LICENCE O.M.I. - T.D. Induction

Corrigé

EXERCICE 1 On rappelle que les nombres de Fibonacci sont définis par $F_n = F_{n-1} + F_{n-2}$ pour $n > 1$ et $F_0 = 0,\ F_1 = 1$.

On va démontrer par récurrence sur $n > 0$ un certain nombre d'égalités. Puisque l'on considère $n > 0$, le cas de base des récurrences sera $n = 1$.

1) À montrer : $F_1 + F_3 + \cdots + F_{2n-1} = F_{2n}$

- Cas de base : $n = 1$.
  On a $F_1 + .. + F_{2n-1} = F_1 = 1 = 0 + 1 = F_0 + F_1 = F_2 = F_{2n}$.
- Hypothèse de récurrence : on suppose que pour un certain $n > 0$, on a $F_1 + .. + F_{2n-1} = F_{2n}$. Montrons que le résultat reste vrai pour $n + 1$.
  On a, $F_2(n+1) = F_{2n+2} = F_{2n+1} + F_{2n}$. Par hypothèse d'induction, $F_{2n} = F_1 + .. + F_{2n-1}$. Comme $2n + 1 = 2(n+1) - 1$, on a bien $F_2(n+1) = F_1 + .. + F_{2n-1} + F_{2(n+1)-1}$.

2) À montrer : $F_{n+1}F_{n-1} - F_n^2 = (-1)^n$

- Cas de base : $n = 1$.
  $F_2.F_0 - F_1^2 = 1.0 - 1 = -1 = (-1)^1$
- Hypothèse de récurrence : $F_{n+1}.F_{n-1} - F_n^2 = (-1)^n$.
  On a

$$
\begin{array}{rcll}
F_{(n+1)+1}.F_{(n+1)-1} - F_{n+1}^2 & = & F_{n+2}.F_n - F_{n+1}^2 & \\
 & = & (F_{n+1} + F_n).F_n - F_{n+1}^2 & \text{par déf. de } F_{n+2} \\
 & = & F_n^2 + F_n.F_{n+1} - F_{n+1}^2 & \text{en développant} \\
 & = & F_n^2 + F_{n+1}(F_n - F_{n+1}) & \text{en factorisant} \\
 & = & F_n^2 + F_{n+1}(F_n - F_{n-1} - F_n) & \text{par déf. de } F_{n+1} \\
 & = & F_n^2 - F_{n+1}.F_{n-1} & \text{en simplifiant} \\
 & = & -(F_{n+1}.F_{n-1} - F_n^2) & \text{en jouant avec les signes} \\
 & = & -(-1)^n & \text{par hypothèse de récurrence} \\
 & = & (-1)^{n+1} &
\end{array}
$$

3) À montrer : $F_1^2 + F_2^2 + \cdots + F_n^2 = F_nF_{n+1}$

- Cas de base : $F_1^2 = 1 = F_1.F_2$
- Hypothèse de récurrence : $F_1^2 + F_2^2 + \cdots + F_n^2 = F_n.F_{n+1}$. Montrons que $F_1^2 + F_2^2 + \cdots + F_n^2 + F_{n+1}^2 = F_{n+1}.F_{n+2}$.

$$
\begin{array}{rcll}
F_1^2 + F_2^2 + \cdots + F_n^2 + F_{n+1}^2 & = & F_n.F_{n+1} + F_{n+1}^2 & \text{par hypothèse de récurrence} \\
 & = & F_{n+1}.(F_n + F_{n+1}) & \text{factorisation} \\
 & = & F_{n+1}.F_{n+2} & \text{par déf.}
\end{array}
$$

4) À montrer : F_{3n} est pair.

- Cas de base : $F_3 = 1 = F_1 + F_2 = 2$ et 2 est pair.
- Hypothèse de récurrence : F_{3n} est pair. Montrons que $F_{3(n+1)}$ est également pair.

$$
\begin{array}{rcll}
F_{3(n+1)} & = & F_{3n+3} & \\
 & = & F_{3n+2} + F_{3n+1} & \text{par déf. de} F_{3n+3} \\
 & = & (F_{3n} + F_{3n+1}) + F_{3n+1} & \text{par déf. de} F_{3n+2} \\
 & = & F_{3n} + 2F_{3n+1} &
\end{array}
$$

Par hypothèse de récurrence, F_{3n}. Comme $2F_{3n+1}$ est pair et que la somme de deux entiers pair est un nombre pair, on a que $F_{3(n+1)}$ est pair.

