Correction TP 7 : Principe d'induction

Licence 2 MASS semestre 2, 2007/2008

Exercice 1 : Ensembles définis inductivement

a- Soit $A^{'} = \{5^n, n \in I\!N\}$

- montrons que $A \subseteq A^{'}$.

  Soit pour tout $a \in A$ la proposition $P(a)$: "$a \in A^{'}$", montrons par **induction** que $\forall a \in A$, $P(a)$ est vraie.

  * **base** : $1 = 5^0 \in A^{'}$ par définition de $A^{'}$, donc $P(1)$ est vraie.
  * **hérédité** : Supposons qu'il existe $a \in A$ tel que $P(a)$ est vraie.
    Montrons que $P(5a)$ est vraie.
    par hypothèse, $a \in A^{'}$ donc $\exists k \in I\!N$ tel que $a = 5^k$.
    Or $5a = 5 \times 5^k = 5^{k+1} \in A^{'}$ par définition de $A^{'}$.
    donc $P(5a)$ est vraie.

  Ainsi d'après le principe d'induction, on peut conclure que $\forall a \in A$, $P(a)$ est vraie et en déduire que $A \subseteq A^{'}$.

- Montrons que $A^{'} \subseteq A$.

  Soit pour tout $k \in I\!N$ la proposition $P(k)$: "$5^k \in A'$, montrons par **récurrence** que $\forall k \in I\!N$, $P(k)$ est vraie.

  * **base** : $5^0 = 1 \in A$, donc $P(0)$ est vraie.
  * **hérédité** : Supposons qu'il existe $k \in I\!N$ tel que $P(k)$ est vraie.
    Montrons que $P(k+1)$ est vraie.
    $5^{k+1} = 5 \times 5^k$, or par hypothèse de récurrence $5^k \in A$ et par la définition inductive de A, $5 \times 5^k \in A$. Ainsi, $P(k+1)$ est vraie.

  D'après le principe de récurrence, on peut conclure que $\forall k \in I\!N$, $P(k)$ est vraie et en déduire que $A^{'} \subseteq A$.

Finalement, $A \subseteq A^{'}$ et $A^{'} \subseteq A$ donc $A = A^{'}$.

b- Soit $B^{'} = \{2^n 3^m, (n,m)^2 \in I\!N - \{(0,0)\}\}$.

- montrons que $B \subseteq B^{'}$.

  Soit pour tout $x \in B$ la proposition $P(x)$: "$x \in B^{'}$, montrons par induction que $\forall x \in B$, $P(x)$ est vraie.

  * **base** :
    $2 = 2^1\ 3^0 \in B^{'}$ car $0, 1 \in I\!N - \{(0,0)\}$,
    $3 = 2^0\ 3^1 \in B^{'}$ car $1, 0 \in I\!N - \{(0,0)\}$,
    donc $P(2)$ et $P(3)$ sont vraies.

* **hérédité** : Supposons qu'il existe deux éléments a et b de B tels que $P(a)$ et $P(b)$ sont vraies, montrons que $P(ab)$ est vraie.
  Par hypothèse, il existe $(m,n) \in I\!N - \{(0,0)\}$ et $(p,q) \in I\!N - \{(0,0)\}$ vérifiants $a = 2^m 3^n$ et $b = 2^p 3^q$.
  $ab = 2^m 3^n 2^p 3^q = 2^{m+p} 3^{n+q}$ avec $(m+n, p+q) \neq (0,0)$, ce qui signifie que $ab \in B^{'}$ par définition de $B^{'}$.
  ainsi, $P(ab)$ est vraie.

D'après le principe d'induction, pour tout $x \in B$ $P(x)$ est vraie, ce qui permet d'en déduire que $B \subseteq B^{'}$.

– Montrons par que $B^{'} \subseteq B$.

Soit pour tout $(n,m) \in I\!N - \{(0,0)\}$ la proposition $P(n,m)$: "$2^n\ 3^m \in B$", montrons par induction que $\forall (n,m) \in I\!N - \{(0,0)\}$, $P(n,m)$ est vraie.

* **base** : il existe deux cas de base les couples $(1,0)$ et $(0,1)$.
  $2^1\ 3^0 = 2$ et $2^0\ 3^1$. Or par définition de B, $2 \in B$ et $3 \in B$ donc $P(1,0)$ et $P(0,1)$ sont vraies.
* **hérédité** : Supposons qu'il existe $(n,m) \in I\!N - \{(0,0)\}$ tel que $P(n,m)$ est vraie, montrons que $P(n+1,m)$ et $P(n,m+1)$ sont vraies.
  · $2^{n+1} 3^m = 2 \times 2^n\ 3^m$, or par hypothèse de récurrence $2^n\ 3^m \in B$ et par la définition inductive de B, $2 \in B$ et $2 \times 2^n\ 3^m \in B$, on en déduit que $2^{n+1}\ 3^m \in B$ et $P(n+1,m)$ est vraie.
  · $2^n\ 3^{m+1} = 3 \times 2^n\ 3^m$, or par hypothèse de récurrence $2^n\ 3^m \in B$ et par la définition inductive de B, $3 \in B$ et $3 \times 2^n\ 3^m \in B$, on en déduit que $2^n\ 3^{m+1} \in B$ et $P(n,m+1)$ est vraie.

  D'après le principe d'induction, pour tout $(n,m) \in I\!N - \{(0,0)\}$ $P(n,m)$ est vraie et on peut en déduire que $B^{'} \subseteq B$.

