Fondements de l'informatique. Examen
Durée: 3h

Sujet proposé par Olivier Bournez
Version 11
(corrigé)

Les 4 parties sont indépendantes, et peuvent être traitées dans un ordre quelconque. On pourra admettre le résultat d'une question pour passer aux questions suivantes. On pourra utiliser tous les résultats et les théorèmes démontrés ou énoncés en cours ou en petite classe ou dans le polycopié sans chercher à les redémontrer. Il est possible d'avoir la note maximale sans répondre à toutes les questions. La difficulté des questions n'est pas une fonction linéaire ni croissante de leur numérotation. La qualité de votre argumentation et de votre rédaction est une partie importante de votre évaluation.

Les paragraphes qui commencent par "Commentaire :" correspondent à des discussions sur les résultats obtenus ou à venir, et peuvent être ignorés. On notera dans cet énoncé la multiplication par le symbole $\cdot$, et par $\mathbb{N}$ l'ensemble des entiers.

1 Machines de Turing

Question 1. *Le problème suivant est-il décidable ? Déterminer si le langage accepté par une machine de Turing M ne contient que des mots de longueur divisible par* 3*.*

Solution : C'est indécidable : il s'agit d'une application directe du théorème de Rice. □

Question 2. *On fixe $c \in \mathbb{N}$. Un langage L est dit* reconnaissable en espace $c \cdot n$ *s'il existe une machine de Turing M qui accepte L et telle que sur tout mot w, M utilise (= écrit) au plus $c \cdot n$ cases de son ruban, où n est la longueur de w.*

Démontrer qu'un langage reconnaissable en espace $c \cdot n$ est décidable.

Solution : L est nécessairement décidable : en effet, il est décidé par une machine de Turing qui simule M en s'arrêtant dès que M essaye d'accéder à une case du ruban à distance plus que $c \cdot n$ fois de la position initiale (dans ce cas on refuse) ou dès que M passe deux fois par la même configuration (dans ce cas on refuse), et en acceptant dès que M accepte : une machine qui utilise uniquement ces $2 \cdot c \cdot n$ cases possède un nombre fini de configurations possibles ; si elle passe deux fois par la même configuration, elle le fera indéfiniment ; sinon, elle ne peut avoir qu'au plus ce nombre de configurations d'étapes avant de s'arrêter. □

2 L'idée d'une startup innovante

Un camarade souhaite créer un nouveau réseau social JeNAimePasEtreContredit. L'originalité de ce réseau social est de garantir à ses utilisateurs d'être mis en correspondance uniquement avec des utilisateurs avec lesquels ils ont déclaré être d'accord. On note $D(x, y)$ pour le fait que

x et y ont déclaré être d'accord lorsqu'ils se sont inscrits [1]. On suppose que D est symétrique et réflexif : $D(x, y)$ implique $D(y, x)$ pour tout x, y, et $D(x, x)$ pour tout x.

Il vous embauche comme informaticien et vous demande de produire un algorithme qui prend en entrée une liste d'utilisateurs $L = \{x_1, x_2, \ldots, x_n\}$, et la liste des couples (i, j) avec $D(x_i, x_j)$, $1 \leq i \leq n$, $1 \leq j \leq n$, ainsi qu'un entier N, et qui produit en sortie un sous-ensemble S constitué de N utilisateurs tel que $D(x, y)$ pour tout $x, y \in S$, si un tel groupe existe, et qui sinon retourne qu'il n'en existe pas.

Question 3. *On suppose que $D(x, y)$ est de plus transitif : $D(x, y)$ et $D(y, z)$ impliquent $D(x, z)$ pour tout x, y, z. Pouvez-vous résoudre son problème en temps polynomial ?*

Solution : Il suffit de parcourir les utilisateurs x, et pour chacun de marquer la classe d'équivalence correspondante : c'est-à-dire de marquer tous les autres y avec $D(x, y)$. Par exemple, en parcourant tous les y et en les marquant si $D(x, y)$ apparait (puisque $D(x, y)$ est transitif, y est dans la classe d'équivalence de x si et seulement si y apparaît). Si l'on trouve une classe d'équivalence avec plus que N sommets on la retourne. Sinon, on renvoie qu'il n'y a pas de tel groupe. Tout cela se fait en temps polynomial. □

On ne fait plus l'hypothèse dorénavant que $D(x, y)$ est transitif.

Question 4. *Pouvez-vous résoudre son problème en temps polynomial ?*

Solution : Ce n'est pas possible de le faire en temps polynomial, sauf si $P = NP$ car c'est le problème CLIQUE qui est NP-complet. Rappelons l'énoncé :
Donnée : Un graphe $G = (V, E)$ non-orienté et un entier N
Réponse : Décider si un graphe contient un clique de taille N.

Ici, le réseau social est le graphe tel que les sommets sont les utilisateurs et les arêtes représentent la relation binaire D.

□

Question 5. *Il vous demande de produire un algorithme qui prend en entrée une liste d'utilisateurs et une liste de couples comme ci-dessus et qui produit un sous-ensemble S avec simultanément :*

1. *S contient au moins 10% des utilisateurs de L*
2. *$D(x, y)$ pour tout $x, y \in S$*

si un tel groupe existe, et qui sinon retourne qu'il n'en existe pas. Pouvez-vous résoudre ce nouveau problème en temps polynomial ?

