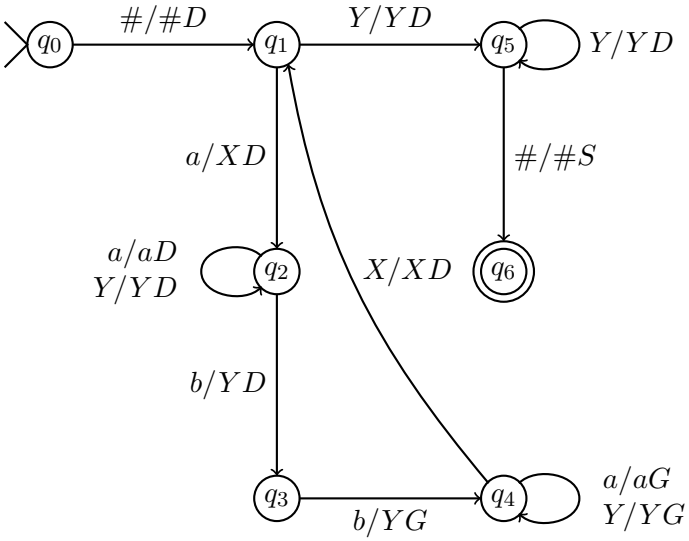
Master Info - 2014-2015
MIF15 Complexité et Calculabilité

Examen Final
Corrigé rédigé par Paul Brunet et Laure Gonnord

Durée 1H30
Notes de cours et de TD autorisées. Livres et appareils électroniques interdits.
Le barème est donné à titre **indicatif** .

1 Machines de Turing

Soit $M = (K, \Sigma, \Gamma, \delta, q_0, F)$ la machine de Turing d'état initial q_0, avec acceptation par état final (unique état acceptant q_6), d'alphabet $\Sigma = \{a, b\}$, d'alphabet de ruban $\Gamma = \{a, b, X, Y, \#\}$, avec la fonction de transition δ schématisée par le dessin ci-dessous :



Question 1 (1 point)
Le mot a^2b^4 est-il accepté par M ? Donner les configurations successives.

Solution:
Partant de la configuration initiale $(\underline{\#}aabbbb\#, q0)$, on obtient sans aucun problème que ce mot est reconnu.

Question 2 (1 point)
Sans donner les configurations successives, dire si les mots a^2b^3 et a^2b^5 sont acceptés par M.

Solution:
Aucun de ces deux mots n'est accepté, le premier mot bloque la machine en q_3, le deuxième dans l'état q_1.

Question 3 (2 points)

Quel langage est reconnu par cette machine de Turing ? *On justifiera proprement par double inclusion.*

Solution:

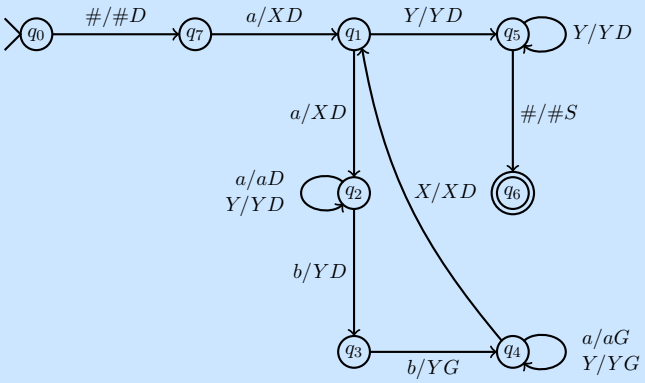
Le langage est $L = \{a^n b^{2n} \mid n \geq 1\}$. Il est clair qu'un mot de L est accepté par la machine de Turing (exhiber une dérivation qui réalise n fois le cycle $(q_1, q_2, q_3, q_4, q_1)$). Dans l'autre sens, un chemin acceptant de la machine de Turing a forcément pris $n > 1$ fois le cycle $(q_1, q_2, q_3, q_4, q_1)$, et ce cycle a pour effet de remplacer un a par X et deux b par un seul Y. Il est assez clair ensuite que les mots reconnus sont uniquement de la forme $a^i b^j$. Par construction, la partie de droite (la fin de l'exécution) de M n'est prise que lorsqu'il ne reste plus de a à lire dans le mot, et cette partie est bloquante s'il reste des b. Ceci montre l'inclusion inverse.

Question 4 (2 points)

Modifier cette machine pour reconnaître le langage $\{a^{n+1} b^{2n} \mid n \geq 1\}$. *Justifier.*

Solution:

Il suffit d'ajouter un état q_7, entre q_0 et q_1, qui remplace le premier a par un X avant d'exécuter la machine M sur le reste du mot.



