Examen du 4 janvier 2006 : Corrigé

(durée: 3 heures)

Exercice 1 (récursivité). **1.** Soit A récursif infini, soit x_0 sont plus petit élément. On définit f par récurrence et minimisation :

$$f(0) = a$$
$$f(n+1) = \mu x.[x \in A \wedge x > f(n)]$$

La fonction$f$ est donc bien récursive et elle est totale car A est récursif, donc il suffit qu'il existe un x tel que $x \in A \wedge x > f(n)$ pour que la fonction soit définie en $n+1$, ce qui est le cas car A est infini. On a bien $A = \mathrm{Im}\, f$.

Réciproquement si f est récursive strictement croissante de $\mathbb{N} \to \mathbb{N}$, alors

$$x \in \mathrm{Im}\, f \text{ ssi } \exists y \leqslant x f(y) = x$$

et donc $\mathrm{Im}\, f$ est récursif.

2. On définit la fonction $g : \mathbb{N}^2 \to \mathbb{N}$:

$$g(i,0) = \varphi(i,0)$$
$$g(i,x+1) = (1 + \mu z.[(g(i,x) < \varphi(i,x+1)]) \cdot \varphi(i,x+1)$$

Cette fonction est définie en $x+1$ ssi $g(i,x)\downarrow$ et $\varphi(i,x+1)\downarrow$ et $g(i,x) < \varphi(i,x+1)$. Elle vaut alors $\varphi(i,x+1)$. Soit a un de ses indices. En posant $\alpha(i) = s^1_1(a,i)$ on a une fonction α telle que cherchée.

3. On définit une fonction récursive partielle h par composition et minimisation à l'aide du prédicat récursif T de terminaison de Kleene et de la fonction récursive U qui extrait le résultat du code du calcul.

$$h(i,x) = \mu c. \left[T[i, \pi^2_1(c), \pi^2_2(c)] \wedge U(\pi^2_2(c)) = x\right]$$

Montrons que la fonction $h(i,x)$ est définie en (i,x) si et seulement si $x \in \mathrm{Im}\,\varphi_i$. En effet le prédicat sur lequel se fait la minimisation est bien récursif (primitif récursif). Si $x \in \mathrm{Im}\,\varphi_i$, il existe d tel que $T[i,x,d]$ et donc $c = \langle x, d\rangle$ convient. Si $x \notin \mathrm{Im}\,\varphi_i$, on n'aura jamais $T[i, \pi^2_1(c), \pi^2_2(c)] \wedge U(\pi^2_2(c) = x$.

Soit b un indice de h, par le théorème smn on obtient une fonction $\beta(i) = s^1_1(b,i)$ qui convient.

4. Montrons que la fonction $\gamma = \beta \circ \alpha$ convient. Par définition de β il suffit de montrer que $\left\{\mathrm{Im}\,\varphi_{\alpha(i)} \;/\; i \in \mathbb{N}\right\}$ est la classe des ensembles récursifs.

Tout d'abord $\mathrm{Im}\,\varphi_{\alpha(i)}$ est récursif : soit $\varphi_{\alpha(i)} = \varphi_i$ et alors $\varphi_{\alpha(i)}$ est récursive totale strictement croissante donc d'ensemble image récursif, soit $\mathrm{Im}\,\varphi_{\alpha(i)}$ est fini donc également récursif. De plus $\left\{\mathrm{Im}\,\varphi_{\alpha(i)} \;/\; i \in \mathbb{N}\right\}$ contient naturellement tous les ensembles récursifs infinis (première question).

Il contient également tous les ensembles finis non vides. En effet, si A est un ensemble fini non vide de plus petit élément a, en définissant f comme à la question **1** :

$$f(0) = a$$
$$f(n+1) = \mu x.[x \in A \wedge x > f(n)]$$

on a bien f récursive partielle et qui énumère dans l'ordre les éléments de A, donc si $f = \varphi_i$, $f = \varphi_{\gamma(i)}$, et donc $A \in \left\{\mathrm{Im}\,\varphi_{\alpha(i)} \;/\; i \in \mathbb{N}\right\}$.

Enfin si i est l'indice de la fonction nulle part définie, on a encore $\varphi_{\alpha(i)} = \varphi_i$, et donc $\emptyset \in \left\{\mathrm{Im}\,\varphi_{\alpha(i)} \;/\; i \in \mathbb{N}\right\}$.

5. D'après le théorème de Rice-Shapiro, l'ensemble de tous les indices des sous-ensembles récursifs de $\mathbb{N}$, qui est extensionnel pour les ensembles, ne peut être récursivement énumérable, puisque la classe des ensembles récursifs n'est pas stable par sur-ensemble : un ensemble récursivement énumérable non récursif contient l'ensemble vide, qui est un ensemble récursif.