Solution : Ce n'est pas possible de le faire en temps polynomial, sauf si $P = NP$ car ce problème est aussi NP-complet. En effet, d'une part le problème est clairement dans NP : la donnée d'un sous-groupe S est un certificat vérifiable en temps polynomial.

Démontrons en effet que le problème CLIQUE se réduit à ce problème.

On considère la fonction suivante f qui envoie une instance $(G = (V, E), N)$ de CLIQUE (la question précédente) en une instance $G' = (V', E')$ de ce problème. On note n le nombre de sommets du graphe G et n' le nombre de sommets de G'. On construit f pour que $G' = f(G)$ contienne toujours le graphe G, et possède α nouveaux sommets. Le choix de α et les arêtes que l'on ajoute dépendent de si l'on a $N < \frac{n}{10}$ ou le contraire.

Pour $N < \frac{n}{10}$, on prend les α nouveaux sommets comme voisins à tous les autres sommets du graphe G, et on prend α de telle sorte que l'on aie environ $N + \alpha = \frac{n+\alpha}{10} = \frac{n'}{10}$. Très précisément, on prend $\alpha = \lceil \frac{n-10 \cdot N}{9} \rceil$, ce qui garantit que $n - 10 \cdot N = 9\alpha - k$ pour $k \in \{0, 1, \ldots, 8\}$, et donc $N + \alpha = \frac{n+\alpha+k}{10}$, soit $N + \alpha = \lceil \frac{n+\alpha}{10} \rceil = \lceil \frac{n'}{10} \rceil$.

1. On suppose que tous les utilisateurs se connaissent.

Sinon, on a $N \geq \frac{n}{10}$. On prend les α nouveaux sommets comme voisins d'aucun sommet, et on prend $\alpha = 10 \cdot N - n$, de telle sorte que $N = \frac{n+\alpha}{10} = \frac{n'}{10}$.

La fonction f ci-dessus se calcule bien en temps polynomial. Il reste à vérifier que G possède une clique avec exactement N sommets si et seulement si $G' = f(G)$ possède une clique avec au moins $\frac{n'}{10}$ sommets.

Supposons que G possède une clique S avec exactement N sommets. Pour $N < \frac{n}{10}$: S union les α nouveau sommets est aussi une clique dans G' : elle est de taille $N + \alpha$ et donc au moins $\frac{n'}{10}$. Pour $N \geq \frac{n}{10}$: S est une clique dans G' : elle est de taille $N = \frac{n'}{10}$. Donc, dans tous les cas G' possède une clique avec au moins $\frac{n'}{10}$ sommets.

Réciproquement, supposons que G' possède une clique S' avec au moins $\frac{n'}{10}$ sommets.

Pour $N < \frac{n}{10}$: puisque $N + \alpha = \lceil \frac{n'}{10} \rceil$, le nombre de sommets β dans la clique S' doit satisfaire $\beta \geq N + \alpha$. Par conséquent, la clique S' contient au moins N sommets de G, et donc il existe une clique $S = S'$ d'au moins N sommets dans G : en prenant n'importe quel sous-ensemble de taille N de S on obtient que G possède une clique de taille exactement N.

Pour $N \geq \frac{n}{10}$: comme tous les nouveaux sommets sont voisin d'aucun autre sommet, ils ne peuvent pas appartenir à une clique de plus de 1 élément. [2] Par conséquent, S' forme nécessairement une clique de G avec au moins $N = \frac{n'}{10}$ sommets : en prenant n'importe quel sous-ensemble de taille N de S' on obtient que G possède une clique de taille exactement N. □

3 Théorème de Cobham

On écrit $||x|| = \lceil \log_2(x+1) \rceil$ pour représenter la taille de l'écriture en binaire de l'entier x. Par exemple, $||23|| = 5$ car $23 = 1 \cdot 2^4 + 0 \cdot 2^3 + 1 \cdot 2^2 + 1 \cdot 2^1 + 1 \cdot 2^0$ s'écrit en binaire 10111 avec 5 lettres. Pour un k-uplet d'entiers $\vec{x} = (x_1, x_2, \ldots, x_k)$, on écrit $||\vec{x}||$ pour $||x_1|| + \cdots + ||x_n||$.

Commentaire : On pourra librement utiliser le fait que l'on a toujours $x < 2^{||x||} \leq 2 \cdot x + 1$ pour tout entier x dans toute la suite.

On utilisera les fonctions suivantes de $\mathbb{N}$ dans $\mathbb{N}$: $\mathsf{s}(x) = x + 1$, $\mathsf{double}_0(x) = 2 \cdot x$ et $\mathsf{double}_1(x) = 2 \cdot x + 1$ ainsi que la fonction $\sharp(x, y)$ de $\mathbb{N}^2$ dans $\mathbb{N}$ définie par $\sharp(x, y) = 2^{||x|| \cdot ||y||} - 1$.

Commentaire : La fonction $\sharp$ vérifie la propriété $||\sharp(x, y)|| = ||x|| \cdot ||y||$ pour tout x et y.

Toutes les fonctions sur les entiers que l'on considère dans cette partie sont totales : c'est-à-dire, définies pour toute valeur de leurs arguments.

