IFT-2002

Informatique Théorique

H14 - cours 10

Julien Marcil - julien.marcil@ift.ulaval.ca

Théorème

Soit un le langage L. Si L est *Turing-décidable*, alors $\overline{L}$ est *Turing-décidable*.

Théorème

Soit un le langage L. Si L et $\overline{L}$ sont *Turing-acceptable*, alors L est *Turing-décidable*.

Aujourd'hui

- Décidabilité
- Réduction

Décidabilité

Notation

Soit un programme M, alors on note $\langle M \rangle$ la chaîne de symboles qui représente M.

Différence entre M et $\langle M \rangle$

Lorsque l'on parle d'un « programme » on ne fait souvent pas la distinction entre le code source du programme et l'exécutable qui lui correspond.

La différence entre M et $\langle M \rangle$ est de la même nature.

- M est une « machine » donc un processus automatique, un exécutable, quelque chose qui reçoit une entrée et retourne (peut-être) une sortie.
- $\langle M \rangle$ est une suite de 0 et de 1. On choisit d'interpréter cette suite de 0 et de 1 comme ayant un sens précis, celui de l'encodage de M.

La Trahison des images (1929, huile sur toile, 59 × 65 cm), René Magritte

A_{AFD}

$$A_{\text{AFD}} = \{\langle B, w\rangle \mid B \text{ est un AFD qui accepte } w\}$$

AFD: automate fini déterminite

Théorème

A_{AFD} est un langage décidable.

Démonstration

Soit la machine de Turing $M_{A_{\mathrm{AFD}}}$ qui décide A_{AFD} .

$M_{A_{\mathrm{AFD}}} =$ `Avec` $\langle B, w \rangle$

1. `Simuler` B `sur` w.
2. `Si la simulation termine sur un état accepteur alors` ACCEPTE. `Si la simulation termine sur un état non-accepteur alors` REJETTE.

A_{AFN}

$$A_{\text{AFN}} = \{\langle B, w\rangle \mid B \text{ est un AFN qui accepte } w\}$$

AFN: automate fini non déterminite

Théorème

A_{AFN} est un langage décidable.

Démonstration

Soit la machine de Turing $M_{A_{\mathrm{AFN}}}$ qui décide A_{AFN} .

$M_{A_{\mathrm{AFN}}} =$ Avec $\langle B, w\rangle$

1. Covertir B en C un AFD équivalent.
2. Executer $M_{A_{\mathrm{AFD}}}$ sur $\langle C, w\rangle$
3. Si $M_{A_{\mathrm{AFD}}}$ accepte alors ACCEPTE sinon REJETTE .

E_{AFD}

$$E_{\text{AFD}} = \{\langle B \rangle \mid B \text{ est un AFD et } L(B) = \emptyset\}$$

Théorème

E_{AFD} est un langage décidable.

Démonstration

Soit la machine de Turing $M_{E_{\text{AFD}}}$ qui décide E_{AFD}.

$M_{E_{\text{AFD}}} =$ `Avec` $\langle B \rangle$

1. Marquer l'état initial de B.
2. Répéter jusqu'à ce qu'il n'y est plus d'état à marquer
   1. Marquer un état pour lequel il existe une transition venant d'un état déjà marqué.
3. Si aucun état accepteur de B n'est marqué alors $\texttt{ACCEPTE}$ sinon $\texttt{REJETTE}$.

EQ_{AFD}

$$EQ_{\text{AFD}} = \{\langle A, B\rangle \mid A \text{ et } B \text{ sont des AFD et } L(A) = L(B)\}$$

Théorème

EQ_{AFD} est un langage décidable.

Théorie des ensembles

Soit C un AFD tel que

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

$$L(A) = L(B) \;\Leftrightarrow\; L(C) = \emptyset$$

Démonstration

Soit la machine de Turing $M_{EQ_{\text{AFD}}}$ qui décide EQ_{AFD}.

$M_{EQ_{\text{AFD}}}$ = Avec $\langle A, B \rangle$

1. Construire l'automate fini déterministe C à partir de A et B.
2. Executer $M_{E_{\text{AFD}}}$ sur $\langle C \rangle$
3. Si $M_{E_{\text{AFD}}}$ accepte alors $\mathtt{ACCEPTE}$ sinon $\mathtt{REJETTE}$.

A_{GHC}

$$A_{\text{GHC}} = \{\langle G, w\rangle \mid G \text{ est une GHC qui génère } w\}$$

GHC: grammaire hors context

Théorème

A_{GHC} est un langage décidable.

Forme normale de Chomsky

Définition: Soit $G = (V, \Sigma, S, R)$ une grammaire. G est dans la **forme normale de Chomsky** si les *règles de réécriture* sont de la forme:

- $A \to BC$ pour $A, B, C \in V$ et $B \neq S$ et $C \neq S$
- $A \to a$ pour $A \in V$, $a \in \Sigma$.
- $S \to \lambda$ pour le *symbole de départ* S.