Exercice 2 (récursivité, réduction, hiérarchie arithmétique). **1.** On a

$$A_j = \{i \in \mathbb{N} \;/\; \forall x(\varphi(j,x)\downarrow \Rightarrow \varphi(i,x)\downarrow)\}\ .$$

Comme $\varphi(i,x)\downarrow$ est Σ_1, et $\varphi(j,x)\downarrow$ est Σ_1 en position négative, $(\varphi(j,x)\downarrow \Rightarrow \varphi(i,x)\downarrow)$ est Σ_1, et $\forall x(\varphi(j,x)\downarrow \Rightarrow \varphi(i,x)\downarrow)$ est donc Π_2. Il est possible que $W_j = \emptyset$ (j indice de la fonction nulle part définie). Dans ce cas $A_j = \mathbb{N}$, et donc A_j est Σ_0 et ne peut être Π_2-complet.

2. On suppose $W_j \neq \emptyset$. On a alors $A_j \neq \mathbb{N}$. On pose $\psi(i,x) = \varphi(i,i)$. Soit a un indice de ψ, $\psi(i,x) = \varphi^2(a,i,x) = \varphi^1(s^1_1(a,i),x)$. On pose $\alpha(i) = s^1_1(a,i)$.

Par construction de ψ, $W_{\alpha(i)} = \mathbb{N}$ si $i \in K$, $W_{\alpha(i)} = \emptyset$ sinon, donc $W_{\alpha(i)} \supset W_j$ si $i \in K$, $W_{\alpha(i)} \not\supset W_j$ sinon. On a bien $i \in K$ ssi $\alpha(i) \in A_j$, i.e. K se réduit à A_j.

3. Supposons W_j, fini non vide, posons $W_j = \{a_1, \ldots, a_n\}$. On a

$$A_j = \{i \in \mathbb{N} \,/\, \bigwedge_{k=1}^{n} (\psi(i, a_k) \downarrow)\}$$

or une conjonction de formules Σ_1 est Σ_1, donc A_j est Σ_1. Comme K est Σ_1-complet et se réduit à A_j, A_j est Σ_1-complet.

4. On pose $\sigma(i,x) = \sum_{k=0}^{x} \varphi(i,k)$. Soit b un indice de σ, $\sigma(i,x) = \varphi^2(b,i,x) = \varphi^1(s_1^1(b,i),x)$. On pose $\beta(i) = s_1^1(b,i)$. On a $\varphi_{\beta(i)} = \lambda x \sigma(i,x)$ est partout définie ssi φ_i est partout définie, et dès que φ_i n'est pas définie en a, $\varphi_{\beta(i)}$ n'est pas définie pour $i > a$.

5. Le domaine de définition de $\lambda x \sigma(i,x)$ est égal à $\mathbb{N}$ si φ_i est totale, fini sinon. On a donc $W_{\beta(i)} \supset W_j$ si φ_i est totale, et si de plus W_j est infini, alors $W_{\beta(i)} \not\supset W_j$ si φ_i n'est pas totale. On a donc bien, dans le cas où W_j est infini, $i \in C$ ssi $\beta(i) \in A_j$, i.e. C se réduit à A_j. Comme A_j est Π_2 et C Π_2-complet, si W_j est infini, A_j est Π_2-complet.

Exercice 3 (arithmétique). **1.** On sait que le prédicat T^1 de Kleene est primitif récursif, donc récursivement énumérable, donc représentable par une formule Σ_1 dans toute théorie arithmétique récursive Σ-cohérente.

2. Soit τ le code de la formule $\forall x_2 \exists x_3\ \mathcal{T}$. On sait que la fonction : $i \mapsto \underline{i}$ est primitive récursive, ainsi que la fonction de substitution d'un terme à une variable. la fonction $f : i \mapsto \forall x_2 \exists x_3\ \mathcal{T}[\underline{i}/x_1]$ est donc primitive récursive.

3. La théorie S étant récursive, on sait que la preuve se code de façon récursive : l'entier p code une preuve dans S de la formule de code a est un prédicat récursif en p et a, soit $Dem_S(p,a)$. On a :

$$\vdash_S \forall x_2 \exists x_3\ \mathcal{T}[\underline{i}/x_1] \text{ ssi } \exists p\ Dem_S(p, f(i))$$

par composition $Dem_S(p, f(i))$ est un prédicat récursif en p et i, donc par projection $\exists p\ Dem_S(p, f(i))$ est un prédicat récursivement énumérable : I est récursivement énumérable.

