TD n°2 - Ensembles énumérables : suite

1 Exercice I : Ensembles énumérables - mieux comprendre

Même exercice que Exercice 2 du TD1

2 Exercice 2 Enumération des fonctions totales

Nous allons montrer dans cet exercice qu'il n'est pas possible d'avoir un système de programmation "raisonnable" où les programmes s'arrêtent toujours. Supposons que ce système existe et notons $[x|y]'$ le résultat du calcul du x-ième programme sur l'entrée y.

1. On suppose que dans un de nos programmes on peut en appeler un autre, et que la fonction successeur soit calculable. Montrez que $g : x \mapsto [x|x]' + 1$ est calculable dans ce système – on notera n son numéro : [n|·] = g(·).
2. Que vaut g(n) ? En déduire qu'un tel système n'existe pas.

2.1 Question n°1 : MQ g est calculable

> On sait que $[x|y]$' est calculable et que la fonction successeur est aussi calculable.

> On a donc un programme $n : x \mapsto [x|x]' + 1$ tel que n : *calculable* + *calculable* alors il calcule bien g donc g est calculable dans ce système.

2.2 Question n°2 : MQ contradiction

> Par l'appel de n sur n on a $[n|n] = g(n)$.

> Mais par la définition de g sur l'entrée n on a $g(n) = [n|n]' + 1$ alors on a une contradiction.

On en déduit alors que le système de programmation "raisonnable" de l'énoncé n'existe pas.

3 Exercice 3 : Ensembles énumérables - clôture

Nous prouvons dans cet exercice que la classe des ensembles énumérables et bien close par union, intersection et produit cartésien. Soient A et B deux ensembles énumérables.

1. Montrez que A ∪ B est énumérable.
2. Montrez que A ∩ B est énumérable.
3. Montrez que A × B $= < x, y > x \in A \wedge y \in B$ est énumérable.

On a deux définitions de l'**énumérabilité** :

- $\underline{x \in A}$: x appartient à A si le programme a, sur l'entrée x, converge.
- $\underline{a(x) \in A}$: a(x) appartient à A si le programme a converge sur l'entrée x. On a alors x qui est l'indice d'un élément qui est dans A ou pas.

Ainsi pour chaque question on va utiliser une des deux versions de la définition.

3.1 Question n°1 : A ∪ B

> On utilise la définition 2, donc on a un programme a qui est l'énumération de l'ensemble A et un programme b qui est l'énumération de l'ensemble B.

> On cherche alors le programme c qui affiche à la fois les éléments de A et à la fois les éléments de B. On va alors afficher une fois sur deux, un élément de A puis un élément de B.

On va donc trouver la fonction d'énumération c de $A \cup B$:

```
c : si x = 2*p alors return [a|p]
    sinon
        si x = 2*p + 1 alors return [b|p]
```

On a donc le programme c qui est l'énumération de $A \cup B$.

3.2 Question n°2 : A ∩ B

> On utilise la définition 1, donc on a un programme a tel que, $[a|x] \searrow$ (converge) si et seulement si $x \in A$ et un programme b tel que, $[b|x] \searrow$ (converge) si et seulement si $x \in B$.

> On cherche alors le programme c qui énumère les éléments de A qui sont aussi dans B. On va alors regarder si a converge, puis voire si b converge.

On va donc trouver la fonction d'énumération c de $A \cap B$:

```
c : [a|x];[b|x];return 1;
```

On a donc le programme c qui est l'énumération de $A \cap B$.

3.3 Question n°3 : A × B

> On utilise la définition 1, donc on a un programme a tel que son domaine soit A et un programme b tel que son domaine soit B.

> On cherche alors le programme c où son domaine est $A \times B$. On va alors regarder si a converge sur l'entrée x, puis voire si b converge sur l'entrée y.

On va donc trouver la fonction d'énumération c de $A \times B$:

On a n = <p, q>, p = Pi1(n) et q = Pi2(n) :

```
c : [a|Pi1(n)];[b|Pi2(n)];return 1;
```

On a donc le programme c qui est l'énumération de $A \times B$.