1. Introduction

Bruno Grenet

Université Grenoble Alpes – IM^{2}AG

L3 Informatique

UE Modèles de calcul – Machines de Turing

https://membres-ljk.imag.fr/Bruno.Grenet/MCAL-MT.html

Table des matières

Table des matières

Présentation

Nom : Bruno Grenet
Email : `...@univ-grenoble-alpes.fr` *Insérer `[MCAL-MT]` en sujet*
Page web : https://membres-ljk.imag.fr/Bruno.Grenet/

Affiliation : Université Grenoble Alpes
UFR IM²AG
Laboratoire Jean Kuntzmann (LJK)
Téléphone : +33 457 421 721
Bureau : Bâtiment IMAG (150 pl. du Torrent) ; 1er étage ; bureau 106

Organisation

Équipe pédagogique

Cours : Bruno Grenet
TD G1 : Ernest Foussard
TD G2 : Thomas Rahab-Lacroix
TD G3 : Ernest Foussard

Contacts : `<Prenom>.<Nom>@univ-grenoble-alpes.fr`

⚠ Planning de cette année ⚠

Cours : 1 à 2 cours par semaine en début de semestre
Pause pendant env. un mois (*congé paternité → date non fixée…*)
1 cours par semaine en fin de semestre

TD : 1 séance par semaine, sans interruption (*sauf peut-être une semaine*)

→ **Communication par email et Mattermost + consultation d'ADE !**

Évaluations

EC: Évaluation continue (*quick*)

Durée : 1h
Date : vendredi 22 mars 2024
Coefficient : 0,4
Contenu : partie « MCAL-MT »

ET: Évaluation terminale

Durée : 2h
Date : semaine des examens (12-17 avril 2024)
Coefficient : 1,2
Contenu : commun « MCAL-MT » et « MCAL-LC »

ES: Évaluation supplémentaire

- Similaire à l'évaluation terminale
- Semaine des 2$^{\text{ndes}}$ sessions (20-28 juin 2024)

Note finale

Session 1 : $\max(ET; \frac{1}{5}(EC_{MT} + EC_{LC} + 3 \cdot ET))$
Session 2 : $\max(ES; \frac{1}{5}(EC_{MT} + EC_{LC} + 3 \cdot ES))$

Page du cours

https://membres-ljk.imag.fr/Bruno.Grenet/MCAL-MT.html

Contenu

- Diapositives de cours, avec annotations
- Sujets de TD
- Bibliographie :
  - livres, notes de cours
  - liens : vidéos, articles de vulgarisation, sites web, …

→ **Très utile à consulter !**

Table des matières

De quoi va-t-on parler ?

Résumé (personnel) de la vidéo

Origines

- *Mécanisation* du raisonnement ? Leibniz (XVII^e siècle)
- *Entscheidungsproblem* : existe-t-il un algorithme pour décider si un énoncé mathématique est correct ? Hilbert & Ackermann (1928)
- Années 1930 : intuition de l'absence d'algorithme $\rightarrow$ comment la prouver ?

Formalisation des algorithmes

- Fonctions μ-récursives Herbrand-Gödel (1931-34)
- λ-calcul Kleene-Church (1932-36)
- Machines de Turing Turing (1936)

Trois définitions prouvées équivalentes $\rightarrow$ c'est la « bonne » définition !

Incalculabilité

- Pas d'algorithme pour le problème de la décision, le problème de l'arrêt, …

$\rightarrow$ Résultats négatifs mais une des origines de l'informatique

Objectifs du cours

Machines de Turing

- Étude d'un des modèles de calcul
- (Autre modèle – λ-calcul – étudié dans l'autre partie du cours)
- Comparaison avec d'autres modèles

Calculabilité

- Qu'est-ce que veut dire *calculer* ?
- Comment démontrer l'*inexistence* d'algorithme pour un problème donné ?
- Liens avec l'informatique *de tous les jours* ?