3.1 FP et Fonctions primitives récursives

Une fonction $f : \mathbb{N}^n \to \mathbb{N}$ est *calculable en temps polynomial* s'il existe une machine de Turing M_f qui, produit $f(x_1, \ldots, x_n)$ (écrit en binaire) à partir de x_1, x_2, ...et x_n (écrits en binaire) en un nombre d'étapes polynomial en $||\vec{x}||$ où $\vec{x} = (x_1, \ldots, x_n)$. On note FP pour la classe des fonctions calculables en temps polynomial.

Question 6. *Démontrer qu'une fonction de* FP *est nécessairement de taille polynomiale : c'est-à-dire qu'il existe un polynome p tel que $||f(\vec{x})|| \leq p(||\vec{x}||)$ pour tout $\vec{x}$.*

Solution : En un temps T , une machine de Turing ne peut pas écrire plus que T cases de son ruban. Donc la taille du résultat ne peut pas être plus grande qu'un polynôme en la taille de l'entrée après un temps polynomial. □

Question 7. *Montrer que la fonction qui à x associe 2^x n'est pas calculable en temps polynomial.*

2. Pour les plus puristes, on observera que la preuve dans le polycopié de la NP-complétude du problème CLIQUE montre que ce dernier reste NP-complet en supposant $N \geq 2$: par exemple en ajoutant des clauses "inutiles" à la formule $3SAT$ dans la preuve de $3SAT \leq STABLE$.

Solution : Cette fonction n'est pas de taille polynomiale : en effet, pour $x = 2^n - 1$, $||x|| = n$, $||2^x|| = \lceil \log_2(2^{2^n-1} + 1) \rceil \geq \lceil \log_2(2^{2^n-1}) \rceil = 2^n - 1 \geq 2^{n-1}$ pour n assez grand, ce qui croit plus vite que tout polynôme en n pour n assez grand. □

Commentaire : La fonction qui à x associe 2^x est primitive récursive (cf PC1, question 1.2). Les fonctions primitives récursives et les fonctions de FP *sont donc deux classes de fonctions distinctes.*

Commentaire : Une idée dûe à Cobham pour s'approcher de FP *est de remplacer la fonction* $\mathsf{s}(x) = x + 1$ *qui sert de fondement dans les définitions par récurence dans le schéma* Rec *des fonctions primitives récursives par les fonctions* $\mathsf{double}_0(x)$ *et* $\mathsf{double}_1(x)$*. On va d'abord voir que cela ne suffit pas (question 9). On va ensuite voir que se restreindre aux récurrences bornées permet de garantir que l'on obtient que des fonctions de* FP *(question 10). On verra ensuite que cela donne même une caractérisation de* FP *(théorème de Cobham).*

3.2 Récurrence et taille

Question 8. *On suppose que f est définie par une récurrence du type*

$$\begin{cases} f(0, x_2, \ldots, x_n) = g(x_2, \ldots, x_n), \\ f(\mathsf{double}_0(x_1), x_2, \ldots, x_n) = h_0(f(x_1, \ldots, x_n), x_1, \ldots, x_n) \text{ pour } x_1 \neq 0 \\ f(\mathsf{double}_1(x_1), x_2, \ldots, x_n) = h_1(f(x_1, \ldots, x_n), x_1, \ldots, x_n), \end{cases}$$

pour tout entiers $x_1, x_2, \ldots, x_n$, avec les fonctions g, h_0, et h_1 calculables en temps polynomial.

Démontrer que si f est de taille polynomiale alors f se calcule en temps polynomial.

Solution : Sur une entrée $\vec{x} = (x_1, x_2, \ldots, x_n)$, il suffit de "dérouler" la définition, ce qui fera $||x_1||$ itérations, chaque itération se faisant en temps polynomial en $||x||$. Tous les valeurs intermédiaires dont on a besoin restent bien de taille polynomiale en $||x||$. □

Question 9. *Donner un exemple de fonction f définie par une récurence du type*

$$\begin{cases} f(0) = 1, \\ f(\mathsf{double}_0(x_1)) = h_0(f(x_1)) \text{ pour } x_1 \neq 0 \\ f(\mathsf{double}_1(x_1)) = h_1(f(x_1)) \end{cases} \tag{1}$$

où h_0 et h_1 sont des fonctions calculables en temps polynomial, mais où f n'est pas de taille polynomiale.

Solution : On considère la fonction $x \mapsto 2^x$ de la question 7 : on a $2^0 = 1$, $2^{\mathsf{double}_0(x)} = (2^x)^2$, et $2^{\mathsf{double}_1(x)} = 2 \cdot (2^x)^2$. On pose donc $h_0(x) = x^2$, et $h_1(x) = 2 \cdot x^2$. h_0 et h_1 se calculent bien en temps polynomial. □

3.3 Récurrence bornée

Définition. On dit qu'une fonction f est définie par *récurrence bornée* à partir des fonctions g, h_0, h_1 et m si

$$\begin{cases} f(0, x_2, \ldots, x_n) = g(x_2, \ldots, x_n), \\ f(\mathsf{double}_0(x_1), x_2, \ldots, x_n) = h_0(f(x_1, \ldots, x_n), x_1, \ldots, x_n) \text{ pour } x_1 \neq 0 \\ f(\mathsf{double}_1(x_1), x_2, \ldots, x_n) = h_1(f(x_1, \ldots, x_n), x_1, \ldots, x_n), \end{cases}$$

et si de plus

$$f(x_1, x_2, \ldots, x_n) \leq m(x_1, \ldots, x_n),$$

pour tout $x_1, x_2, \ldots, x_n$.

