

Complexité et Calculabilité : TD4

Ensembles dénombrables et programmes LOOP.

4.1 Ensembles dénombrables

Rappels : Une énumération d'un ensemble infini D est une bijection $f : \mathbb{N} \rightarrow D$. On a donc $D = \{f(0), f(1), \dots\}$ et $f(i) \neq f(j)$ pour tout $i \neq j$. Un ensemble D s'appelle *dénombrable* s'il peut être énuméré. Il s'appelle *au plus dénombrable* s'il est fini ou dénombrable. On s'intéresse surtout aux énumérations calculables : la bijection $f : \mathbb{N} \rightarrow D$ doit être calculable. Un tel ensemble D s'appelle *dénombrable de manière effective*.

Exercice 4.1

Montrez que si D est un ensemble dénombrable, alors tout sous-ensemble $D' \subseteq D$ est au plus dénombrable.

Exercice 4.2

1. Montrez que \mathbb{Z} est un ensemble dénombrable.
2. Montrez que \mathbb{N}^2 est un ensemble dénombrable.
3. Montrez que si D et D' sont deux ensembles dénombrables, alors leur *produit cartésien* $D \times D'$ est un ensemble dénombrable.
4. Déduisez-en que si D est un ensemble dénombrable, alors pour tout k , D^k est un ensemble dénombrable.
5. On rappelle que $D^* = \bigcup_{k \in \mathbb{N}} D^k$. Peut-on déduire de ce qui précède que si D est dénombrable, alors D^* est dénombrable ?

Exercice 4.3

Montrez que les ensembles suivants sont dénombrables (de manière effective) :

1. l'ensemble des matrices avec entrées entières,
2. l'ensemble Σ^* des mots sur un alphabet (fini) Σ ,
3. l'ensemble des programmes C,
4. l'ensemble des arbres orientés,
5. l'ensemble des graphes orientés.

Exercice 4.4

On considère l'ensemble des suites infinies d'entiers $e : \mathbb{N} \rightarrow \mathbb{N}$. On suppose que cet ensemble est dénombrable, et qu'il est donc de la forme $\{e_1, e_2, \dots\}$.

1. On définit la suite f telle que pour tout $n \in \mathbb{N}$, $f(n) = e_n(n) + 1$. Cette suite est-elle une suite d'entiers ?
2. f peut-elle être l'un des e_n ?
3. Qu'en déduisez-vous sur la dénombrabilité de l'ensemble des suites infinies d'entiers ?

4. Cela contredit-il ce que vous savez sur la dénombrabilité des suites finies d'entiers ?
5. Utilisez cela pour déterminer si l'ensemble des nombres réels entre 0 et 1 est dénombrable ou non.

4.2 Programmes LOOP

Les programmes que vous aurez à écrire ici ne manipuleront que des entiers positifs ou nuls. Les seules instructions que vous pouvez utiliser (en-dehors des séquences que vous aurez définies), sont les instructions :

$x:=y+c$ ou $x:=c$: où c est une constante (0,1,2 ...), x et y sont des variables (potentiellement la même)

$x:=y-c$: avec les mêmes conditions (attention, ceci calcule la différence tronquée, c'est à dire $\max(y-c, 0)$).

LOOP(x) DO P OD : qui exécute exactement x fois les instructions du programme P (indépendamment des changements sur la variable x).

Exercice 4.5

Écrivez deux programmes LOOP qui calculent respectivement les fonctions suivantes et affectent la valeur calculée à `res` :

$$isPositive(x) = \begin{cases} 1 & \text{si } x > 0, \\ 0 & \text{si } x = 0. \end{cases} \quad isZero(x) = \begin{cases} 1 & \text{si } x = 0, \\ 0 & \text{sinon.} \end{cases}$$

Exercice 4.6

Soient P, Q deux programmes LOOP. Écrivez un programme LOOP qui simule l'instruction if-then-else :

IF ($x > 0$) THEN P ELSE Q FI

Exercice 4.7

Écrivez un programme LOOP *Somme* qui calcule la somme de deux entiers x et y .

Exercice 4.8

Écrivez un programme LOOP *DiffTr* qui calcule la différence tronquée $x \dot{-} y$, définie par $x \dot{-} y = x - y$ si $x \geq y$, et 0 sinon.

Exercice 4.9

Écrivez un programme LOOP *Prod* qui calcule le produit de deux entiers x et y .

Exercice 4.10

Écrivez un programme LOOP *Min* qui calcule $\min(x, y)$.

Exercice 4.11

Écrivez un programme LOOP *Quotient* qui calcule le quotient de la division entière de x par y .

Exercice 4.12

Écrivez un programme LOOP *Dec* qui calcule le nombre de décimales de la représentation décimale d'un entier. Par exemple $Dec(5) = 1$, $Dec(1548) = 4$, $Dec(1236987) = 7$.

Exercice 4.13

Écrivez un programme LOOP qui simule la séquence d'instructions en C suivante :

```
{ int x = n;
  int prop=1;
  while(x > 0 && prop > 0){
    Q;
    x=x-1;
  }}
```

où le programme Q (codable en LOOP) modifie `prop` mais pas `x`. Pourquoi la variable `x` joue-t-elle un rôle important ?