2 Fonctions primitives récursives et μ-récursives

Question 5 (1 point)

Sous quelle(s) condition(s) (suffisante(s)) une fonction définie par cas est-elle primitive récursive ? *On donnera un énoncé précis, qu'on ne demande pas de prouver, et que l'on considérera comme vrai pour la suite.*

Solution:

Soit $f : \mathbb{N}^p \to \mathbb{N}$ et $g : \mathbb{N}^p \to \mathbb{N}$ deux fonctions primitives récursives et $\pi : \mathbb{N}^p \to \{0, 1\}$ un prédicat primitif récursif, alors $h : \mathbb{N}^p \to \mathbb{N}$ définie par :

$$h(x_1, \ldots, x_p) = \begin{cases} f(x_1, \ldots, x_p) \text{ si } \pi(x_1, \ldots x_p) = 1 \\ g(x_1, \ldots, x_p) \text{ sinon} \end{cases}$$

est primitive récursive.

Question 6 (1 point)

Montrer que la fonction $\mathsf{max_n} : \mathbb{N}^n \to \mathbb{N}$ qui à $(x_1, \dots x_n)$ associe le maximum des n entiers $x_1, \dots x_n$ est primitive récursive. *On peut considérer acquis que le prédicat $\leq$ est primitif récursif. On attend une rédaction avec des arguments précis, mais sans forcément détailler l'utilisation des opérateurs* $\mathsf{Comp}, \mathsf{Rec}, \mathsf{id}$, *...*

Solution:

On montre par récurrence sur $n \geq 2$ que le max n-aire est primitif récursif :

— Pour $n = 2$, $\mathsf{max_2}(x_1, x_2) = \begin{cases} x_1 \text{ si } x_2 \leq x_1 \\ x_2 \text{ sinon.} \end{cases}$ et la question précédente suffisent à conclure.

— Supposons que $\mathsf{max_n}$ d'arité $n \geq 2$ soit primitif. Alors, en écrivant :

$$\mathsf{max_{n+1}}(x_1, \dots, x_n, 0) = \mathsf{max}_n(x_1, \dots, x_n)$$

$$\mathsf{max_{n+1}}(x_1, \dots, x_n, m+1) = \begin{cases} m+1 \text{ si } \mathsf{max}_n(x_1, \dots, x_n) \leq m+1 \\ \mathsf{max}_n(x_1, \dots, x_n) \text{ sinon.} \end{cases}$$

Ceci montre que $\mathsf{max_{n+1}}$. est primitive récursive.

Question 7 (1 point)

Montrer, en utilisant le schéma de minimisation ("schéma-μ"), que les fonctions suivantes sont μ-récursives :

(a) $f(n) = \sqrt{n}$ si n est un carré parfait, indéfini sinon.

(b) $f(x, y) = E(x/y)$ la partie entière de la division de x par y.

Solution:

Sans trop de suspense :

(a) $f(n) = \mu.k[k^2 = n]$

(b) $f(x, y) = \mu.k[k * y > x] - 1$.

3 Grammaires

Question 8 (1 point)

Quels langages sont engendrés par les grammaires suivantes :

(a) $G = (\{S, A\}, \{a, b, c\}, R, S)$ avec $R = \{S \to aS \mid bA, A \to bA \mid c\}$.

(b) $G = (\{S\}, \{a\}, R, S)$ avec $R = \{S \to aSaaS \mid aaa\}$.

On justifiera soigneusement, mais rapidement.

Solution:

Par flemme du correcteur, on donne uniquement les langages.

(a) $L(G) = a^\star bb^\star c$.

(b) $L(G) = \{a^{3k} \mid k \text{ impair}\}$.

Question 9 (2 points)

Donner des grammaires pour les langages suivants :

(a) Les mots sur l'alphabet $\{a, b\}$ qui sont égaux à leur mot miroir.

(b) Les mots sur l'alphabet $\{a, b, c\}$ qui contiennent autant de a que de b.

Solution:

(a) $S \to aSa \mid bSb \mid a \mid b \mid \varepsilon$

(b) Autant de a que de b n'est pas difficile à réaliser, et ensuite on bricole pour ajouter des c un peu partout :

$$\begin{aligned} S &\to CaCSCbC \mid CbCSCaC \mid \varepsilon \mid C \\ C &\to cC \mid \varepsilon \end{aligned}$$

Une autre solution, en utilisant une grammaire contextuelle :

$$\begin{aligned} S &\to ABS \mid cS \mid \varepsilon \\ AB &\to BA \\ Ac &\to cA \\ A &\to a \\ Bc &\to cB \\ B &\to b \end{aligned}$$

Pour cette grammaire, on introduit autant de a que de b et un nombre arbitraire de c, puis on mélange les lettres pour obtenir le mot recherché.

