

Algorithmes. Modèles de calcul. PC 2.

Correction

1 Inverse d'une fonction bijective calculable

Montrer que l'inverse d'une fonction $f : \mathbb{N} \rightarrow \mathbb{N}$ calculable et bijective, est calculable.

Correction: *On peut calculer l'inverse de g de la façon suivante : sur l'entrée y , on teste pour $x = 0, 1, \dots$, si $f(x) = y$. On s'arrête dès qu'un x a trouvé un tel x . Puisque f est surjective on doit s'arrêter.*

2 Décidabilité et énumération croissante

1. Soit $E \subset \mathbb{N}$ un ensemble récursivement énumérable, énuméré par une fonction calculable f strictement croissante. Montrer que E est décidable.

Correction: *Pour tester si $x \in E$, on énumère les éléments de E jusqu'à trouver soit x , auquel cas on accepte, soit un élément plus grand, auquel cas on refuse.*

2. En déduire que tout sous-ensemble récursivement énumérable infini de \mathbb{N} contient un ensemble décidable infini.

Correction: *Pour extraire un ensemble décidable infini d'un ensemble récursivement énumérable infini, il suffit d'extraire une sous-suite strictement croissante de la suite des $f(n)$: on part avec $y = \max = 0$. Pour $n = 0, 1, \dots$,*

- *On calcule $y = f(n)$.*
- *Si $y > \max$ alors on fait $\max := y$, et on affiche $f(n)$*

3. Soit $E \subset \mathbb{N}$ un ensemble décidable. Montrer qu'il est énuméré par une fonction calculable f strictement croissante.

Correction: *Pour $n = 0, 1, \dots$, on teste si n est dans l'ensemble, et si oui, on l'affiche.*

3 RE = \exists R

Soit $A \subset \mathbb{N}^2$ un ensemble décidable de couples d'entiers.

On note $\exists A$ pour la (première) projection de A , à savoir le sous-ensemble de \mathbb{N} défini par

$$\exists A = \{x | \exists y \in \mathbb{N} \text{ tel que } (x, y) \in A\}.$$

1. Montrer que la projection d'un ensemble décidable est récursivement énumérable.
2. Montrer que tout ensemble récursivement énumérable est la projection d'un ensemble décidable.

Correction: *Pour tester si $x \in \exists A$, il suffit de tester pour $y = 0, 1, \dots$ si $(x, y) \in A$. On s'arrête dès qu'on trouve un y .*

Tout langage récursivement énumérable est énuméré par une fonction calculable f . Il correspond donc à la projection de l'ensemble $A = \{(n, f(n)) | n \in \mathbb{N}\}$. Cet ensemble A est bien décidable.

4 Quelques problèmes

Les problèmes suivants, où l'on se donne un algorithme A et l'on veut savoir

1. si $L(A)$ contient au moins deux entrées distinctes
2. si $L(A)$ accepte aucune entrée

sont-ils décidables ? semi-décidables ?

Correction: *Leur non-décidabilité découle du théorème de Rice.*

Le premier est récursivement énumérable : on énumère les triples (a, b, t) avec a et b deux mots distincts, t un entiers, et pour chacun on teste si A accepte a et b en temps t . Si oui, on accepte.

Le second n'est pas récursivement énumérable car son complément est récursivement énumérable : on énumère les paires (a, t) et on teste si a est accepté par A en temps t . Si oui on accepte.

5 Réels calculables

Un nombre réel a est dit *récursif* s'il existe des fonctions calculables F et G de \mathbb{N} dans \mathbb{N} telles que pour tout $n > 0$ on ait $G(n) > 0$ et

$$\left| a - \frac{F(n)}{G(n)} \right| \leq \frac{1}{n}.$$

1. Montrer que tout nombre rationnel est récursif.
2. Montrer que $\sqrt{2}$, π , e sont récursifs.

3. Montrer que le nombre réel $0 < a < 1$ est récursif si et seulement s'il existe un développement décimal récursif de a , c'est-à-dire une fonction calculable $H : \mathbb{N} \rightarrow \mathbb{N}$ telle que pour tout $n > 0$ on ait $0 \leq H(n) \leq 9$ et

$$|a| = \sum_{n=0}^{\infty} \frac{H(n)}{10^n}.$$

4. Montrer que l'ensemble des réels récursifs forme un sous corps dénombrable de \mathbb{R} , tel que tout polynôme de degré impair possède une racine.
5. Donner un exemple de réel non-récursif.

Correction: *Laissée au lecteur.*

6 Algorithme de taille minimale

On appelle *taille d'un algorithme* le nombre de symboles nécessaire pour le décrire comme un mot. On dit qu'un algorithme est *minimal* s'il n'existe aucun autre algorithme équivalent avec une description plus courte.

Montrer que le problème de déterminer si un algorithme est minimal n'est pas décidable (indice : utiliser le théorème de récursion).

Correction: *Supposons qu'un algorithme E énumère MIN, l'ensemble des algorithmes minimaux.*

On construit l'algorithme C suivant :

- Sur une entrée w , C obtient par le théorème de récursion son propre code $\langle C \rangle$.
- Puis C utilise l'énumérateur E pour trouver un algorithme D qui est de longueur plus grande que celle de C .
- Puis C simule D sur l'entrée w .

Parce que MIN est infini, C trouvera toujours un algorithme D de longueur plus grande que lui-même. Puisque C simule D, C et D sont équivalents. Puisque C est plus court que D, et lui est équivalent, D n'est pas minimal. Mais D est sur la liste des algorithmes minimaux, puisque produit par E. Contradiction.

7 Quines

Proposer un programme JAVA qui produit son propre code.

Correction:

```
public class Quine {
public static void main(String[] args) {
String[] str = {
"public class Quine {"
```

```
" public static void main(String[] args) {"  
" String[] str = {"  
" };"  
" for(int i=0;i<3;i++)System.out.println(str[i]);"  
" for(int i=0;i<9;i++)System.out.println((char)34+str[i]+(char)34+',');"  
" for(int i=3;i<9;i++)System.out.println(str[i]);"  
" }"  
"}"  
};  
for(int i=0;i<3;i++)System.out.println(str[i]);  
for(int i=0;i<9;i++)System.out.println((char)34+str[i]+(char)34+',');  
for(int i=3;i<9;i++)System.out.println(str[i]);  
}  
}
```