

Chapitre 7

(Chapitres 4 et 5 de Sipser)

Langages indécidables
(ou... NP-ardu ? t'as rien vu !)

Rappels

- L **décidable** : MT accepte tout $w \in L$ et rejette tout $w \notin L$ (donc en particulier s'arrête toujours).
- L **reconnaissable** : MT accepte tout $w \in L$ et rejette/boucle sur tout $w \notin L$ (donc peut boucler à l'infini sur $w \notin L$).
- Church-Turing :
décidable par "algorithme" = décidable par MT.
- $A \leq B$ et B **décidable** $\Rightarrow A$ **décidable**.

Nous rencontrerons des langages dont les mots encodent :

1. automate fini
2. expression régulière
3. grammaire hors-contexte
4. machine de Turing
5. paire de machines de Turing

Exemples

- I. $\{ \langle M, w \rangle : w \in L(M) \}$
- II. $\{ \langle M \rangle : L(M) = \emptyset \}, \{ \langle M \rangle : L(M) = \Sigma^* \}, \dots$
- III. $\{ \langle M_1, M_2 \rangle : L(M_1) = L(M_2) \}$

Problèmes de langages réguliers

Proposition 7.1. Tout langage régulier $\mathcal{Y}$ est décidable. ■

Preuve. Une parmi de nombreuses façons : voir un AFD A tel que $L(A) = \mathcal{Y}$ comme une MT M :

- mêmes états,
- transitions :
 $\delta_A(q, a) = q' \Rightarrow \delta_M(q, a) = (q', a, R)$
- rejette/accepte sur symbole blanc.

Note : chaque $\mathcal{Y}$ donne lieu à une MT. ▲

Théorème 7.2. (Sipser 4.1)
Le langage A_{AFD} défini comme

$$\{\langle B, w \rangle : B \text{ est un AFD et } w \in L(B)\}$$

est **décidable**. ■

Preuve. Sur entrée $\langle B, w \rangle$, M simule B sur w , acceptant si B accepte w et rejetant sinon. ■

Remarque 7.3. Ici une seule MT suffit et est capable de s'ajuster à tous les couples $\langle$ un AFD, mot w $\rangle$ soumis en entrée. ▲

Théorème 7.4. (Sipser 4.2)
Le langage A_{AFN} défini comme

$$\{\langle B, w \rangle : B \text{ est un AFN et } w \in L(B)\}$$

est **décidable**.

■
Preuve. On a

$$A_{AFN} \leq A_{AFD}$$

(on sait construire un AFD équivalent à un AFN) et on vient de voir que A_{AFD} est décidable. ■

Théorème 7.5. (Sipser 4.3)

Le langage A_{ER} défini comme

$$\{\langle E, w \rangle : E \text{ est une ER et } w \in L(E)\}$$

est **décidable**.

■
Preuve. On a

$$A_{ER} \leq A_{AFD}$$

(on sait construire un AFD équivalent à une ER) et on a vu que A_{AFD} est décidable. ■

Théorème 7.6. (Sipser 4.4)
Le langage VIDE_{AF} défini comme

$$\{\langle B \rangle : B \text{ est un AF et } L(B) = \emptyset\}$$

est **décidable**.

Preuve. Vérifier s'il y a un état acceptant qui est accessible à partir de l'état initial.

Si oui, alors rejeter, sinon accepter. ■

Théorème 7.7. (Sipser 4.5)
Le langage EGAL_{AF} défini comme

$$\{\langle A, B \rangle : L(\text{!AF } A) = L(\text{!AF } B)\}$$

est **décidable**.

Preuve. $\text{EGAL}_{\text{AF}} \leq \text{VIDE}_{\text{AF}}$ car

$$\langle A, B \rangle \in \text{EGAL}_{\text{AF}} \Leftrightarrow \langle C \rangle \in \text{VIDE}_{\text{AF}}$$

où

$$L(C) = \left(L(A) \cap \overline{L(B)} \right) \cup \left(\overline{L(A)} \cap L(B) \right).$$

Et on a vu que VIDE_{AF} est décidable. ■

Problèmes de lang. hors-contexte

Théorème 7.8. (Sipser 4.7)

Le langage AGHC défini comme

$$\{\langle G, w \rangle : G \text{ est une GHC et } w \in L(G)\}$$

est **décidable**.