Finalement $B = B^{'}$.

Exercice 2 : Définition inductive sur les mots

a- N'importe quelle chaîne de caractères ne contenant que des a des b et des c fait l'affaire !

b- $M = \{a^k b^k | k \in I\!N\}$, définition inductive

– **base** : $\epsilon \in M$
– **induction** : si $u \in M$ alors $aub \in M$

Ce qui donne les ensembles :

– $\mathbf{B} = \{\epsilon\}$
– $\mathbf{F} = \{$augmente : $\begin{array}{ccc} M & \rightarrow & M \\ u & \rightarrow & aub \end{array}$ $\}$

Exercice 3 : Terminaison d'algorithme récursif

En utilisant la définition inductive de l'ensemble des listes, montrons par induction que l'algorithme *longueur* se termine.

- **base** : pour la liste vide noté ϵle test n'est pas vérifié et l'algorithme s'arrête en exécutant "retourner 0".
- **induction** : supposons que l'algorithme s'arrête pour une liste l. Montrons qu'il s'arrête pour toutes listes (e, l) avec $e \in \mathcal{A}$.

  $(e, l) \neq \epsilon$ n'est pas la liste vide, le test n'est donc pas vérifié.

  Par hypothèse, l'algorithme *longueur* s'arrête pour la liste l, donc l'exécution de "retourner 1 + longueur(listeQueue())" se termine aussi.

  Donc *logueur* se termine pour toutes listes (e, l).

D'àprès le principe de récurrence, l'algorithme termine quelle que soit la liste donnée en paramètre.

Exercice 4 : Définition inductive de fonctions

a- Soit pour tout entier n, $P(n)$: longueur$((a_1, a_2, \ldots, a_n)) = n$. Montrons par récurrence que $\forall n \in I\!N$ $P(n)$ est vraie.

- **base** : pour $n = 0$, la liste est vide, elle ne contient aucun élément, donc la propriété est vraie,
- **hérédité** : Supposons qu'il existe un entier n tel que $P(n)$ soit vraie. Par définition inductive des listes, il existe $a_0 \in A$ et $l = (a_1, a_2, \ldots, a_n)$ tel que $(a_0, a_1, a_2, \ldots, a_n) = \text{ajout}_{a_0}(l)$. Par définition de la fonction longueur, longueur(ajout$_{a_0}((a_1, a_2, \ldots, a_n)) = 1 +$ longueur$((a_1, a_2, \ldots, a_n)$. Donc par hypothèse, longueur$((a_0, a_1, a_2, \ldots, a_n)) = n + 1$. D'o $P(n + 1)$ est vraie.

D'après le principe de récurrence, pour tout n entier, $P(n)$ est vraie.

b- Concaténation de deux listes :

- **base** : concat$(\epsilon, l_2) = l_2$.
- **induction** : si $e \in A$, $l_1 \in L$ et $l_2 \in L$, concat$(ajout_e(l_1),\ l_2) = ajout_e(concat(l_1, l_2))$

c- preuve par induction sur l'ensemble des listes...

- **base** : pour toute liste l_2, $concat(\epsilon, l_2) = l2$ or $(\epsilon.l_2) = l_2$

- **hérédité** : Supposons qu'il existe $(l_1, l_2) \in L^2$ tel que $concat(l_1, l_2) = (l_1.l_2)$.
  $concat(ajout_e(l_1), l_2) = ajout_e(concat(l_1, l_2)) = (e.l_1.l_2)$. Or $(ajout_e(l_1).l_2) = (e.l_1.l_2)$
  D'o $concat(ajout_e(l_1), l_2) = (ajout_e(l_1).l_2)$.

D'après le principe d'induction, pour tout couple de listes, $concat(l_1, l_2) = (l_1.l_2)$.

Exercice 5 : Arbre binaire

b- Ensemble des arbres binaires $\mathcal{T}$:

- **B** $= \{\epsilon\}$
- **F** $= \{enracine_a \mid a \in A$ et $enracine_a(g, d) = (a, g, d)\}$

c-
- hauteur :
  - **base** : h(ϵ) = 0
  - **induction** : h(a,g,d) = 1 + max(h(g), h(d))
- nombre de feuilles :
  - **base** : f(ϵ) = 0, f((a, ϵ, ϵ)) = 1,
  - **induction** : f((a,g,d)) = f(g) + f(d)
- nombre de noeuds :
  - **base** : n(ϵ) = 0
  - **induction** : n((a,g,d)) = 1 + n(g) + n(d)

d- Preuve par induction :
n(x) $\leq 2^{h(x)} - 1$:

- **base** : n(ϵ) $= 0 \leq 0 = 2^0 - 1 = 2^{h(\epsilon)} - 1$, donc la propriété est vraie pour les cas de base,
- **induction** : Soient $a \in A$, g, d $\in \mathcal{T}$, supposons que la propriété est vraie pour ces deux arbres. D'après la définition inductive des fonctions on a n((a,g,d)) = 1 + n(g) + n(d) et h(a,g,d) = 1 + max$_a$(h(g), h(d)). Or par hypothèse : n((a,g,d)) = 1 + n(g) + n(d) $\leq 2^{h(g)} - 1 + 2^{h(d)} - 1 + 1 \leq 2^{max(h(g),h(d))} + 2^{max(h(g),h(d))} - 1$ $\leq 2^{1+max(h(g),h(d))} - 1 \leq 2^{h(a,g,d)} - 1$