5) Soit $\varphi = \frac{1+\sqrt{5}}{2}$, à montrer : $\varphi^{n-2} \leq F_n \leq \varphi^{n-1}$.
NB : on utilise une récurrence totale où l'hypothèse a la forme : *pour tout n', si $n' \leq n$ alors ....*

- Cas de base : $n = 1$. Il faut montrer que $\varphi^{-1} \leq F_1 \leq \varphi^1$. On vérifie facilement que $\frac{2}{1+\sqrt{5}} \leq 1$ et $1 \leq \frac{1+\sqrt{5}}{2}$.
- Hypothèse de récurrence : *pour tout n', si $n' \leq n$ alors $\varphi^{n-2} \leq F_n$ et $F_n \leq \varphi^{n-1}$*. Montrons *a)* $\varphi^{n-1} \leq F_{n+1}$ et *b)* $F_{n+1} \leq \varphi^n$.
  Remarquons au préalable que $\varphi^2 = (1 + \varphi)$.
  a) En appliquant l'hypothèse de récurrence à F_n et F_{n-1} (qui sont chacun inférieur ou égal à n) on a les deux inégalités :
  $$\begin{array}{rcl} \varphi^{n-3} & \leq & F_{n-1} \\ \varphi^{n-2} & \leq & F_n \end{array}$$
  Comme toutes les valeurs sont positives, on obtient par sommation :
  $$\varphi^{n-3} + \varphi^{n-2} \leq F_{n-1} + F_n$$
  c'est à dire :
  $$\varphi^{n-3}(1 + \varphi) \leq F_{n+1}$$
  et, en tenant compte de la remarque préalable :
  $$\varphi^{n-1} \leq F_{n+1}$$
  b) De façon analogue, en utilisant l'hypothèse de récurrence pour $n - 1$ et n on obtient que
  $$F_{n-1} + F_n \leq \varphi^{n-2} + \varphi^{n-1}$$
  Ce qui, compte tenu de la remarque préalable donne bien que $F_{n+1} \leq \varphi^{n-1}$.

EXERCICE 2 Le but de cet exercice est de montrer qu'il ne faut jamais oublier létude du cas de base dans une démonstration par récurrence.

1) Étude de la propriété $P(n) := 9|10^n - 1$ avec $n \in I\!N$ et $9|10^n - 1 \equiv \exists k.10^n - 1 = 9k$.

a) On montre la validité du pas de récurrence, c'est-à-dire la validité de la formule : $\forall n \in I\!N.P(n) \Rightarrow P(n + 1)$.
Supposons donc qu'il existe k tel que $10^n - 1 = 9k$. On a que $10^{n+1} - 1 = 10^{n+1} - 10 + 10 - 1 = 10(10^n - 1) + 10 - 1 = 10(10^n - 1) + 9$. Or, par hypothèse, il existe k tel que $10^n - 1 = 9k$. D'où $10^{n+1} - 1 = 10(9k) + 9 = 9(10k + 1)$. Il existe donc $k' = 10k + 1$ tel que $10^{n+1} - 1 = 9k'$ ce nous donne $P(n + 1)$.

b) On vérifie facilement que $P(0)$ puisque $10^0 - 1 = 0 = 9.0$.

c) On peut donne légitimenent déduire que $\forall n \in I\!N.P(n)$.

2) Étude de $Q(n) := 9|10^n + 1$.

a) On montre, là aussi la validité de pas de récurrence : $\forall n \in I\!N.Q(n) \Rightarrow Q(n + 1)$.
Supposons k tel que $10^n + 1 = 9k$. En remarquant que $10^{n+1} + 1 = 10(10^n + 1) - 9$, on obtient $Q(n + 1)$.

b) En revanche, on ne peut trouver aucun entier m tel que $Q(m)$. Si $m = 0$, on a $10^0 + 1 = 2$ qui n'est pas divisible par 9. On peut même montrer que pour tout $n \in I\!N$, $10^n + 1 \bmod 9 = 2$.
Par induction sur n