Preuve. Transformer G en f.n. de Chomsky, puis

1. Pour $w \neq \varepsilon$ essayer toutes les dérivations comprenant $2|w| - 1$ étapes, ou encore

2. Programmation dynamique sur un tableau de taille $|w| \times |w|$. ■

Théorème 7.9. (Sipser 4.8)
Le langage VIDE_{GHC} défini comme

$$\{\langle G \rangle : G \text{ est une GHC et } L(G) = \emptyset\}$$

est **décidable**.

Preuve.

Marquer progressivement toutes les variables de G qui peuvent générer une suite de terminaux.

Accepter à la fin ssi la variable de départ est marquée. ■

Théorème 7.10. (Sipser 4.9)
Tout langage hors-contexte Y est **décidable**.

Preuve. Déjà vue ! Preuve alternative :
Soit G la GHC pour Y . Alors

$$Y \leq A_{GHC}$$

par la réduction

$$\begin{aligned} f : \Sigma^* &\rightarrow \Sigma^* \\ w &\mapsto \langle G, w \rangle. \end{aligned}$$

Et le théorème 7.8 nous a montré que A_{GHC} est décidable. ■

Le langage $EGAL_{GHC}$ défini comme

$$\left\{ \langle G, H \rangle : G, H \text{ sont GHCs, } L(G) = L(H) \right\},$$

est-il **décidable** ?



Peut-on réduire

$$EGAL_{GHC} \text{ à } VIDE_{GHC}$$

comme on a réduit

$$EGAL_{AF} \text{ à } VIDE_{AF}$$

à l'aide de $(L(G) \cap \overline{L(H)}) \cup (\overline{L(G)} \cap L(H))$?

Un mur infranchissable

Cette piste et **toute autre piste** pour solutionner $EGAL_{GHC}$ sont vouées à l'échec
puisque

$EGAL_{GHC}$ est indécidable

i.e. il **n'existe pas** de MT décidant $EGAL_{GHC}$

i.e. toute MT s'arrêtant toujours se trompe sur une infinité d'exemplaires de
 $EGAL_{GHC}$

(preuve à venir).

Incursion en territoire infini

- nombres naturels : $\mathbb{N} = \{0, 1, 2, \dots\}$
 - nombres naturels pairs :
 $\mathbb{N}_{\text{pair}} = \{0, 2, 4, 6, \dots\}$
 - nombres réels : $\mathbb{R}$
 - réels de l'intervalle $(0, 1)$: $\mathbb{R} \cap (0, 1)$
 - réels de l'intervalle $[0, 1]$: $\mathbb{R} \cap [0, 1]$
 - points réels dans l'espace 3D : $\mathcal{A}_{3D}$
 - sous-ensembles des nombres naturels : $\mathcal{P}(\mathbb{N})$
 - sous-ensembles des nombres réels : $\mathcal{P}(\mathbb{R})$
 - ...
- Ces ensembles sont-ils de même cardinalité ?
Comment comparer leurs cardinalités ?

Wallis (1656) : notation ∞

Cantor (1887) : il y a plusieurs infinis !

Définition 7.11.

$|\mathcal{A}| \leq |\mathcal{B}|$ ssi une application injective $f: \mathcal{A} \rightarrow \mathcal{B}$ existe. ▲

Définition 7.12.

$|\mathcal{A}| = |\mathcal{B}|$ ssi une application bijective $f: \mathcal{A} \rightarrow \mathcal{B}$ existe. ▲

Théorème 7.13. (Cantor-Bernstein, aperçu)

$|\mathcal{A}| = |\mathcal{B}|$ ssi $|\mathcal{A}| \leq |\mathcal{B}|$ et $|\mathcal{B}| \leq |\mathcal{A}|$. ■

On peut démontrer :

$$\begin{aligned} |\mathbb{N}| &= |\mathbb{N}_{\text{pair}}| \\ &< |\mathbb{R}| = |\mathcal{P}(\mathbb{N})| = |\mathbb{R} \cap [0, 1]| = |\mathcal{A}_{3D}| \\ &< |\mathcal{P}(\mathbb{R})| \end{aligned}$$

$|\mathbb{N}|$ est la cardinalité d'un ensemble infini dénombrable, appelée l'infini dénombrable $\aleph_0$ (aleph zéro).

