Correction devoir 4

#2. $L_{\overline{REC}} = \{\langle M\rangle \mid L(M) \in \overline{REC}\}$

$REC = \{L \mid$ il existe une M.T. t.q. $L(M) = L\}$

#3 Il y a un nombre fini de MT possédant 13 états. En effet, la table de transition est de grosseur fixe et il y a un nombre fini de possibilités de transitions

$$M = (\underbrace{Q}_{13\text{ états}}, \underbrace{\Sigma, \Gamma}_{\text{sont fixes}}, \delta, \underbrace{q_0}_{13\text{ poss. de } q_0}, F)$$

Il y a une façon de compter le nombre de machines.

$13 \cdot n \cdot 2 = 26n$

	q_0	q_1	...	--	q_{12}
a_1	$(q_i, a_j, D/G)$				
a_2					
$\vdots$					
a_n					

Soit S l'ensemble des langages reconnus par cet ensemble de MT à 13 états.

$\Rightarrow L_{13} = \{\langle M \rangle \mid M \text{ est une MT et } L(M) \in S\}$

Il existe une MT M^* t.q. $L(M^*) \in S$
en prenant M^* une machine à 13 états.

Il existe une MT M^+ t.q. $L(M^+) \notin S$
car S contient un nombre fini de langages
et il existe un nombre infini de langages
reconnaissables.

$\Rightarrow L_{13} \notin DEC$ par le théorème de Rice.

#4. 2 méthodes:

1) Réduction: $\overline{A_{MT}} \leq L_n$

2) $\underbrace{L_n \notin DEC}_{A_{MT} \leq L_n}$ et $\underbrace{\overline{L_n} \in REC}_{\text{Trouver une MT}}$

2) $\underline{L_n \notin DEC}$

$VIDE_{MT} \leq L_n$

$A_{MT} = \{\langle M, w \rangle \mid M \text{ une MT}, M \text{ accepte } w\}$

On veut une fonction calculable

$\overline{A_{MT}} = \{y \mid y$ pas de la forme $\langle M, w \rangle$ ou y de la forme $\langle M, w \rangle$ et M rejette ou boucle sur $w\}$

$f: \Sigma^* \rightarrow \Sigma^*$

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

$x \in VIDE_{MT} \Leftrightarrow f(x) \in L_n$

$\Updownarrow$

$\overline{VIDE_{MT}} = \{y \mid y$ pas de la forme $\langle M \rangle$ ou $y = \langle M \rangle$ et le $L(M) \neq \emptyset\}$

$x = \langle M \rangle$ t.q. $L(M) = \emptyset \Leftrightarrow f(x) = \langle M_1, M_2 \rangle$ t.q. $L(M_1) \cap L(M_2) = \emptyset$

Voici f:

Input: x

Si x n'est pas de la forme $x = \langle M \rangle$ alors $f(x) = \varepsilon$

Si $x = \langle M \rangle$ alors $f(x) = \langle M, M_+ \rangle$ où M_+ est une machine qui accepte tout $(L(M_+) = \Sigma^*)$

① f est belle et bien calculable.

② $x \in VIDE_{MT}$: $x = \langle M \rangle$ et $L(M) = \emptyset$

$f(x) = \langle M, M_+ \rangle$

$L(M) \cap L(M_+) = \emptyset \cap \Sigma^* = \emptyset$

③ $x \notin VIDE_{MT}$:

3.1 x pas de la forme $\langle M \rangle$

$\Rightarrow f(x) = \varepsilon$

$\Rightarrow f(x) \notin L_\cap$

3.2 $x = \langle M \rangle$ et $L(M) \neq \emptyset$

$\Rightarrow$ il y a au moins un mot $w \in L(M)$

$\Rightarrow f(x) = \langle M, M_+ \rangle$

$w \in L(M)$, $w \in L(M_+)$ $\Rightarrow L(M) \cap L(M_+) \neq \emptyset$

$\Rightarrow f(x) \notin L_\cap$

$\Rightarrow L_\cap \notin DEC$

$\overline{L_n} \in REC$

$L_n = \{ y \mid y$ pas de la forme $\langle M_1, M_2 \rangle$
ou $y = \langle M_1, M_2 \rangle$ et $L(M_1) \cap L(M_2) \neq \emptyset \}$

Voici une MT qui reconnaît $\overline{L_n}$.