Définition. Une fonction $f : \mathbb{N}^n \to \mathbb{N}$ est *dans la classe de Cobham* si elle est l'une des fonctions :

— la constante 0 (alors $n = 0$)
— $\mathsf{double}_0 : x \mapsto 2 \cdot x$ et $\mathsf{double}_1 : x \mapsto 2 \cdot x + 1$ (alors $n = 1$) ;
— $\sharp : (x, y) \mapsto 2^{||x|| \cdot ||y||} - 1$ (alors $n = 2$) ;
— $\mathsf{Proj}_n^i : (x_1, \ldots, x_n) \mapsto x_i$ les fonctions de projection, pour $1 \leq i \leq n$;
— $\mathsf{Comp}_n(g, h_1, \ldots, h_m) : (x_1, \ldots, x_n) \mapsto g(h_1(x_1, \ldots, x_n), \ldots, h_m(x_1, \ldots, x_n))$ la composition des fonctions dans la classe de Cobham $g, h_1, \ldots, h_m$ (pour $m \geq 0$) ;
— $\mathsf{BoundedRecNotation}(g, h_0, h_1, m)$ une fonction f définie par *récurrence bornée* à partir des fonctions g, h_0, h_1 et m, avec g, h_0, h_1 et m dans la classe de Cobham.

Par exemple la fonction successeur $\mathsf{s}(x) = x + 1$ est dans la classe de Cobham car elle se définit par récurrence bornée [3]

$$\begin{cases} \mathsf{s}(0) = \mathsf{double}_1(0) \\ \mathsf{s}(\mathsf{double}_0(x)) = \mathsf{double}_1(x) \text{ pour } x \neq 0 \\ \mathsf{s}(\mathsf{double}_1(x)) = \mathsf{double}_0(\mathsf{s}(x)) \\ \mathsf{s}(x) \leq \mathsf{double}_1(x) \end{cases}$$

Deuxième exemple : la fonction taille $||x||$ est dans la classe de Cobham car $||0|| = 0$, $||\mathsf{double}_0(x)|| = \mathsf{s}(||x||)$ pour $x \neq 0$, $||\mathsf{double}_1(x)|| = \mathsf{s}(||x||)$, $||x|| \leq Proj_1^1(x) = x$.

Troisième exemple, la fonction $\mathsf{concat}(\mathsf{y}, \mathsf{x}) = 2^{||y||} \cdot x + y$ qui retourne un entier dont l'écriture en binaire est la concaténation de l'écriture de x et de y en binaire : elle est dans la classe de Cobham car $\mathsf{concat}(0, x) = x$, $\mathsf{concat}(\mathsf{double}_0(y), x) = \mathsf{double}_0(\mathsf{concat}(y, x))$, $\mathsf{concat}(\mathsf{double}_1(y), x) = \mathsf{double}_1(\mathsf{concat}(y, x))$, $\mathsf{concat}(y, x) \leq \sharp(||x||, ||y||)$.

Question 10. *Démontrer que toute fonction de la classe de Cobham se calcule en temps polynomial.*

Solution : Cela se fait par induction sur la définition. Les fonctions s, double_0 et double_1 sont dans FP. Les fonctions projections aussi. La fonction $\sharp$ est dans FP : il suffit d'écrire en binaire un 1 suivi de 0 avec un nombre de 0 donné par le produit de la longueur de x et de de la longueur de y moins 1. Tout cela se calcule en temps polynomial.

Les fonctions de FP sont stables par composition (composer les machines de Turing).

Supposons que f est définie par récurrence bornée à partir de g, h_0, h_1 et m dans FP. Par hypothèse d'induction, toutes ces fonctions sont dans FP. Or la taille de $f(x_1, \ldots, x_n)$ est bornée par $||m(x_1, \ldots, x_n)||$ qui est polynomiale : la fonction m étant dans FP sa taille est polynomiale par la question 6. Cela implique que f est dans FP par la question 8. □

Question 11. *Montrer que les fonctions suivantes sont dans la classe de Cobham :*

1. *la fonction* $\mathsf{mod}_2(x)$ *qui donne le reste de la division de* x *par* 2,
2. *la fonction* $\mathsf{quo}_2(\mathsf{x})$ *qui donne le quotient de la division de* x *par* 2,
3. *la fonction* $\mathsf{cond}(\mathsf{x}, \mathsf{y})$ *qui donne* 0 *pour* $x = 0$ *et* y *sinon.*

Solution : $\mathsf{mod}_2(0) = 0$, $\mathsf{mod}_2(\mathsf{double}_0(x)) = 0$, $\mathsf{mod}_2(\mathsf{double}_1(x)) = 1$, $\mathsf{mod}_2(\mathsf{x}) \leq \mathsf{Proj}_1^1(x) = x$.
$\mathsf{quo}_2(0) = 0$, $\mathsf{quo}_2(\mathsf{double}_0(x)) = x$ pour $x \neq 0$, $\mathsf{quo}_2(\mathsf{double}_1(x)) = x$, $\mathsf{quo}_2(\mathsf{x}) \leq \mathsf{Proj}_1^1(x) = x$.
$\mathsf{cond}(0, y) = 0$, $\mathsf{cond}_0(\mathsf{double}_0(x), y) = \mathsf{cond}_0(x, y)$ pour $x \neq 0$, $\mathsf{cond}_0(\mathsf{double}_1(x), y) = y$, $\mathsf{cond}_0(x, y) \leq \mathsf{Proj}_2^2(x, y) = y$.