- Cas de base : *cf supra*

– Hypothèse de récurrence: $10^n + 1 \bmod 9 = 2$. Montrons $10^{n+1} + 1 \bmod 9 = 2$.
Par hypothèse de récurrence et définition du modulo, il existe k telque $10^n + 1 = 9k + 2$. On a: $10^{n+1} + 1 = 10(10^n + 1) - 9 = 10(9k + 2) - 9 = 9(10k + 1) + 2$. D'où le résultat escompté.

c) Même si le pas de récurrence est valide ($Q(n) \Rightarrow Q(n+1)$), il ne suffit pas à lui seul à conclure que $\forall n \in IN.Q(n)$ puisque l'on ne peut jamais « amorcer la pompe » de la récurrence.

3) Étude de $R(n) := 3 \mid 4^n + 7^n$

a) Montrons $R(n) \Rightarrow R(n+1)$.
Posons $A_n = 4^n + 7^n$ et supposons que $3 \mid 4^n + 7^n$ pour un certain n. On remarque que $A_{n+1} - A_n = 3.(4^n + 2.7^n)$. Par hypothèse, il existe k tel que $A_n = 3.k$, d'où $A_{n+1} = 3.(4^n + 2.7^n + k)$, i.e. $R(n+1)$.

b) Comme pour $Q(n)$, il n'existe aucun entier n satisfaisant $R(n)$. En fait, on peut montrer que $A_n \bmod 3 = 2$, pour tout $n \in IN$.

EXERCICE 3 Pour donner une définition récursive de $f(n) = a^{2^n}$, il faut savoir exprimer $a^{2^{n+1}}$ en fonction de a^{2^n}. C'est le sens de l'indication: « *On pourra remarquer que* $a^{2^{n+1}} = (a^{2^n})^2$. »

Il est alors trivial de donner la définition récursive de f:

$$\begin{cases} f(0) &= a \\ f(n+1) &= f(n)^2 \end{cases}$$

EXERCICE 4 Une définition récursive qui réclame 4 cas de base et un pas de 4.

1) En développant l'expression $(n+1)^2 - (n+2)^2 - (n+3)^2 + (n+4)^2$, on obtient:

$$\begin{aligned} (n+1)^2 - (n+2)^2 - (n+3)^2 + (n+4)^2 &= n^2 + 1 + 2n - n^2 - 4 - 4n - n^2 - 9 - 6n + n^2 + 16 + 8n \\ &= 4 \end{aligned}$$

2) On veut établir que $\forall m \in IN \exists n \in IN.m = \sum_{i=1}^{n} \varepsilon_i i^2$ avec $\varepsilon_i \in \{1, -1\}$. On procède par récurrence sur m en considérerant 4 cas de base.

a) si $m = 0$, on prend $n = 7$ et on a $0 = 1^2 + 2^2 - 3^2 + 4^2 - 5^2 - 6^2 + 7^2$.
b) si $m = 1$, on prend $n = 1$ et on a $1 = 1^2$.
c) si $m = 2$, on prend $n = 4$ et on a $2 = -1^2 - 2^2 - 3^2 + 4^2$.
d) si $m = 3$, on prend $n = 2$ et on a $3 = -1^2 + 2^2$.
e) Étapde de récurrence: supposons que qu'il existe n tel que $\sum_{i=1}^{n} \varepsilon_i i^2$ et montrons que $\exists n \in IN. \sum_{i=1}^{n+4} \varepsilon_i i^2$.
Par hypothèse de récurrence, on sait que $m = \sum_{i=1}^{n} \varepsilon_i i^2$. En utilisant 1), on peut écrire $m + 4 = \sum_{i=1}^{n} \varepsilon_i i^2 + n + 1)^2 - (n+2)^2 - (n+3)^2 + (n+4)^2$. En posant $n' = n + 4$ on a donc montré qu'il existe n' tel que $m + 4 = \sum_{i=1}^{n'} \varepsilon_i i^2$.

Remarque: on a bien un résultat valide surtout les entiers puisqu'il est vari des 4 premiers et que tout entiers $m > 3$ peut s'écrire sous la forme $m' + 4$.