- Input: y
- Si y pas de la forme $y = \langle M_1, M_2 \rangle$
  $\hookrightarrow$ accepter y
- Si $y = \langle M_1, M_2 \rangle$

  Simuler en diagonale sur chaque mot de Σ^* en ordre lexicographique pour les deux machines M_1 et M_2.
  $\hookrightarrow$ Simuler M_1 sur le mot w pendant k étapes et
  Simuler M_2 sur le mot w pendant k étapes
  où w et k sont obtenus par diagonalisation

Accepte y si M_1 et M_2 acceptent w
Sinon on continue la simulation en diagonale

—//—

① Si $y \in \overline{L_n}$:

(1.1) y pas de la forme $\langle M_1, M_2 \rangle$
$\Rightarrow$ Accepter y

(1.2) $y = \langle M_1, M_2 \rangle$ et $L(M_1) \cap L(M_2) \neq \emptyset$
$\Rightarrow \exists w$ t.q. $w \in L(M_1)$ et $w \in L(M_2)$
Donc il existe un nombre d'étapes (de transitions) t.q. les deux machines vont accepter w. Donc on va accepter y.

② Si $y \notin \overline{L_\cap}$: $y = \langle M_1, M_2 \rangle$ et $L(M_1) \cap L(M_2) = \emptyset$

$\Rightarrow$ Aucun mot n'est accepté par les deux MT.
$\Rightarrow$ Boucler sur y.

(suite ici)

1) $\overline{A_{MT}} \leq L_\cap$
On veut une fonction calculable
$f : \Sigma^* \to \Sigma^*$

$$x \in \overline{A_{MT}} \Leftrightarrow f(x) \in L_\cap$$

si x pas $\langle M, w \rangle$ ou $x = \langle M, w \rangle$ et M rej/boucle sur w
$\Rightarrow f(x) = \langle M_1, M_2 \rangle$ et $L(M_1) \cap L(M_2) = \emptyset$

$x = \langle M, w \rangle$ et M accepte w $\Rightarrow$ $f(x)$ pas une $\langle M_1, M_2 \rangle$ ou $f(x) = \langle M_1, M_2 \rangle$ et $L(M_1) \cap L(M_2) \neq \emptyset$.

Voici f:

- Input: x
- Si x n'est pas de la forme $x = \langle M, w \rangle$ alors alors $f(x) = \langle M^*, M^* \rangle$ où M^* est une MT qui rejette tout donc $L(M^*) = \emptyset$
- Si $x = \langle M, w \rangle$ alors $f(x) = \langle M^+, M^+ \rangle$ où la MT M^+ est:
  - Input: z
  - Simule M sur w
    Si M rejette w, alors rejette z.
    Si M accepte w, alors accepte z.

① f est belle et bien calculable

② $x \in \overline{A_{MT}}$

2.1 x n'est pas de la forme $\langle M, w \rangle$

$f(x) = \langle M^*, M^* \rangle$

$L(M^*) \cap L(M^*) = \emptyset \cap \emptyset = \emptyset$

$\Rightarrow f(x) \in L_\cap$

2.2 $x = \langle M, w \rangle$ et M rej/boucle sur w

$f(x) = \langle M^+, M^+ \rangle$

Si M rejette w, M^+ rejette tout $\Rightarrow L(M^+) = \emptyset$

Si M boucle sur w, M^+ boucle sur tout $\Rightarrow L(M^+) = \emptyset$

$L(M^+) \cap L(M^+) = \emptyset \cap \emptyset = \emptyset$

$\Rightarrow f(x) \in L_\cap$

③ $x \notin \overline{A_{MT}}$ $x = \langle M, w \rangle$ et M accepte w

$f(x) = \langle M^+, M^+ \rangle$

Si M accepte w, M^+ accepte tout

$\Rightarrow L(M^+) = \Sigma^*$

$L(M^+) \cap L(M^+) = \Sigma^* \cap \Sigma^* \neq \emptyset$

$\Rightarrow f(x) \notin L_\cap$

$\Rightarrow \overline{A_{MT}} \leq L_\cap$

$\Rightarrow L_\cap \notin REC$

#5. Montrons que $A_{MT} \leq L_{\neq}$
On veut une fonction calculable
$f : \Sigma^* \to \Sigma^*$

$x \in A_{MT} \Leftrightarrow f(x) \in L_{\neq}$

Voici la fonction f
- Input : x
- Si x n'est pas de la forme $x = \langle M, w \rangle$ alors $f(x) = \varepsilon$
- Si $x = \langle M, w \rangle$ alors $f(x) = \langle M_1, M_2, w \rangle$

où M_1 est la MT suivante :
Prend z
Simule M sur w
Si M acc w, accepter z
Si M rej w, rejeter z

et M_2 est la MT suivante
Prend z
Simule M sur w
Si M acc w, rejeter z
Si M rej w, rejetter z