Avant / après

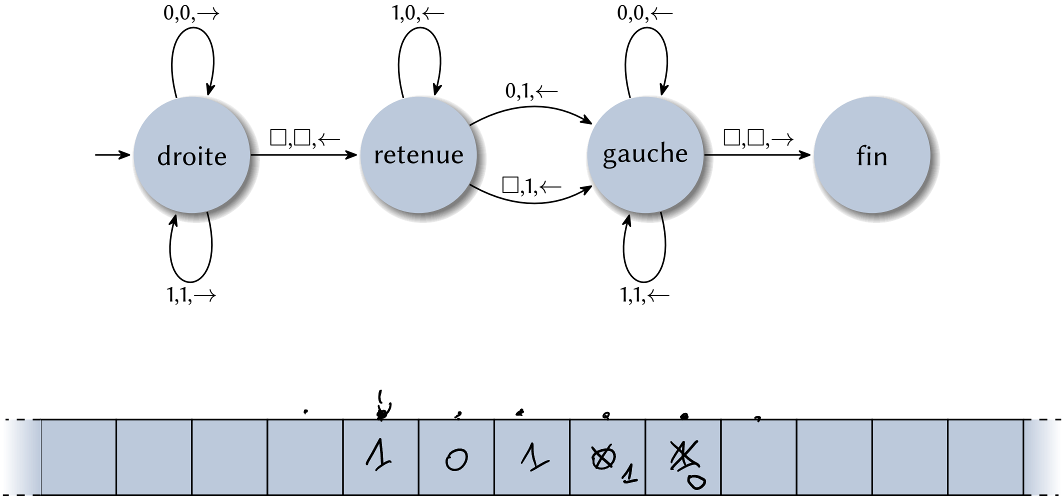
- Automates et Langages (L2) ; Analyse syntaxique (L3)
  - Automates finis ; automates à pile
- Complexité Algorithmique ; Fundamentals of Computer Science (M1 Mosig / INFO)
  - Théorie de la complexité

Plan (approximatif)

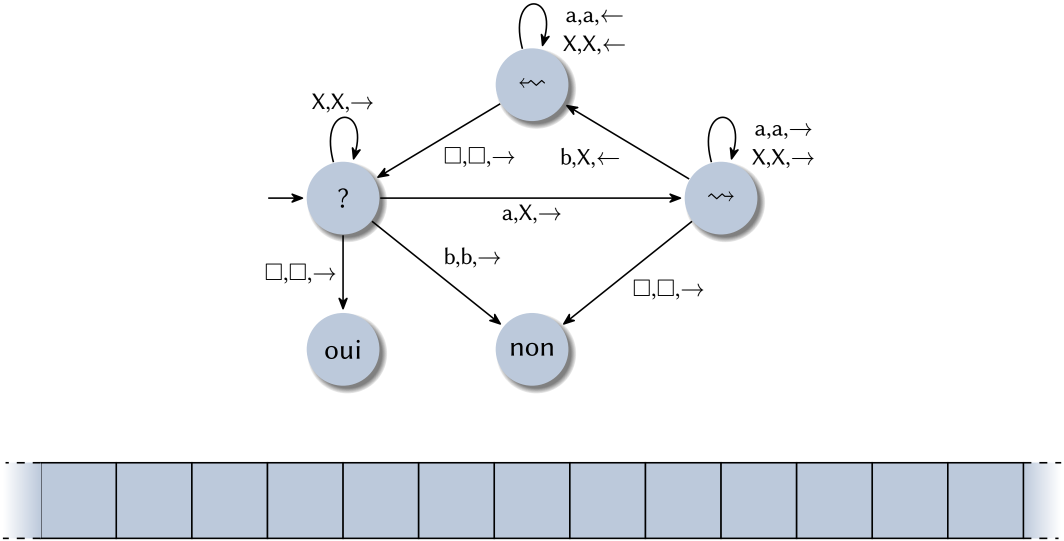
1. Machines de Turing et variantes
   - Exemples de machines
   - Définition (informelle et formelle)
   - Robustesse : les variantes ne changent pas grand chose
2. Thèse de Church-Turing
   - Algorithme = Machine de Turing (ou λ-calcul, ou ...)
   - Équivalence entre modèles
3. Incalculabilité
   - Il existe des fonctions non calculables (*diagonalisation*)
   - Exemples concrets
4. Réductions entre problèmes
5. Autres résultats de calculabilité — si le temps le permet

Table des matières

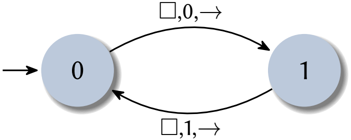
Premier exemple

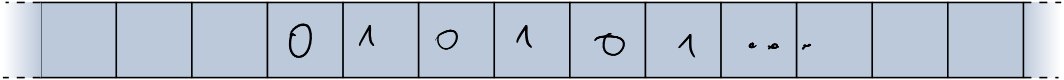


Deuxième exemple



Troisième exemple





Description des machines de Turing

Constituants d'une machine de Turing

- Un ensemble d'*états* Q, dont un *état initial* q_0 *sommets*
- Une *table de transition* δ *automate*
- Un *ruban* constitué d'une infinité de *cases*
- Une *tête* de lecture/écriture

Le ruban

- Chaque case du ruban contient un *symbole* d'un *alphabet* Σ
- L'alphabet contient un symbole spécial « blanc » $\square$ *case vide*
- Le ruban ne contient qu'un nombre fini de symboles $\neq \square$