□

3. Une définition plus formelle serait de dire que $\mathsf{s}(x)$ est la fonction

$$BoundedRecNotation(Comp_0(\mathsf{double}_1, 0), Comp_2(\mathsf{double}_1, Proj_2^2), Comp_2(\mathsf{double}_0, Proj_2^1), \mathsf{double}_1)$$

mais on ne demandera pas des écritures aussi formelles.

Question 12. *On considère le polynôme $p(n) = c{\cdot}n^h$ pour deux entiers c et h. Montrer qu'il existe une fonction $T(x)$ de $\mathbb{N}$ dans $\mathbb{N}$ dans la classe de Cobham telle que pour tout x, $2^{p(||x||)} \leq T(x)$. Indice : on pourra chercher à construire une fonction M avec $p(||x||) \leq ||M(x)||$ pour tout x.*

Solution : Etant donné un entier d, on peut construire une fonction f_d qui à x asocie un mot $f_d(x)$ de taille $d \cdot ||x||$: par exemple en prenant $f_d(x)$ qui est la concaténation d fois de x (par exemple à l'aide de la fonction concat). On peut alors définir $M(x)$ par $f_d(x)\sharp f_d(x)\sharp \ldots \sharp f_d(x)$ où f_d est répété h fois pour une constante entière $d \geq \sqrt[h]{c}$, et où on note $x\sharp y$ pour $\sharp(x, y)$ et utilise l'associativité de cette opération. Par construction, $||M(x)||$ est de taille $(d \cdot ||x||)^h$ donc plus grande que $c \cdot ||x||^h = p(||x||)$. On pose alors $T(x) = 2 \cdot M(2^{||x||}) + 1$ en utilisant le fait que $2^{||M(x)||} \leq 2 \cdot M(x) + 1$. La fonction T est bien dans la classe de Cobham, car elle s'écrit $\mathsf{double}_1(M(\mathsf{g}(x)))$ avec $\mathsf{g}(x) = 2^{||x||}$ donnée par exemple par $\mathsf{g}(0) = 1$, $\mathsf{g}(\mathsf{double}_0(x)) = \mathsf{double}_0(\mathsf{g}(x))$ pour $x \neq 0$, $\mathsf{g}(\mathsf{double}_1(x)) = \mathsf{double}_0(\mathsf{g}(x))$ et $\mathsf{g}(x) \leq f_2(x)$. □

On admettra qu'il existe une fonction dans la classe de Cobham $(x, y, z) \mapsto \langle x, y, z\rangle$ qui envoie bijectivement $\mathbb{N}^3$ sur $\mathbb{N}$ ainsi que trois fonctions $\pi_1, \pi_2, \pi_3 : \mathbb{N} \to \mathbb{N}$ dans la classe de Cobham telles que $\pi_i(\langle x_1, x_2, x_3\rangle) = x_i$ pour $i = 1, 2, 3$.

Une configuration d'une machine de Turing peut se coder par un entier. Celui-ci code un triplet : un entier qui marque le numéro de l'état ; un entier qui code la partie du ruban à gauche de la tête de lecture jusqu'au premier[4] B ; un entier qui code la partie du ruban à droite de la tête de lecture jusqu'au premier B, ses bits de poids faible étant les plus proches de la tête de lecture. Par exemple, le ruban $(B, 1, 1, \underline{0}, 0, 0, 1, B)$ avec la machine dans l'état numéro 5 et la tête de lecture au niveau du caractère souligné peut se coder par $\langle 5, 3, 8\rangle$ car $3 = 1 \cdot 2^1 + 1$ et $8 = 0 \cdot 2^0 + 0 \cdot 2^1 + 0 \cdot 2^2 + 1 \cdot 2^3$. Observer l'ordre des puissances de 2 dans ces écritures.

On fixe une machine de Turing M qui calcule une fonction de FP de $\mathbb{N}$ dans $\mathbb{N}$.

Question 13. *Exprimer par une fonction de la classe de Cobham le déplacement de la tête de lecture vers la droite, sans changer l'état de la machine et le caractère lu.*

Solution : La configuration est donnée par un certain $\langle a, b, c\rangle$. Lors d'un mouvement vers la droite, c est divisé par 2, tandis que b est multiplié par 2 et le bit final de c devient son dernier bit. Ceci s'écrit

$$\mathsf{RightMove}(x) = \langle \pi_1(x), \mathsf{Cond}(\mathsf{mod}_2(\pi_3(x)), \mathsf{double}_0(\pi_2(x)), \mathsf{double}_1(\pi_2(x))), \mathsf{quo}_2(\pi_3(x)))\rangle.$$

□

À ce stade on admet qu'on peut écrire une fonction $\mathsf{Next}_M(\vec{x})$ dans la classe de Cobham qui code la fonction de transition d'une machine de Turing M donnée : si $\vec{x}$ code une configuration, alors $\mathsf{Next}_M(\vec{x})$ code la configuration suivante.