4. Soit $i \in I$. On a donc que $\vdash_S \forall x_2 \exists x_3\ \mathcal{T}[\underline{i}/x_1]$. En particulier pour tout entier n, $\vdash_S \exists x_3\ \mathcal{T}[\underline{i}/x_1, \underline{n}/x_2]$. Ces formules sont Σ et démontrables dans S. Comme S est Σ-cohérente, elles sont vraies dans $\mathbb{N}$, et donc on a pour tout entier n un entier d tel que $\mathbb{N} \models \mathcal{T}[\underline{i}/x_1, \underline{n}/x_2, \underline{d}/x_3]$. Comme $\mathcal{T}$ est Σ, cela signifie bien que pour tout entier n il existe un entier d tel que $T^1(i,n,d)$, c'est à dire que φ_i est totale.

5. Si I est vide , comme il existe bien des machines qui terminent sur chaque entrée (toutes celles qui calculent des fonctions récursives totales), on a le résultat.

Si I est non vide, alors I est l'image d'une fonction récursive totale, soit α, et on peut procéder par diagonalisation : la fonction $h : \mathbb{N} \to \mathbb{N}$ qui à m associe $\varphi(\alpha(m), m) + 1$ est récursive totale car comme $\alpha(m) \in I$, $\varphi_{\alpha(m)}$ est totale. Cette fonction diffère de $\varphi_{\alpha(m)}$ en m. Aucun de ses indices, donc des indices des machines qui la calculent, n'est donc dans $I = \operatorname{Im} \alpha$.

6. Soit e un indice de la fontion h de la question précédente. Comme $e \notin I$, $\not\vdash_S \forall x_2 \exists x_3\ \mathcal{T}[\underline{e}/x_1]$. Comme h est totale, $\mathbb{N} \models \forall x_2 \exists x_3\ \mathcal{T}[\underline{e}/x_1]$. Comme l'énoncé $\mathcal{T}$ est Σ_1, l'énoncé précédent est Π_2.

7. Il suffit de poser $A = \exists x_3\ \mathcal{T}[\underline{e}/x_1]$. D'après la question précédente, on a bien $\not\vdash_S \forall x_2\ A$, et de plus $\mathbb{N} \models \forall x_2\ A$. On a donc pour tout entier n, $\mathbb{N} \models A[\underline{n}/x_2]$, cette formule étant Σ, et S étant arithmétique, $\vdash_S A[\underline{n}/x_2]$.

Exercice 4 (complexité). **1.** On suppose d'abord qu'il existe $x_1, \ldots, x_n$ tels que $\sum_{i=1}^{n} c_i x_i = K$. Soit $m = \sum_{i=1}^{n} x_i$.

On considère la suite de longueur m définie de la manière suivante :

$(\delta_1, \ldots, \delta_m) = (\underbrace{c_1, \ldots, c_1}_{x_1 \text{ fois}}, \underbrace{c_2, \ldots, c_2}_{x_2 \text{ fois}}, \ldots, \underbrace{c_n, \ldots, c_n}_{x_n \text{ fois}})$

La suite de sommets suivante est un chemin de 0 à K dans le graphe G :

$i_0 = 0$ et $i_j = \sum_{i=1}^{j} \delta_i$ $(j = 1, \ldots, m)$.

Inversement, on suppose que $i_0 = 0, i_1, \ldots, i_m = K$ est un chemin de 0 à K dans le graphe G. On pose : $\delta_j = i_j - i_{j-1}$ $(j = 1, \ldots, m)$. Il est clair que : $\sum_{j=1}^{m} \delta_j = K$. D'après la définition du graphe G, $\delta_1, \ldots, \delta_m$ prennent leur valeur parmi $c_1, \ldots, c_n$. En prenant x_i égal au nombre de fois où c_i apparaît dans $\delta_1, \ldots, \delta_m$, on obtient : $\sum_{i=1}^{n} c_i x_i = K$.

2. Le nombre de sommets du graphe G étant égal à $K + 1$ et son nombre d'arêtes égal à nK, la construction du graphe peut s'effectuer en temps $O(nK)$, en considérant l'addition comme une opération élémentaire (le coût de l'addition sur les entiers $\leqslant K$ est en fait en O(log K)).

La recherche d'un chemin entre deux sommets dans un graphe (orienté) peut s'effectuer à l'aide d'un algorithme de *marquage* dont la complexité en temps est $O(nK)$ (le nombre d'arêtes étant nK).

3. La taille d'une instance du problème du SAC A DOS est de l'ordre de $n(log\ K)$. L'algorithme précédent, de complexité en temps $O(nK)$, n'est donc pas un algorithme en temps polynomial pour ce problème, qui est par ailleurs $\mathcal{NP}$-complet.