machine de Turing M et un mot w

Est-ce que M accepte le mot w ?

Théorème de Rice : Toute propriété sémantique non-triviale d'un programme (*i.e.* sur son comportement et non son code) est indécidable.

Quelques exemples

Entrée : une Machine de Turing M et un mot w

Est-ce que M accepte le mot w ?

Théorème de Rice : Toute propriété sémantique non-triviale d'un programme (*i.e.* sur son comportement et non son code) est indécidable.

Entrée : x , un des élèves de cette classe

Est-ce que l'élève x réussira ses concours cette année ?

Quelques exemples (2)

Une équation diophantienne est une équation polynomiale à plusieurs inconnues dont les solutions sont cherchées parmi les nombres entiers.

$x = 3$, $y = 4$ et $z = 5$ sont une solution de l'équation diophantienne $x^2 + y^2 = z^2$.

Entrée : une équation diophantienne P sur des variables x_1, \dots, x_k

Est-ce qu'il existe des entiers n_1, \dots, n_k tels que l'équation P où x_i est remplacé par n_i est vraie ?

Quelques références

Quelques liens vers des documents ayant aidé à réaliser ce cours :

- <https://perso.liris.cnrs.fr/christine.solnon/langages.pdf>
- http://www.discmath.ulg.ac.be/cours/main_autom.pdf
- <https://pageperso.lis-lab.fr/frederic.olive/Materiel/langagesL2/cours.pdf>
- <https://damien.nouvel.net/fr/enseignement>
- https://www.i3s.unice.fr/nlt/cours/licence/it/s6_itdut_poly.pdf