La table de transition

- Pour *certains* couples $(q, x) \in Q \times \Sigma$, $\delta(q, x) = (q', y, \leftrightarrow)$ où
  - $q, q' \in Q$ sont l'*ancien* et le *nouvel état*
  - x et $y \in \Sigma$ sont les symboles *lu* et *écrit*
  - $\leftrightarrow \in \{\leftarrow, \rightarrow\}$ est le *déplacement*
- Sinon, $\delta(q, x)$ est *indéfini*

Fonctionnement d'une machine de Turing

Une étape de calcul

- *Configuration* : machine dans l'état q et lettre x lue sur le ruban
- Si $\delta(q, x)$ n'est pas défini dans la table : arrêt (définitif) de la machine
- Si $\delta(q, x) = (q', y, \leftrightarrow)$, nouvelle configuration :
  - la machine passe dans l'état q'
  - la tête écrit y à la place de x
  - la tête est déplacée à gauche (si $\leftrightarrow = \leftarrow$) ou droite (si $\leftrightarrow = \rightarrow$)

Calcul complet

- On commence dans la *configuration initiale* :
  - Machine dans l'état initial q_0
  - Entrée w du calcul inscrite sur le ruban
  - Tête sur la 1ère lettre de w
- On applique des étapes de calcul tant qu'on peut
  - Soit la machine finit par se bloquer : calcul terminé $\qquad \delta(q, x)$ indéfini
  - Sinon : la machine *boucle*
- Remarque : pas de *sémantique* $\rightarrow$ reconnaissance de langage, calcul de fonction, …

Utilisation d'une machine de Turing

Reconnaissance de langage

- Hypothèse : $\mathcal{M}$ contient un état spécial « oui »
- Langage $L(\mathcal{M})$ reconnu par $\mathcal{M}$: sur l'entrée w,
  - si le calcul termine dans l'état « oui » : $w \in L(\mathcal{M})$
  - sinon (calcul non terminé ou autre état final) : $w \notin L(\mathcal{M})$ pas besoin de « non »

Calcul de fonction

- Fonction $f_{\mathcal{M}}$ calculée par $\mathcal{M}$: sur l'entrée w
  - si le calcul termine, $f_{\mathcal{M}}(w)$ est le mot inscrit sur le ruban à la fin du calcul
  - sinon $f_{\mathcal{M}}(w)$ est *indéfini*, noté $f_{\mathcal{M}}(w)\uparrow$: $\mathcal{M}$ *boucle sur* w

Remarques

- Autres conventions possibles
  - États d'acceptation et de rejet
  - Prise en compte de l'état final et du contenu final du ruban en même temps
- Définition possible d'*énumération* (infinie)

Formalisation : machine de Turing

Définition

Une machine de Turing est un quadruplet $\mathcal{M} = (Q, \Sigma, q_0, \delta)$ où

- Q est un ensemble fini d'*états*
- Σ est l'*alphabet* contenant le blanc $\square$
- $q_0 \in Q$ est l'*état initial*
- $\delta : Q \times \Sigma \to Q \times \Sigma \times \{\to, \leftarrow\}$ est la *table de transition*

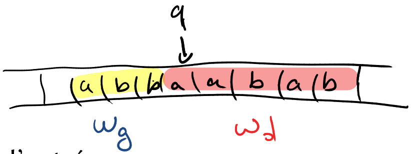
Remarques

- Ruban implicite et *non borné* plutôt qu'infini
- Table de transition : fonction *partielle* $\to \delta(q, x)$ peut être *indéfini*
- Il existe d'autres formalisations équivalentes

Formalisation : configurations

Configuration : état global de la machine à un instant donné
$\rightarrow$ état $q \in Q$ + mot w sur le ruban + position de la tête de lecture sur w

Représentation

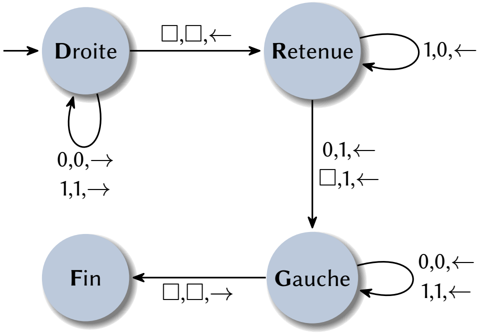


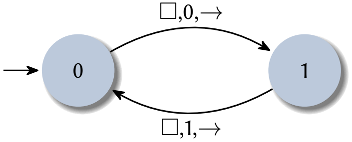
- Triplet $C = (q, w_g, w_d)$ où
  - $w = w_g w_d$ (concaténation)
  - la tête de lecture est sur la 1ère lettre de w_d
- Configuration initiale $C_0 = (q_0, \square, w)$ où w est l'entrée