$\mathbb{R}$ n'est pas dénombrable (diagonalisation).

Hypothèse du continuum



(<http://www-groups.dcs.st-and.ac.uk/history/Mathematicians/Cantor.html>)

Mathématicien, théorie des nombres et des ensembles, diagonalisation, bijections, obsession du continuum, idées en avance de son temps : 100 ans trop tôt ?, dépressions après 1884, Shakespeare ?, etc.

Georg Cantor, 1845-1918

Rappel : la diagonalisation

Soit R une relation binaire sur un ensemble A .

Exemple :

$$A = \{a, b, c, d\}$$

R	a	b	c	d
a		✓		✓
b		✓	✓	
c			✓	
d	✓	✓		

Définissons $\forall x \in A$

$$R_x = \{y : y \in A \text{ et } (x, y) \in R\}$$

Exemple :

R	a	b	c	d
a		✓		✓
b		✓	✓	
c			✓	
d	✓	✓		

$$R_a = \{b, d\}$$

$$R_b = \{b, c\}$$

$$R_c = \{c\}$$

$$R_d = \{a, b\}$$

L'ensemble diagonal est

$$D = \{y \mid y \in A \text{ et } (y, y) \notin R\}.$$

Ici : $D = \{a, d\}$.

Le truc de la diagonalisation consistait à observer tout simplement que :

$$(\forall x \in A) R_x \neq D$$

i.e. aucune des "lignes" du tableau de la relation R n'est égale à la "ligne" D .

Application vue en démo

Théorème 7.14. $|\mathbb{N}| < |\mathcal{P}(\mathbb{N})|$.

Preuve.

Supposons au contraire que $|\mathbb{N}| = |\mathcal{P}(\mathbb{N})|$. Alors on peut énumérer les sous-ensembles $\mathcal{B}_0, \mathcal{B}_1, \mathcal{B}_2, \dots$ de $\mathbb{N}$ car bijection $f: \mathcal{P}(\mathbb{N}) \rightarrow \mathbb{N}$.

Mais alors l'ensemble diagonal

$$\{y : y \notin \mathcal{B}_y\}$$

de la relation $\{(x, y) : x \in \mathcal{B}_y\}$ manque à la liste.

Contradiction. Donc $|\mathbb{N}| \neq |\mathcal{P}(\mathbb{N})|$. ■

Retour sur l'encodage des MTs

Revoyons comment décrire sur alphabet $\{0, 1\}$ n'importe quelle MT $M = (Q, \Sigma, \Gamma, \delta, q_0, q_a, q_r)$.

Choisissons l'encodage suivant :

$$111\langle \Sigma \rangle 111\langle \Gamma \rangle 111\langle q_0 \rangle 111\langle q_a \rangle 111\langle q_r \rangle 111\langle \delta \rangle 111$$

où

- les symboles d’alphabet sont encodés en unaire sur l’alphabet $\{0\}$ et un alphabet est une suite de symboles séparés par des 1.
- on suppose que les états sont $\{q_1, q_2, \dots, q_m\}$ et q_i est représenté par 0^i .
- 1. la représentation de $\delta(q_i, \alpha_j) = (q_k, \alpha_l, L)$ est $0^i 1 0^j 1 0^k 1 0^l 1 0$.
- 2. la représentation de $\delta(q_i, \alpha_j) = (q_k, \alpha_l, R)$ est $0^i 1 0^j 1 0^k 1 0^l 1 0 0$.
- 3. δ au complet est encodée entrée par entrée avec séparateurs 11.