Il resterait à justifier.

4 Indécidabilité : le Problème de Correspondance de Post.

Soit Σ un alphabet fini, et $P \subseteq \Sigma^\star \times \Sigma^\star$ un ensemble fini de dominos étiquetés par des mots sur l'alphabet Σ (des paires de mots, donc). Le Problème de Correspondance de Post (PCP), introduit par Emil Post en 1946, consiste à déterminer s'il existe une séquence de dominos de P tels que le mot obtenu par la concaténation des premières composantes est identique à celui formé par la concaténation des secondes composantes. Plus formellement, on cherche à déterminer l'existence d'une suite $(u_i, v_i)_{0 \leqslant i \leqslant n}$ telle que : $\forall 0 \leqslant i \leqslant n, (u_i, v_i) \in P$ et $u_0 \cdot u_1 \cdots u_n = v_0 \cdot v_1 \cdots v_n$.

Question 10 (2 points)

Résoudre PCP **"à la main"** pour les instances suivantes. *Si une correspondance existe, on la donnera explicitement. Sinon, on justifiera soigneusement qu'une telle correspondance n'existe pas.*

(a) $P_0 = \{(a, aaa), (aaaa, a)\}$;

(b) $P_1 = \{(aab, ab), (bab, ba), (aab, abab)\}$;

(c) $P_2 = \{(a, ab), (ba, aba), (b, aba), (bba, b)\}$;

(d) $P_3 = \{(ab, bb), (aa, ba), (ab, abb), (bb, bab)\}$.

Solution:
Les solutions qui suivent mélangent deux notations, la notation par paire $(gauche, droite)$ qui est celle de l'énoncé et la notation par domino $\boxed{\begin{array}{|c|}\hline haut \\ \hline bas \\ \hline \end{array}}$, qui est plus lisible.

(a) On a une correspondance, en faisant $\boxed{\begin{array}{|c|}\hline a \\ \hline aaa \\ \hline \end{array}}\boxed{\begin{array}{|c|}\hline a \\ \hline aaa \\ \hline \end{array}}\boxed{\begin{array}{|c|}\hline a \\ \hline aaa \\ \hline \end{array}}\boxed{\begin{array}{|c|}\hline aaaa \\ \hline a \\ \hline \end{array}}\boxed{\begin{array}{|c|}\hline aaaa \\ \hline a \\ \hline \end{array}}$, ce qui donne en haut comme en bas $aaaaaaaaaaa$.

(b) Une telle correspondance n'existe pas. En effet, une correspondance ne peut démarrer par les paires 1 ou 3 parce qu'alors le mot de gauche commence par aab et celui de droite par ab. Si elle commence par la deuxième paire, alors à gauche on a bab et à droite ba, du coup comme aucune des deux autres paires ne contient un mot qui commence par b comme deuxième composante, on est obligé de rajouter une occurence de la paire 2. Seules les paires 2 peuvent donc être enchaînées pour créer une correspondance. Comme les mots de la paire 2 ont des tailles différentes, on ne pourra jamais obtenir de correspondance.

(c) Il existe une correspondance : $\boxed{\begin{array}{|c|}\hline a \\ \hline ab \\ \hline \end{array}}\boxed{\begin{array}{|c|}\hline bba \\ \hline b \\ \hline \end{array}}\boxed{\begin{array}{|c|}\hline ba \\ \hline aba \\ \hline \end{array}}$. Cela donne des deux côtés $abbaba$.

(d) Impossible aussi, car les paires 3 et 4 font grossir le mot de droite (plus vite que celui de gauche), et on n'a aucune possibilité de rétablir la taille. De plus on ne peut démarrer une correspondance par 1 ou 2.

Question 11 (1 point)
Donner un algorithme pour décider PCP dans le cas où Σ ne contient qu'une seule lettre.

Solution:
Soit $\Sigma = \{a\}$, on peut remarquer que pour tout mot $w \in \Sigma^\star$, on a $w = a^{|w|}$. On peut montrer qu'il existe une correspondance dans P si et seulement si :

— soit on a $\boxed{\begin{array}{|c|}\hline u \\ \hline v \\ \hline \end{array}} \in P$ tel que $|u| = |v|$;

— soit il existe $\boxed{\begin{array}{|c|}\hline u_1 \\ \hline v_1 \\ \hline \end{array}}, \boxed{\begin{array}{|c|}\hline u_2 \\ \hline v_2 \\ \hline \end{array}} \in P$ tels que $|u_1| > |v_1|$ et $|u_2| < |v_2|$.