Question 14. *On rappelle que M calcule une fonction de FP de $\mathbb{N}$ dans $\mathbb{N}$. Montrer qu'il existe une fonction $\mathsf{Run}_M(y, x)$ dans la classe de Cobham qui retourne la configuration de la machine après $||y||$ étapes sur l'entrée x.*

Solution : On peut supposer sans perte de généralité que l'état initial q_0 de la machine M porte le numéro 0.

Le ruban de la machine M ne peut pas s'être déplacé de plus de $t = ||y||$ cases au temps t. Par ailleurs, comme la machine M calcule une fonction de FP, elle effectue un nombre d'étapes t au plus polynomial en $||x||$. D'autre part, puisque $\langle \ldots, \ldots, \ldots\rangle$ est dans la classe de Cobham, la taille de $\langle a, b, c\rangle$ reste de taille polynomiale en celle du triplet (a, b, c). On en déduit qu'il existe un polynôme p tel que $||\mathsf{Run}_M(y, x)|| \leq p(||x|| + ||y||)$ pour tout x et y. Par conséquent,

4. B est le symbole de blanc.

$\mathsf{Run}_M(y,x) \leq 2^{p(||x||+||y||)} = 2^{p(||\mathsf{concat}(x,y)||)} \leq T(\mathsf{concat}(x,y))$ où T est la fonction dans la classe de Cobham de la question 12 pour le polynôme p.

La fonction Run_M est bien dans la classe de Cobham car elle s'écrit :

$$\begin{cases} \mathsf{Run}_M(0,x) = \langle 0,0,x\rangle \\ \mathsf{Run}_M(\mathsf{double}_0(y),x) = \mathsf{Next}(\mathsf{Run}_M(y,x)) \qquad \text{pour } y \neq 0, \\ \mathsf{Run}_M(\mathsf{double}_1(y),x) = \mathsf{Next}(\mathsf{Run}_M(y,x)), \\ \mathsf{Run}_M(y,x) \leq T(\mathsf{concat}(x,y)) \end{cases}$$

□

En déduire le théorème de Cobham :

Question 15. *Une fonction est dans classe de Cobham si et seulement si elle appartient à* FP.

Solution : Le fait qu'une fonction dans la classe de Cobham est dans FP est la question 10. Réciproquement, si une fonction f de $\mathbb{N}$ dans $\mathbb{N}$ est calculable en temps polynomial $c \cdot n^k$, alors f s'écrit $\pi_3(\mathsf{Run}_M(M(x),x))$ où la fonction M est celle de la solution de la question 12, vérifiant $c \cdot ||x||^k \leq ||M(x)||$.

Par ailleurs, tout ce raisonnement se généralise aux fonctions de $\mathbb{N}^k$ dans $\mathbb{N}$ sans difficulté : considérer des machines avec k rubans par exemple (un ruban par argument). □

4 Complétude de la théorie des corps algébriquement clos & Théorème d'Ax

Dans tout cet exercice, on ne considère que des formules et théories du premier ordre (c'est-à-dire les formules considérées dans le cours et le polycopié). On suppose que les signatures sont dénombrables.

4.1 Complétude d'une théorie

Une théorie consistante T sur une signature Σ est dite *complète* si pour toute formule close ϕ sur la signature Σ, on a $T \vdash \phi$ ou $T \vdash \neg\phi$.

Question 16. *Soit T une théorie complète, dont les axiomes sont récursivement énumérables. Montrer qu'elle est décidable : il y a un algorithme qui prend en entrée une formule close ϕ et qui détermine si $T \vdash \phi$.*

Solution : Par la complétude de T, étant donnée ϕ soit il y a une preuve de ϕ soit il y a une preuve de $\neg\phi$ à partir de T. On peut énumérer systématiquement toutes les formules prouvables à partir des axiomes de T jusqu'à tomber soit sur la preuve de ϕ soit sur la preuve de $\neg\phi$. On répond vrai dans le premier cas, et faux dans le second. Cette machine ne boucle pas car l'un des deux cas doit se produire. □

Deux structures M et N sur une signature Σ sont *isomorphes* s'il existe une bijection entre l'ensemble de base de M et de N qui préserve l'interprétation des symboles de fonctions, de relations et des symboles de constantes : on admettra [5] que dans ce cas, pour toute formule close ϕ sur la signature Σ, on a $M \models \phi$ si et seulement si $N \models \phi$: autrement dit, M et N satisfont exactement les mêmes formules closes sur la signature Σ, quand M et N sont isomorphes.

5. Cela se prouve par une induction sans surprise.

4.2 Corps algébriquement clos

On considère la théorie ACF des corps algébriquement clos [6] : c'est la théorie constituée des axiomes des corps commutatifs auquels on ajoute pour chaque $n \geq 1$ l'axiome

$$\forall x_0 \forall x_1 \cdots \forall x_{n-1} \exists x (x_0 + x_1 \cdot x + x_2 \cdot x^2 + \cdots + x_{n-1} \cdot x^{n-1} + x^n) = \mathbf{0}$$

comme dans le cours. On notera Σ la signature de cette théorie.