Exemple

Soit une MT

$$(\{q_1, q_2, q_3, q_4\}, \{0, 1\}, \{0, 1, B\}, \delta, q_1, q_2, q_4)$$

où voici quelques entrées de δ :

$$\begin{array}{ll} \delta(q_1, 1) = (q_3, 0, R) & \delta(q_3, 0) = (q_1, 1, R) \\ \delta(q_3, 1) = (q_2, 0, R) & \delta(q_3, \sqcup) = (q_4, 1, L). \end{array}$$

Alors l’encodage binaire est :

```

111  0  1  00
111  0  1  00  1  000
111  0
111  00
111 0000
111  0  1  00  1  000  1  0  1  00  11
    000 1  0  1  0  1  00  1  00  11
    000 1  00 1  00  1  0  1  00  11
    000 1 000 1 0000 1 00 1  0  11
    ... 1 ... 1 ... 1 ... 1 ... 11
    ... 1 ... 1 ... 1 ... 1 ... 11
    ... 1 ... 1 ... 1 ... 1 ...
111

```

Remarque 7.15. Pour tout alphabet fini Σ , $|\Sigma^*| = |\mathbb{N}|$.

Pourquoi ? Car l'ordre lexicographique est une bijection entre $\mathbb{N}$ et Σ^* . ▲

Remarque 7.16. L'ensemble $\mathcal{M}_\Sigma$ des machines de Turing sur Σ vérifie $|\mathcal{M}_\Sigma| = |\mathbb{N}|$.

Pourquoi ? Car encodage d'une MT
+ ordre lexicographique des encodages
= bijection entre $\mathcal{M}_\Sigma$ et $\mathbb{N}$. ▲

Langages non reconnaissables

Théorème 7.17. (Sipser 4.18)

Il existe des langages **non reconnaissables**. ■

Preuve.

L'ensemble des langages sur Σ est $\mathcal{P}(\Sigma^*)$.

On a $|\Sigma^*| = |\mathbb{N}|$ et donc $|\mathcal{P}(\Sigma^*)| > |\mathbb{N}| = |\mathcal{M}_\Sigma|$.

Donc certains langages ne correspondent à aucune machine de Turing. ■

En fait il existe un continuum de langages non reconnaissables !

⇒ un langage choisi "au hasard" n'est pas reconnaissable avec probabilité 1.

Peut-on identifier explicitement un langage non reconnaissable ?

■
Les langages non reconnaissables sont-ils des concoctions ésotériques sans lien informatique ?

Le problème de l'arrêt

Donnée : machine de Turing M et mot w .

Question : M **s'arrête-t-elle** sur entrée w ?

Le **problème d'arrêt** sous forme de langage :

HALT_{TM} =

$\{ \langle M, w \rangle \mid \text{la machine de Turing } M \text{ ne} \\ \text{boucle pas à l'infini sur entrée } w \},$

ou de manière équivalente

$\{ \langle M, w \rangle \mid \text{la machine de Turing } M \\ \text{sur entrée } w \text{ accepte ou rejette} \}.$

Théorème 7.18. (Sipser 5.1) HALT_{TM} est indécidable.

Preuve. Soit $M_1, M_2, \dots$ la liste de toutes les machines de Turing.

Appelons R la relation sur $\mathbb{N}$:

$$R = \{(x, w) : M_x \text{ s'arrête sur } w\}$$

Visuellement :

R	1	2	3	$\dots$	w	$\dots$
1		✓		$\dots$	✓	$\dots$
2			✓	$\dots$		$\dots$
3	✓		✓	$\dots$	✓	$\dots$
$\vdots$				$\vdots$		$\vdots$
x		✓		✓		$\dots$
$\vdots$						$\vdots$

où ✓ comme d'habitude signifie "reliés".

Supposons au contraire une MT H qui décide HALT_{TM} . ■

Alors la MT suivante

sur entrée w :

simuler H sur $\langle M_w, w \rangle$
si H accepte alors boucler
si H refuse alors accepter

est parfaitement légitime et ■

elle s'arrête sur w

$\iff$

M_w ne s'arrête pas sur w .

Cette MT a aussi un numéro x dans la liste de toutes les machines de Turing. ■

Sa "ligne" R_x est :

$$\begin{aligned} R_x &= \{w : w \in \mathbb{N} \text{ et } (x, w) \in R\} \blacksquare \\ &= \{w : M_x \text{ s'arrête sur } w\} \blacksquare \\ &= \{w : M_w \text{ ne s'arrête pas sur } w\} \blacksquare \\ &= \{w : w \in \mathbb{N} \text{ et } (w, w) \notin R\} \blacksquare \\ &= \text{l'ensemble diagonal } D. \end{aligned}$$

Contradiction. Donc HALT_{TM} est indécidable.