Si il existe $\boxed{\begin{array}{|c|}\hline u \\ \hline v \\ \hline \end{array}} \in P$ tel que $|u| = |v|$, la séquence contenant uniquement $\boxed{\begin{array}{|c|}\hline u \\ \hline v \\ \hline \end{array}}$ est une correspondance, puisque $u = a^{|u|} = a^{|v|} = v$. Si il existe $\boxed{\begin{array}{|c|}\hline u_1 \\ \hline v_1 \\ \hline \end{array}}, \boxed{\begin{array}{|c|}\hline u_2 \\ \hline v_2 \\ \hline \end{array}} \in P$ tels que $|u_1| > |v_1|$ et $|u_2| < |v_2|$, alors $|v_2| - |u_2| > 0$ et $|u_1| - |v_1| > 0$ et la séquence

$$\boxed{\begin{array}{|c|}\hline u_1 \\ \hline v_1 \\ \hline \end{array}}^{|v_2|-|u_2|}\boxed{\begin{array}{|c|}\hline u_2 \\ \hline v_2 \\ \hline \end{array}}^{|u_1|-|v_1|}$$

forme une correspondance car on obtient en haut

$$a^{|u_1|\times(|v_2|-|u_2|)} \cdot a^{|u_2|\times(|u_1|-|v_1|)} = a^{|u_1||v_2|+|u_1||u_2|-|u_1||u_2|-|u_2||v_1|} = a^{|u_1||v_2|-|u_2||v_1|}$$

qui est égal à $a^{|v_1|\times(|v_2|-|u_2|)} \cdot a^{|v_2|\times(|u_1|-|v_1|)}$, le mot qu'on obtient en bas.
Si en revanche on n'est pas dans cette situation, cela signifie que

— soit $\forall \boxed{\begin{array}{|c|}\hline u \\ \hline v \\ \hline \end{array}} \in P, |u| > |v|$;

— soit $\forall \boxed{\frac{u}{v}} \in P, |u| < |v|$.

Ces deux cas étant symétriques, on ne traitera ici que le premier. Si $\boxed{\frac{u}{v}} \in P, |u| > |v|$, alors pour toute séquence de dominos $\boxed{\frac{u_0}{v_0}} \cdots \boxed{\frac{u_n}{v_n}}$, on a

$$|u_0 \cdot u_1 \cdots u_n| = |u_0| + |u_1| + \cdots + |u_n| > |v_0| + |v_1| + \cdots + |v_n| = |v_0 \cdot v_1 \cdots v_n|.$$

On en déduit que

$$u_0 \cdot u_1 \cdots u_n = a^{|u_0 \cdot u_1 \cdots u_n|} \neq a^{|v_0 \cdot v_1 \cdots v_n|} = v_0 \cdot v_1 \cdots v_n.$$

Par conséquent tester si il existe une correspondance constituée de dominos de P revient à tester si il existe $\boxed{\frac{u_1}{v_1}}$ et $\boxed{\frac{u_2}{v_2}}$ dans P vérifiant $|u_1| \geqslant |v_1|$ et $|u_2| \leqslant |v_2|$, ce qui est décidable.

On considère maintenant une variante de PCP, le Problème de Correspondance de Post Modifié (PCPM). Pour ce problème, on considère un ensemble fini de dominos P et un domino initial $(u_0, v_0) \in P$, et on cherche à déterminer l'existence d'une correspondance qui commence par (u_0, v_0).

On considère comme acquis que PCPM est indécidable : en effet il existe une réduction du problème de l'arrêt d'une machine de Turing vers PCPM. On va maintenant montrer que PCP est indécidable, en réduisant depuis PCPM. On commence par introduire deux fonctions p et s. Soit \$ un nouveau symbole n'appartenant pas à Σ, pour tout mot $w = a_1 a_2 \cdots a_n$ (les a_i sont les lettres), on définit :

$$\begin{cases} p(w) = \$a_1\$a_2 \cdots \$a_n \\ s(w) = a_1\$a_2\$ \cdots a_n\$ \end{cases}$$

Question 12 (2 points)

Montrer que les fonctions p et s vérifient les propriétés suivantes :

(a) pour deux mots v et w, $p(vw) = p(v)p(w)$ et $s(vw) = s(v)s(w)$;

Solution:

Si $v = a_1 a_2 \cdots a_n$ et $w = b_1 b_2 \cdots b_m$, alors

$$\begin{aligned} p(vw) =& p(a_1 a_2 \cdots a_n b_1 b_2 \cdots b_m) \\ =& \$a_1\$a_2 \cdots \$a_n\$b_1\$b_2 \cdots \$b_m \\ =& (\$a_1\$a_2 \cdots \$a_n) \cdot (\$b_1\$b_2 \cdots \$b_m) = p(w) \cdot p(v) \\ s(vw) =& s(a_1 a_2 \cdots a_n b_1 b_2 \cdots b_m) \\ =& a_1\$a_2\$ \cdots a_n\$b_1\$b_2\$ \cdots b_m\$ \\ =& (a_1\$a_2\$ \cdots a_n\$) \cdot (b_1\$b_2\$ \cdots b_m\$) = s(w) \cdot s(v) \end{aligned}$$

(b) pour tout mot w, $p(w)\$ = \$s(w)$.

Solution:
$p(a_1a_2\cdots a_n)\$ = \$a_1\$a_2\cdots\$a_n\cdot\$ = \$\cdot a_1\$a_2\cdots\$a_n\$ = \$s(a_1a_2\cdots a_n)$

Soit $P, (u_0, v_0)$ une instance une PCPM, on définit $P' = \{(p(u_0), \$s(v_0))\} \cup P_1 \cup P_2$ avec :

$$\begin{cases} P_1 = \{(p(u), s(v)) \mid (u, v) \in P\} \\ P_2 = \{(p(u)\$, s(v)) \mid (u, v) \in P\} \end{cases}$$

Question 13 (2 points)
Montrer que $P, (u_0, v_0)$ admet une solution pour PCPM si et seulement si P' admet une solution pour PCP.

Solution:
Supposons d'abord que l'instance originale de PCPM a une solution $\boxed{\frac{u_0}{v_0}}\cdots\boxed{\frac{u_n}{v_n}}$, avec le haut et le bas égaux à un certain mot w :

$$u_0\cdot u_1\cdots u_n = v_0\cdot v_1\cdots v_n = w$$

Grâce à la question 12, on a :

$$\begin{aligned} p(w)\$ =& p(u_0)p(u_1)p(u_2)\cdots p(u_{n-1})p(u_n)\$ \\ =& \$s(w) \\ =& \$s(v_0)s(v_1)s(v_2)\cdots s(v_{n-1})s(v_n). \end{aligned}$$

Par définition de P', on a

— $\boxed{\frac{p(u_0)}{\$s(v_0)}} \in P'$;

— $\forall 1 \leqslant k < n, \boxed{\frac{p(u_k)}{s(u_k)}} \in P_1 \subseteq P'$;

— $\boxed{\frac{p(u_n)\$}{s(v_n)}} \in P_2 \subseteq P'$.

Par conséquent la séquence

$$\boxed{\frac{p(u_0)}{\$s(v_0)}}\boxed{\frac{p(u_1)}{s(v_1)}}\boxed{\frac{p(u_2)}{s(v_2)}}\cdots\boxed{\frac{p(u_{n-1})}{s(v_{n-1})}}\boxed{\frac{p(u_n)\$}{s(v_n)}}$$

est une correspondance de P'.

Si P' admet une solution, on peut remarquer qu'elle commence nécessairement par $\boxed{\frac{p(u_0)}{\$s(v_0)}}$ (c'est le seul domino pour lequel la partie supérieure et la partie inférieure commencent par la même lettre) et finit forcément par un domino de P_2, puisque ce sont les seuls dont les parties supérieure et inférieure finissent par la même lettre. Par conséquent toute solution de P' est de la forme

$$\boxed{\frac{p(u_0)}{\$s(v_0)}}\boxed{\frac{p(u_1)}{s(v_1)}}\cdots\boxed{\frac{p(u_{n-1})}{s(v_{n-1})}}\boxed{\frac{p(u_n)\$}{s(v_n)}}$$

où

$$\boxed{\frac{u_0}{v_0}}\boxed{\frac{u_1}{v_1}}\boxed{\frac{u_2}{v_2}}\cdots\boxed{\frac{u_{n-1}}{v_{n-1}}}\boxed{\frac{u_n}{v_n}}$$

est une solution de P.

Question 14 (1 point)
En déduire que PCP est indécidable.

Solution:
Notons $\rho(P, (u_0, v_0)) = P'$, défini comme précédement. ρ est une fonction calculable, et grâce à la question 13 on sait que :

$$(P, (u_0, v_0)) \in \text{PCPM} \Leftrightarrow \rho(P) \in \text{PCP}.$$

Par conséquent ρ est une *réduction* de PCPM vers PCP, et comme PCPM est indécidable, PCP l'est aussi.