$L_{\neq} \in REC$?
Oui, voici la MT qui reconnait $L_{\neq}$
Prend y
Si y pas de la forme $\langle M, M', x \rangle$
rejette y
Si $y = \langle M, M', x \rangle$
Simule M sur x
Simule M' sur x
Accepte y si (M acc x et M' rej x)
ou si (M rej x et M' acc x)
Rejette y sinon.

alors $f(x) = \langle M, M^*, w \rangle$ où M^* est une MT qui rejette tout

① f est belle et bien calculable

② $x \in A_{MT}$ $x = \langle M, w \rangle$ et M accepte w

→ $f(x) = \langle M, M^*, w \rangle$

M accepte w

M^* rejette w

$\Rightarrow f(x) \in L_{\neq}$

③ $x \notin A_{MT}$

3.1 x n'est pas de la forme $\langle M, w \rangle$

$\Rightarrow f(x) = \varepsilon$

$\Rightarrow f(x) \notin L_{\neq}$

3.2 $x = \langle M, w \rangle$ et M rejette/boucle sur w

$\Rightarrow f(x) = \langle M, M^*, w \rangle$

M rej/boucle sur w

M^* rej w

$\Rightarrow f(x) \notin L_{\neq}$.

$\Rightarrow A_{MT} \leq L_{\neq}$

$\Rightarrow L_{\neq} \notin DEC$

Si $y \in L_{\neq} \Rightarrow y \in \langle M, M', x \rangle$ et (M acc x et M' rej x) ou (M rej x et M' acc x)

$\Rightarrow$ accepte y

Si $y \notin L_{\neq}$ ① $y \neq \langle M, M', x \rangle \Rightarrow$ Rejette y

② $y = \langle M, M', x \rangle$ et au moins une des deux machines boucle sur x

$\Rightarrow$ Boucle sur y

③ $y = \langle M, M', x \rangle$ et les deux acc ou rej $\Rightarrow$ Rej y.

$\underline{\Rightarrow L_{\neq} \in REC}$

#6. $TOUT_{GHC} = \{\langle G \rangle \mid G \text{ est une GHC et } L(G) = \Sigma^*\}$

On veut une fonction calculable

$f : \Sigma^* \to \Sigma^*$

$x \in TOUT_{GHC} \Leftrightarrow f(x) \in INCL_{GHC}$

Voici f:

Input x

Si x n'est pas de la forme $x = \langle G \rangle$ alors $f(x) = \varepsilon$

Si $x = \langle G \rangle$ alors $f(x) = \langle G^*, G \rangle$

où $L(G^*) = \Sigma^* \setminus \{\varepsilon\}$

Une telle GHC existe, on a qu'à prendre une GHC qui engendre Σ^* en FNC et enlever la règle $S \to \varepsilon$.

① f est calculable

② $x \in TOUT_{GHC} : x = \langle G \rangle,\ L(G) = \Sigma^*$

$\Rightarrow f(x) = \langle G^*, G \rangle$

$L(G^*) = \Sigma^* \setminus \{\varepsilon\}$
$L(G) = \Sigma^*$ $\Rightarrow L(G^*) \subset L(G)$ car $\Sigma^* \setminus \{\varepsilon\} \subset \Sigma^*$

$\Rightarrow f(x) \in INCL_{GHC}$

③ $x \notin TOUT_{GHC}$

3.1 x pas de la forme $x = \langle G \rangle$

$\Rightarrow f(x) = \varepsilon$

$\Rightarrow f(x) \notin INCL_{GHC}$

3.2 $x = \langle G \rangle$ et $L(G) \neq \Sigma^*$

$f(x) = \langle G^*, G \rangle$

$L(G^*) = \Sigma^* \setminus \{\varepsilon\}$ $\Rightarrow L(G^*) \not\subset L(G)$

$L(G) \neq \Sigma^*$ $\Rightarrow f(x) \notin INCL_{GHC}$

$\Rightarrow INCL_{GHC} \notin DEC$

C(M,w)

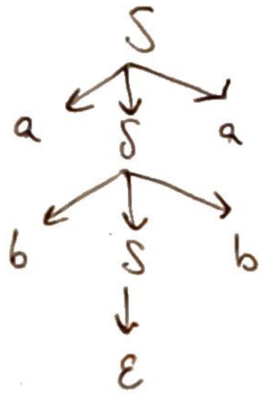
Mots qui représentent une chaîne de transition de la machine M pour accepter w

Diapo 36

$$C(M,w) = \{ \underbrace{C_1}_{\in HC} \# C_2^R \# C_3 \# C_4^R \# \ldots \# C_\ell^{(R \text{ ou pas})} \}$$

$L = \{ w \in \Sigma^* \mid w = w^R \}$

$w = abba$



ab ε ba
= abba