Démonstration

Soit la machine de Turing $M_{A_{\text{GHC}}}$ qui décide A_{GHC}.

$M_{A_{\text{GHC}}} =$ Avec $\langle G, w \rangle$

1. Convertir G à une grammaire équivalente dans la forme normale de Chomsky
2. Lister toutes les dérivations de $2n-1$ productions (ou de seulement une production si $w = \lambda$)
3. Si un des mots générés est w alors $\mathrm{ACCEPTE}$ sinon $\mathrm{REJETTE}$.

E_{GHC}

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

Théorème

E_{GHC} est un langage décidable.

Démonstration

Soit la machine de Turing $M_{E_{\text{GHC}}}$ qui décide E_{GHC}.

$M_{E_{\text{GHC}}} =$ `Avec` $\langle G \rangle$

1. Marquer tous les symboles terminaux de G.
2. Répéter jusqu'à ce qu'il n'y est plus de variables à marquer
   1. Marquer une variable A tel que G a une règle $A \to U_1 \cdots U_k$ et que les symboles $U_1, \ldots, U_k$ sont tous marqués.
3. Si la variable initiale est marquée alors $\mathtt{ACCEPTE}$ sinon $\mathtt{REJETTE}$.

EQ_{GHC}

$$EQ_{\text{GHC}} = \{\langle G, H\rangle \mid G \text{ et } H \text{ sont des GHC et } L(G) = L(H)\}$$

Problème

Les langages hors contexte ne sont pas fermés sur les opérations *complément* et *intersection*.

Théorème

Soit le langage L.

$$L \text{ est hors contexte} \Rightarrow L \text{ est décidable}$$

Démonstration

L est hors contexte si il exite une grammaire G qui génère L.

Soit la machine de Turing M_G qui décide $L(G)$.

$M_G =$ `Avec` $\langle w \rangle$

1. `Exécuter` $M_{A_{\text{GHC}}}$ `sur` $\langle G, w \rangle$
2. `Si` $M_{A_{\text{GHC}}}$ `accepte alors` ACCEPTE `sinon` REJETTE.

Indécidabilité

A_{MT}

$$A_{\mathrm{MT}} = \{\langle M, w\rangle \mid M \text{ est une machine de Turing qui accepte } w\}$$

MT: machine de Turing

Théorème

A_{TM} est un langage *Turing-acceptable*.

Théorème

A_{TM} n'est pas un langage décidable.

Corrolaire

$\overline{A_{\mathrm{TM}}}$ n'est pas un langage *Turing-acceptable*.

$HALT_{\mathrm{MT}}$

$$HALT_{\mathrm{MT}} = \{\langle M, w\rangle \mid M \text{ est une MT et } M \text{ s'arrête sur } w\}$$

Théorème

$HALT_{\mathrm{MT}}$ n'est pas un langage décidable.

Réduction

Réduction

Une **réduction** est un algorithme transformant un problème en un autre.

Si un problème A peut être réduit à (i.e. transformé en) un problème B, et que le problème A est difficile alors le problème B est au moins aussi difficile. On écrit alors $A \leq_m B$.

Réduction

Un language L se réduit au language K, noté $L \leq_m K$ si il existe $f : \Sigma^* \rightarrow \Sigma^*$ une fonction calculable tel que

$$\forall_{w \in \Sigma^*} \quad w \in L \Leftrightarrow f(w) \in K$$

Théorème

Si $A \leq_m B$ et B est décidable, alors A est décidable.

Corrolaire

Si $A \leq_m B$ et A n'est pas décidable, alors B n'est pas décidable.

Théorème

$A_{\mathrm{MT}} \leq_m HALT_{\mathrm{MT}}$

Ceci implique que $HALT_{\mathrm{MT}}$ n'est pas un langage décidable.

E_{MT}

$$E_{\mathrm{MT}} = \{\langle M \rangle \mid M \text{ est une MT et } L(M) = \emptyset\}$$

Théorème

$A_{\mathrm{MT}} \leq_m E_{\mathrm{MT}}$

Ceci implique que E_{MT} n'est pas un langage décidable.

$REGULIER_{\mathrm{MT}}$

$REGULIER_{\mathrm{MT}} = \{\langle M \rangle \mid M \text{ est une MT et } L(M) \text{ est régulier}\}$

Théorème

$A_{\mathrm{MT}} \leq_m REGULIER_{\mathrm{MT}}$

Ceci implique que $REGULIER_{\mathrm{MT}}$ n'est pas un langage décidable.

EQ_{MT}

$EQ_{\mathrm{MT}} = \{\langle M_1, M_2\rangle \mid M_1 \text{ et } M_2 \text{ sont des MT et } L(M_1) = L(M_2)\}$

Théorème

$E_{\mathrm{MT}} \leq_m EQ_{\mathrm{MT}}$

Ceci implique que EQ_{MT} n'est pas un langage décidable.