Le problème de l'appartenance

Théorème 7.19. (Sipser 4.11)

Le langage A_{TM} défini comme

$$\{\langle M, w \rangle : M \text{ une MT et } M \text{ accepte } w\}$$

est **indécidable**.

Preuve.

Il suffit de montrer que $HALT_{TM} \leq A_{TM}$.

Comment faire ceci ?

Par la fonction **calculable**

$$f : \Sigma^* \rightarrow \Sigma^* \\ \langle M, w \rangle \mapsto \langle M', w \rangle$$

où M' sur son entrée x **simule** M sur x et **accepte** si M **s'arrête**. ■

Alors pour tout $\langle M, w \rangle$

$$\begin{aligned} \langle M, w \rangle \in HALT_{TM} &\Leftrightarrow M \text{ s'arrête sur } w \\ &\Leftrightarrow M' \text{ accepte } w \\ &\Leftrightarrow \langle M', w \rangle \in A_{TM} \\ &\Leftrightarrow f(\langle M, w \rangle) \in A_{TM}. \blacktriangle \end{aligned}$$

À la recherche d'un langage non reconnaissable explicite

Théorème 7.20. (Sipser 4.22)

Un langage L est **décidable** ssi L et $\bar{L}$ sont **reconnaissables**.

Preuve. $\Rightarrow$: facile...pourquoi ?

$\Leftarrow$: Soient M_1 et M_2 **reconnaissant** L et $\bar{L}$. ■

Construisons M qui, sur entrée w :

- simule **en parallèle** M_1 et M_2 sur w . ■
- accepte dès que M_1 accepte w ,
- rejette dès que M_2 accepte w . ■

Alors M **décide** L car

- $w \in L \Rightarrow M_1$ accepte $w \Rightarrow M$ accepte w .
- $w \notin L \Rightarrow M_2$ accepte $w \Rightarrow M$ rejette w .
- M **s'arrête à coup sûr**. ■

▲

Est-ce que A_{TM} est reconnaissable ? ■

Oui ! ■

Par la MT qui sur entrée $\langle M, w \rangle$, fait exactement comme M ferait sur w (i.e. simule M sur w) en

- acceptant si M accepte, ■
- rejetant si M rejette, ■
- bouclant si M boucle.

Remarque 7.21. Cette MT qui reconnaît A_{TM} est “universelle” puisqu’elle simule toutes les autres MTs. ▲

Le fait que A_{TM} soit reconnaissable bien que indécidable nous mène à notre premier langage non reconnaissable :

Corollaire 7.22. (Sipser 4.23)

Le langage $\overline{A_{TM}}$ n’est pas reconnaissable.

Preuve. ■

Si $\overline{A_{TM}}$ était reconnaissable, alors A_{TM} et son complément seraient tous deux reconnaissables. ■

Alors A_{TM} serait décidable. ■

Or A_{TM} est indécidable. ■

D’autres problèmes indécidables

Pour démontrer qu’un langage L est indécidable :

- preuve par diagonalisation, ou
- preuve par réduction $A \leq L$ sachant que A est un langage indécidable, ou
- preuve par “réduction” plus générale : en se servant d’une hypothétique machine M décidant L pour parvenir à décider un langage indécidable A .

Théorème 7.23. $\overline{A_{TM}} \leq VIDE_{TM}$, où le langage $VIDE_{TM} = \{ \langle M \rangle : L(\text{la MT } M) = \emptyset \}$.

■

Preuve. Par la fonction

$$f : \Sigma^* \rightarrow \Sigma^* \\ \langle M, w \rangle \mapsto \langle M' \rangle$$

où M' est la MT qui fait sur entrée x :
si $x \neq w$ alors rejeter
sinon simuler M sur w .