Pour un entier p, on note comme dans le cours F_p la formule $\mathbf{1} + \cdots + \mathbf{1} = \mathbf{0}$, où $\mathbf{1}$ est répété p fois. On note ACF_p la théorie des corps algébriquement clos de caractéristique p, pour $p \geq 0$. Formellement [7] : pour $p \geq 1$, ACF_p est constituée des axiomes de ACF et de la formule F_p et des formules $\neg F_q$ pour tout entier $0 < q < p$. ACF_0 est constituée des axiomes de ACF auquels on ajoute pour chaque $n \geq 1$ la formule $\neg F_n$.

On dit que deux ensembles A et B sont de même cardinal s'il existe une bijection entre A et B.

Tout ce qui suit est basé uniquement sur des arguments de logique, et nécessite uniquement les concepts et résultats suivants d'algèbre [8] :

(α) $\mathbb{C}$, le corps des complexes, est un corps algébriquement clos de caractéristique 0.

(β) Pour chaque entier premier p, il y a un corps $\overline{K_p}$ algébriquement clos de caractéristique p.

(γ) Si deux corps algébriquement clos sont de même caractéristique et de même cardinal alors ils sont isomorphes.

Question 17. *Montrer que ACF n'est pas une théorie complète.*

Solution : En effet, par exemple on n'a pas $ACF \vdash F_2$ ni $ACF \vdash \neg F_2$: $\mathbb{C}$ est un corps qui vérifie ACF et qui n'est pas de caractéristique 2 et donc on ne saurait avoir $ACF \vdash F_2$ (par le théorème de correction, tout modèle de ACF devrait satisfaire F_2). De même on connait des corps algébriquement clos de caractéristique 2 et donc on ne saurait avoir $ACF \vdash \neg F_2$. □

4.3 ACF_0 est complète

On dira qu'un modèle est fini (respectivement : infini) si son ensemble de base l'est.

On admettra qu'une modification de la preuve du théorème de complétude permet de le renforcer en le résultat suivant : Soit A un ensemble avec un nombre infini d'éléments. Toute théorie T sur une signature dénombrable [9] qui possède un modèle infini possède un modèle (dont l'ensemble de base est) de même cardinal que A.

Question 18. *Soit A un ensemble avec un nombre infini d'éléments. Soit T une théorie (consistante) dont tous les modèles sont infinis. Supposons que tous les modèles de T de même cardinal que A sont isomorphes. Démontrer que T est complète.*

Solution : On raisonne par l'absurde. Si T n'est pas complète, alors il existe une formule ϕ telle que $T \cup \{\phi\}$ et $T \cup \{\neg\phi\}$ admettent des modèles. Comme ces modèles sont en particulier modèles de T, ils sont infinis. D'après le résultat qui précède la question, on peut les prendre de même

6. On rappelle qu'un corps algébriquement clos est un corps commutatif où tout polynôme de degré supérieur ou égal à un admet une racine.

7. Ce n'est pas exactement comme dans le cours, car on met ici explicitement les formules $\neg F_q$ pour tout entier $0 < q < p$ (note : cela est équivalent quand p est premier, mais cette remarque n'est pas utile pour cet exercice et on prendra dans la suite la définition indiquée dans cet énoncé).

8. Pour les connaisseurs en algèbre : pour (β), il s'agit de la clôture algébrique de $\mathbb{Z}/p\mathbb{Z}$; pour (γ), cela découle du fait qu'un corps algébriquement clos est déterminé à isomorphisme près par sa caractéristique et son degré de transcendance. Mais comprendre ces concepts et ces faits n'est pas nécessaire pour ce sujet.

9. C'est-à-dire avec un nombre dénombrable de symboles de constantes, de fonctions et de relations. C'est le cas de la signature Σ.

cardinal que A. Alors par hypothèse ils sont isomorphes, mais l'isomorphisme ne peut préserver l'interprétation de ϕ, d'où la contradiction. □

Question 19. *Montrer que tous les modèles de ACF_0 sont infinis.*

Solution : On note $\overline{0}$ l'interprétation de $\mathbf{0}$, et $\overline{p}$ l'interprétation de $\mathbf{1}+\mathbf{1}+\cdots+\mathbf{1}$ où la constante $\mathbf{1}$ est répétée p fois, pour chaque entier p. Pour tout couple d'entier $i < j$, on a nécessairement $\overline{i} \neq \overline{j}$: en effet, sinon, par les axiomes des corps, on devrait avoir $\overline{j-i} = \overline{0}$, ce qui est imposible par l'axiome $\neg F_{j-i}$. Il y a donc au moins autant d'éléments que d'entiers dans l'ensemble de base du corps. □

Question 20. *En déduire que ACF_0 est complète.*

Solution : C'est une application directe de la question 18 : ACF_0 possède un modèle infini ($\mathbb{C}$) et deux modèles de ACF_0 de même cardinal sont isomorphes par le fait (γ). □

Commentaire : On peut aussi montrer que ACF_p est complète pour $p \geq 1$.

4.4 Théorème d'Ax

On veut d'abord démontrer le résultat suivant : Les propositions suivantes sont équivalentes. Soit ϕ une formule close sur la signature Σ.

i ϕ est vraie sur $\mathbb{C}$
ii ϕ est vraie dans tous les corps algébriquement clos de caractéristique 0
iii ϕ est vraie dans au moins un corps algébriquement clos de caractéristique 0
iv Pour tout entier m, il y a un entier premier $p > m$ tel que ϕ est vraie dans un corps algébriquement clos de caractéristique p
v Il y a un entier m tel que pour tout entier premier $p > m$, ϕ est vraie dans tous les corps algébriquement clos de caractéristique p

Le fait que [i] implique [iii] et que [v] implique [iv] est évident.