Étapes de calcul : notations

- $C \vdash C'$ si on passe de C à C' en appliquant la table de transition
- $C \vdash^* C'$ s'il existe $C_1, \ldots, C_k$ telles que $C \vdash C_1 \vdash C_2 \vdash \cdots \vdash C_k \vdash C'$
- $C \downarrow$ si aucune règle de la table de transition ne s'applique à C (calcul terminé)
- Calcul complet sur l'entrée w :
  - Suite infinie $C_0 \vdash C_1 \vdash \cdots$ si la machine boucle
  - Suite finie $C_0 \vdash C_1 \vdash \cdots \vdash C_k \downarrow$ si le calcul termine

Exemples





$$(D, \square, 1011) \vdash (D, 1, 011) \vdash (D, 10, 11)$$
$$\vdash (D, 101, 1) \vdash (D, 1011, \square) \vdash (R, 101, 1)$$
$$\vdash (R, 10, 10) \vdash (R, 1, 000) \vdash (G, \square, 1100)$$
$$\vdash (G, \square, \square 1100) \vdash (F, \square, 1100) \downarrow$$

$$(0, \square, \square) \vdash (1, 0, \square) \vdash (0, 01, \square)$$
$$\vdash (1, 010, \square) \vdash (0, 0101, \square) \vdash \cdots$$

Formalisation : résultat d'un calcul

Calcul d'une machine $\mathcal{M}$ sur l'entrée w

- Calcul de $\mathcal{M}$ sur w : suite finie ou infinie $C_0 \vdash C_1 \vdash C_2 \vdash \cdots$
  - $\mathcal{M}(w) = C_k$ si $C_0 \vdash^* C_k \downarrow$
  - $\mathcal{M}(w) \uparrow$ si la suite est infinie
- Remarque : $\mathcal{M}(w)$ est la *configuration* finale (si elle existe)

Reconnaissance et calcul de fonction

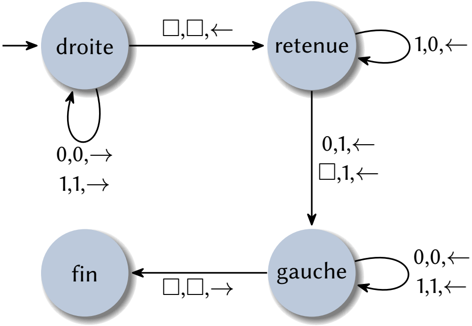
- $L(\mathcal{M}) = \{w : \mathcal{M}(w) = (\text{« oui »}, *, *)\}$ avec $* =$ n'importe quel mot
- $f_{\mathcal{M}}(w) = w'_g w'_d$ si $\mathcal{M}(w) = (q, w'_g, w'_d)$ pour n'importe quel état q

Exemple 1 : $f_M(n) = n+1$ où n est écrit en binaire

Exemple 2 : $L(M) = \{w : w$ a autant de a que de b, tout préfixe a $\geq$ a que b $\}$

Exemple 3 : $f_M(\square)\uparrow$ $f_M(w) = w$ pour $w \neq \square$ $L(M) = \{w : w \neq \square\}$

Représentation de la table de transition



État	Lu	Écrit	Dépl.	Nv. état
droite	0	0	$\rightarrow$	droite
	1	1	$\rightarrow$	droite
	□	□	$\leftarrow$	retenue
retenue	1	0	$\leftarrow$	retenue
	0	1	$\leftarrow$	gauche
	□	1	$\leftarrow$	gauche
gauche	0	0	$\leftarrow$	gauche
	1	1	$\leftarrow$	gauche
	□	□	$\rightarrow$	fin

- δ : liste de quintuplets (état, lu, écrit, dépl., nv. état)
- δ suffit à représenter $\mathcal{M}$:
  - Q : liste des états dans la table
  - Σ : liste des symboles dans la table
  - q_0 : 1$^{\text{er}}$ état de la table (par convention)

Conclusion

Machine de Turing

- Extension des automates avec un ruban non borné et une table de transition
- Définitions formelles :
  - Machine de Turing : $\mathcal{M} = (Q, \Sigma, q_0, \delta)$
  - Table de transition : δ = liste de quintuplets $(q, x, q', y, \leftrightarrow)$
  - Configuration : $C = (q, w_g, w_d)$
  - Calcul : $C_0 \vdash C_1 \vdash C_2 \vdash \cdots$

Utilisation des machines de Turing

- Reconnaissance : $L(\mathcal{M}) = \{w$: sur l'entrée $w, \mathcal{M}$ termine dans l'état « oui »$\}$
- Calcul : $f_{\mathcal{M}}(w)$ = mot sur le ruban en fin de calcul, ou $\uparrow$ si la machine boucle
- Autre possibilité : énumération infinie

À savoir faire

- Exécuter une machine (simple) sur une entrée
- Décrire le calcul effectué / le langage reconnu par une machine (simple)
- Créer une machine pour un problème simple