Ainsi f est calculable (n'est-ce pas ?). ■

Et pour tout $\langle M, w \rangle$

$$\begin{aligned}\langle M, w \rangle \in A_{TM} &\Leftrightarrow M \text{ accepte } w \\ &\Leftrightarrow L(M') = \{w\} \\ &\Leftrightarrow L(M') \neq \emptyset \\ &\Leftrightarrow f(\langle M, w \rangle) \notin VIDE_{TM}\end{aligned}$$

d'où

$$\langle M, w \rangle \in \overline{A_{TM}} \Leftrightarrow f(\langle M, w \rangle) \in VIDE_{TM}. \blacktriangle$$

Corollaire 7.24. (Sipser 5.28)
Le langage $VIDE_{TM}$ est indécidable.

Preuve. Découle du théorème 7.23 puisque $\overline{A_{TM}}$ n'est pas reconnaissable et a fortiori n'est pas décidable. ■

Proposition 7.25. (Sipser 5.2) Si $A \leq B$ et B est reconnaissable alors A est reconnaissable.

Preuve. Exercice. ▲

On peut donc dire mieux :

Corollaire 7.26. Le langage $VIDE_{TM}$ n'est pas reconnaissable.

Preuve. $\overline{A_{TM}} \leq VIDE_{TM}$ par le théorème 7.23 et A_{TM} n'est pas reconnaissable. ■

Théorème 7.27. (Sipser 5.3) Le langage REG_{TM} , soit

$$\{ \langle M \rangle : L(\text{la MT } M) \text{ est en réalité régulier} \}$$

n'est pas reconnaissable (donc ni décidable).

Preuve. Suffit de montrer $\overline{A_{TM}} \leq REG_{TM}$ en modifiant la réduction f du théorème 7.23 de manière à ce que

$$M \text{ accepte } w \text{ ssi } L(M') \text{ n'est pas régulier.}$$

Comment ?

Sur entrée $\langle M, w \rangle$, f produit maintenant la MT M' qui fait **sur son entrée x** :

simuler M sur entrée w (peut boucler ici)
 si M accepte w alors
 si x est de la forme $0^n 1^n$ accepter
 sinon rejeter
 sinon rejeter.

Le reste ne change pas.



Réductions via l'historique

Idée : considérer les configurations successives d'un calcul d'une MT comme formant un mot. ■

Reprenons l'exemple du tableau des configurations d'une MT qui accepte le mot $w_1 w_2 \dots w_n$:

q_0	w_1	w_2	$\dots$	$\sqcup$	$\sqcup$	$\dots$	$\dots$	$\sqcup$
d	q_4	w_2	$\dots$	$\sqcup$	$\sqcup$	$\dots$	$\dots$	$\sqcup$
d	c	q_{67}	$\dots$	$\sqcup$	$\sqcup$	$\dots$	$\dots$	$\sqcup$
d	q_{31}	c	$\dots$	$\sqcup$	$\sqcup$	$\dots$	$\dots$	$\sqcup$
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$
c	a	b	$\dots$	b	q_{52}	c	$\dots$	a
c	a	b	$\dots$	q_8	b	d	$\dots$	a
c	a	b	$\dots$	f	q_{12}	d	$\dots$	a
c	a	b	$\dots$	f	g	q_7	$\dots$	a
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$
$\dots$	$\dots$	$\dots$	c	q_9	d	$\dots$	$\dots$	$\dots$
$\dots$	$\dots$	$\dots$	q_a	c	a	$\dots$	$\dots$	$\dots$

Fixons une MT M et un mot w .

M accepte w ssi il existe un mot

$$\#C_1\#C_2\#C_3\#C_4\#\dots\#C_l\#$$

tels que

- chaque C_i est une configuration de M ,
- C_1 est la configuration initiale de M sur w ,
- l'état de M en configuration C_l est q_a , et
- $C_1 \vdash_M C_2 \vdash_M C_3 \vdash_M \dots \vdash_M C_l$.

Ou encore (nous verrons l'intérêt bientôt) :

M accepte w ssi il existe un mot

$$\#C_1\#C_2^R\#C_3\#C_4^R\#\dots\#C_i\#$$

vérifiant ces mêmes conditions, où C_i^R est C_i écrit de droite à gauche.