Question 21. *Montrer que [i], [ii] et [iii] sont équivalents.*

Solution : Ce n'est rien d'autre que la complétude de ACF_0. □

Question 22. *Montrer que [ii] implique [v]*

Solution : Supposons que $ACF_0 \models \phi$. Par le théorème de complétude, il y a une preuve de ϕ à partir des axiomes de ACF_0. Cette preuve n'utilise qu'un nombre fini d'assertions $\neg F_q$, donc pour un p suffisamment grand, $ACF_p \vdash \phi$, et donc aussi $ACF_p \models \phi$. □

Question 23. *Montrer que [iv] implique [ii]*

Solution : Supposons que $ACF_0 \not\models \phi$. Par la complétude de ACF_0, on a $ACF_0 \models \neg\phi$. Par le théorème de complétude, il y a une preuve de $\neg\phi$ à partir des axiomes de ACF_0. Cette preuve n'utilise qu'un nombre fini d'assertions $\neg F_q$, donc pour tout p suffisamment grand, $ACF_p \vdash \neg\phi$, et donc aussi $ACF_p \models \neg\phi$. Cela veut dire que donc [iv] devient fausse, contradiction. □

On admettra que pour tout entier premier p, le corps algébriquement clos $\overline{K_p}$ est tel que toute fonction polynomiale injective de $(\overline{K_p})^n$ dans $(\overline{K_p})^n$ est surjective.

Question 24. *Démontrer le théorème d'Ax : Toute fonction polynomiale injective de $\mathbb{C}^n \to \mathbb{C}^n$ est surjective.*

Solution : Supposons que ce ne soit pas le cas. Notons $\vec{x} = (X_1, X_2, \ldots, X_n)$. Soit $F(\vec{x})$ un contre-exemple avec $F(\vec{x}) = (F_1(\vec{x}), \ldots, F_n(\vec{x}))$ où chaque polynôme $F_i \in \mathbb{C}[X_1, X_2, \ldots, X_n]$ est de degré au plus d. On peut écrire une formule close $\Phi_{n,d}$ telle que, pour K un corps, $K \models \Phi_{n,d}$ si et seulement si toute fonction injective polynomiale $G : K^n \to K^n$ dont les fonctions coordonnées ont un degré au plus d est surjective. On peut quantifier sur les polynômes de degré au plus d en quantifiant sur les coefficients.

Par exemple, $\Phi_{2,2}$ est la formule

$$\forall a_{0,0} \forall a_{0,1} \forall a_{0,2} \forall a_{1,1} \forall a_{2,0} \forall b_{0,0} \forall b_{0,1} \forall b_{0,2} \forall b_{1,0} \forall b_{1,1} \forall b_{2,0}$$
$$\big(\forall x_1 \forall y_1 \forall x_2 \forall y_2$$
$$\sum a_{i,j} x_1^i y_1^j = \sum a_{i,j} x_2^i y_2^j \wedge \sum b_{i,j} x_1^i y_1^j = \sum b_{i,j} x_2^i y_2^j \Rightarrow x_1 = x_2 \wedge y_1 = y_2\big)$$
$$\Rightarrow \forall u \forall v \exists x \exists y \sum a_{i,j} x^i y^j = u \wedge \sum b_{i,j} x^y y^j = v.$$

On montre la propriété iv en prenant $\Phi_{n,d}$ pour la formule ϕ. Ainsi en appliquant les questions précédentes, on en déduira que $\mathbb{C} \models \Phi_{n,d}$ ce qui mène à une contradiction.

Montrons la propriété iv : soit un entier m. On prend un entier premier $p > m$. Par la propriété (β), il y a un corps algébriquement clos $\overline{K_p}$ de caractéristique p. Par la propriété ci-dessus, il satisfait la formule $\Phi_{n,d}$. □

Notes bibliographiques

La partie sur le théorème de Cobham est inspirée de notes de Cours de Arnaud Durand. Le théorème a été énoncé par Cobham dans [2], mais sans vrais détails sur la preuve. Se référer à [4] ou [1] pour des preuves détaillées.

L'idée de la partie sur la complétude des corps algébriquement clos est née du post dans le blog *Xor's Hammer* de Mkoconnor du 15 Août 2008. Voir `https://xorshammer.com/2008/08/15/axs-theorem/`. Elle est au final inspirée de [3].

Références

[1] Peter Clote and Evangelos Kranakis. *Boolean functions and computation models.* Springer Science & Business Media, 2013.

[2] A. Cobham. The intrinsic computational difficulty of functions. In Y. Bar-Hillel, editor, *Proceedings of the International Conference on Logic, Methodology, and Philosophy of Science*, pages 24–30. North-Holland, Amsterdam, 1962.

[3] David Marker et al. Introduction to the model theory of fields. In *Model theory of Fields*, pages 1–37. Association for Symbolic Logic, 1996.

[4] H. E. Rose. *Subrecursion, Functions and Hierarchies.* Clarendon Press, Oxford, 1984.