Appelons maintenant

$$\text{CALCUL}(M, w)$$

le langage qui contient l'unique mot ci-dessus si M accepte w , et qui est vide sinon.

L'intérêt ?

Il est possible de calculer à partir de M et de w une GHC G ayant $L(G) = \overline{\text{CALCUL}(M, w)}$.

Ceci pourra servir à une réduction puisque

M refuse w ss:

Comment calculer G ?

Résumons : $\overline{\text{CALCULS}(M, w)}$ est formé des mots

$$\#C_1\#C_2^R\#C_3\#C_4^R\#\dots\#C_i\#$$

tels que

1. il existe un C_i qui n'est pas une configuration de M , ou encore
2. C_1 n'est pas la configuration initiale de M sur w , ou encore
3. l'état de M en configuration C_i n'est pas q_a , ou encore
4. pour un certain i , $C_i \xrightarrow{M} C_{i+1}$ ne tient pas.

Un mot $x \in \overline{\text{CALCULS}(M, w)}$ ssi

1. x n'est pas de la forme $\#(\Gamma^*Q\Gamma^*\#)^*$ (ER)
2. ou x ne commence pas par $\#q_0w$ (ER)
3. ou x ne finit pas par $\Gamma^*q_a\Gamma^*\#$ (ER)
4. ou x inclut une transition invalide, ie
 - 4.1 ou bien $C_{2i}^R\#C_{2i+1}$ incorrect (APN)
 - 4.2 ou bien $C_{2i-1}\#C_{2i}^R$ incorrect (APN)

Union de langages hors contextes !

Théorème 7.28. (Sipser 5.13) Le langage $\text{TOUT}_{\text{GHC}} = \{ \langle G \rangle \mid \text{le langage de la grammaire hors-contexte } G \text{ est } \Sigma^* \}$ est **indécidable**.

Preuve. $\overline{\text{A}_{\text{TM}}} \leq \text{TOUT}_{\text{GHC}}$ car on a vu une fonction qui calcule sur entrée $\langle M, w \rangle$ une GHC G engendrant $\overline{\text{CALCULS}(M, w)}$. Or

$$\begin{aligned} \langle M, w \rangle \notin \text{A}_{\text{TM}} &\Leftrightarrow \text{CALCULS}(M, w) = \emptyset \\ &\Leftrightarrow \text{CALCULS}(M, w) = \Sigma^* \\ &\Leftrightarrow G \in \text{TOUT}_{\text{GHC}}. \end{aligned}$$

Donc TOUT_{GHC} n'est pas reconnaissable, et a fortiori n'est pas décidable. ■

(Rappel :) Le langage EGAL_{GHC} défini comme

$$\left\{ \langle G, H \rangle : G, H \text{ sont GHCs, } L(G) = L(H) \right\},$$

est-il décidable ?

■
NON !

Car $\text{TOUT}_{\text{GHC}} \leq \text{EGAL}_{\text{GHC}}$ (exercice).

Grammaires générales

Définition 7.29. Une **grammaire générale** est un quadruplet (V, Σ, R, S) comme dans le cas d'une GHC, sauf que les règles sont de la forme

$$\alpha \rightarrow \beta$$

où

$$\alpha \in (V \cup \Sigma)^* V (V \cup \Sigma)^*.$$

▲

Exemple de grammaire générale

$G = (V, \{a\}, R, S)$ où $V = \{A, B, C, D, E\}$ et les règles sont

$$\begin{array}{lll} S \rightarrow ACaB & Ca \rightarrow aaC & CB \rightarrow DB \\ aD \rightarrow Da & AD \rightarrow AC & \\ CB \rightarrow E & aE \rightarrow Ea & AE \rightarrow \varepsilon \end{array}$$

On a que $L(G) = \{a^{2^k} : k = 1, 2, \dots\}$.

Faits remarquables

Théorème A : Étant donné une grammaire générale G , il est possible de construire une MT M telle que $L(M) = L(G)$.

Théorème B : Étant donné une MT M , il est possible de construire une grammaire générale G telle que $L(M) = L(G)$.

Et le langage A_{CG} défini comme

$$\{\langle G, w \rangle : G \text{ est une GG et } w \in L(G)\}$$

est indécidable.

La grammaire générale est donc à la MT ce que la GHC est à l'APN.

Les 4 prochains transparents esquissent les preuves des théorèmes A et B pour votre culture personnelle...

Théorème A : Étant donné une grammaire générale G , il est possible de construire une MT M telle que $L(M) = L(G)$.

Idée de la preuve :

On construit une MT non-déterministe :

- un ruban pour l'entrée
- l'alphabet de ruban inclut les variables de G
- un autre ruban pour les étapes de la dérivation, c-à-d des chaînes de la forme $(V \cup \Sigma)^*$
- la MT ne modifie jamais le premier ruban.

1. Pour commencer, avec l'entrée w sur le premier ruban, M écrit S sur le deuxième.

2. M choisit une position sur le deuxième ruban d'une manière non déterministe.

3. M choisit une règle $\alpha \rightarrow \beta$. Si α est sous la deuxième tête, alors M remplace α par β (en décalant si $|\alpha| \neq |\beta|$).

4. M compare les mots sur les deux rubans : si les mêmes, M accepte, sinon retourne à l'étape 2. ▲

Théorème B : Étant donné une MT M , il est possible de construire une grammaire générale G telle que $L(M) = L(G)$.

Preuve : Soit $M = (Q, \Sigma, \Gamma, \delta, q_0, q_a, q_r)$. On construit $G = (V, \Sigma, R, S)$ qui génère deux copies d'un mot, simule M sur une des copies et ne retient que l'autre copie advenant le succès de M . On pose $V = ((\Sigma_\varepsilon \times \Gamma) \cup \{S, C, I\})$ et :

1. Pour l'effet de la configuration initiale :
 $S \rightarrow q_0 C$.
2. Pour arriver à deux copies du mot sur l'entrée :
 $C \rightarrow (\sigma, \sigma) C$ où $\sigma \in \Sigma$.
3. Pour l'effet du ruban infini :
 $C \rightarrow I, I \rightarrow (\varepsilon, \sqcup) I \mid \varepsilon$.

4. Pour l'effet des transitions :

4.1 $q(\sigma, b) \rightarrow (\sigma, b')q'$
où $\sigma \in \Sigma_\varepsilon, b, b' \in \Gamma, q, q' \in Q$ t.q. $\delta(q, b) = (q', b', R)$

4.2 $(\sigma, a)q(\sigma', b) \rightarrow q'(\sigma, a)(\sigma', b')$
où $\sigma, \sigma' \in \Sigma_\varepsilon, a, b, b' \in \Gamma, q, q' \in Q$ tels que $\delta(q, b) = (q', b', L)$

5. Pour l'effet de l'état acceptant :
 $(\sigma, b)q_a \rightarrow q_a \sigma q_a, q_a(\sigma, b) \rightarrow q_a \sigma q_a, q_a \rightarrow \varepsilon$
où $\sigma \in \Sigma_\varepsilon, b \in \Gamma$.



Autres problèmes indécidables

Est-ce que tous les problèmes indécidables sont au sujet de la théorie des langages ?

Non !

Quelques exemples...

PCP (Prob. correspondance de Post)

Donnée : ensemble de tuiles, où chaque tuile est une paire de mots placés l'un au-dessus de l'autre.

Question : est-il possible de juxtaposer des tuiles pour former en haut et en bas la même concaténation de mots ?

10-ième problème de Hilbert

Donnée : $p(x_1, \dots, x_n)$ un polynôme à coefficients entiers.

Question : est-ce que $p(x_1, \dots, x_n) = 0$ a une solution entière? (Matiyasevich 1970)

Jeu de life (Conway)

Donnée : configuration initiale du jeu.

Question : la population croîtra-t-elle à l'infini ?

NB : Chaque case est adjacente à 8 cases et est occupée par une cellule, ou libre. À chaque étape,

1. une cellule naît dans chaque case libre ayant précisément 3 cases adjacentes occupées
2. une cellule meurt si elle a 0 ou 1 (ennui) ou 4 ou 5 ou 6 ou 7 ou 8 (surpeuplement) voisines
3. les cellules ayant 2 ou 3 voisines survivent.