EXERCICE 5 On rappelle la définition (inductive) de l'ensemble des arbres binaires étiquetés:

1. $\emptyset$ est un arbre.
2. si a est une étiquette et si t_1 et t_2 sont des arbres alors $t = (a, t_1, t_2)$ est un arbre.

On appelle *feuille* un arbre de la forme $(a, \emptyset, \emptyset)$.

$n(t)$ Le nombre de nœuds d'un arbre est le nombre d'étiquette qu'il contient. On les compte récursivement ainsi:

$$\begin{cases} n(\emptyset) &= 0 \\ n(a, t_1, t_2) &= 1 + n(t_1) + n(t_2) \end{cases}$$

$ar(t)$ Le nombre d'arêtes s'onbtient de façon analogue en comptant 2 chaque fois que l'on descend récursivement :

$$\begin{cases} ar(\emptyset) & = & 0 \\ ar(a, t_1, t_2) & = & 2 + ar(t_1) + ar(t_2) \end{cases}$$

$h(t)$ La hauteur d'un arbre est la longueur maximale obtenue en comptant 1 chaque fois que l'on descend récursivement :

$$\begin{cases} h(\emptyset) & = & 0 \\ h(a, t_1, t_2) & = & 1 + max(h(t_1), h(t_2)) \end{cases}$$

$f(t)$ Le nombre de feuille est le nombre de sous arbres de la forme $(a, \emptyset, \emptyset)$. Pour les compter, il faut ici aussi descendre récursivement dans l'arbre, mais s'arréter dés que l'on trouve une feuille :

$$\begin{cases} f(\emptyset) & = & 0 \\ f(a, \emptyset, \emptyset) & = & 1 \\ f(a, t_1, t_2) & = & f(t_1) + f(t_2) \end{cases}$$

Remarque : la définition de f donne $f(\emptyset) = 0$, ce qui premet de compter correctement le nombre de feuilles d'un arbre de la forme $f(a, \emptyset, t)$ où $t \neq \emptyset$.

EXERCICE 6 La définition inductive de l'ensemble des arbres donne le principe de raisonnement par récurrence suivant : si t est un arbre, pour montrer $P(t)$, on montre :

- Cas de base : $P(\emptyset)$.
- Étape de récurrence : on suppose $P(t_1)$ et $P(t_2)$ pour montrer que $P(a, t_1, t_2)$ pour une étiquette a quelconque.

1) Montrons, selon ce principe, que $n(t) \leq 2^{h(t)} - 1$ où $n(t)$ et $h(t)$ sont les fonctions définies à l'exercice précédent.

- Cas de base : il faut montrer que $n(\emptyset) \leq 2^{h(\emptyset)} - 1$, c'est-à-dire, par définition de n et h : $0 \leq 2^0 - 1$. Ce qui est immédiat.
- Soit $t = (a, t_1, t_2)$. Supposons que $n(t_1) \leq 2^{h(t_1)} - 1$ et que $n(t_2) \leq 2^{h(t_2)} - 1$. Il faut montrer que $n(t) \leq 2^{h(t)} - 1$.
  Remarquons que, par définition de h (et de max), on a $h(t_1) \leq h(t) - 1$ et $h(t_2) \leq h(t) - 1$. En utilisant les hypothèses de récurrence et les deux inégalités remarquées, on obtient par sommation que $n(t_1) + n(t_2) \leq 2^{h(t)-1} + 2^{h(t)-1} - 2$. D'où, en utilisant l'égalité $n(a, t_1, t_2) = n(t_1) + n(t_2) + 1$ et en simplifiant : $n(t) \leq 2.2^{h(t)-1} - 1 = 2^{h(t)} - 1$.

2) Montrons $f(t) \leq 2^{h(t)-1}$. Le principe de récurrence suit ici le principe de définition de f avec deux cas de base ($\emptyset$ et $(a, \emptyset, \emptyset)$).

- Premier cas de base : il faut montrer que $f(\emptyset) \leq 2^{h(\emptyset)-1}$, c'est-à-dire $0 \leq 2^{-1}$. Ce qui est vrai.
- Second cas de base : il faut montrer que $f(a, \emptyset, \emptyset) \leq 2^{h(a,\emptyset\emptyset,)-1}$, c'est-à-dire $1 \leq 2^{1-1}$. Ce qui est également vrai.
- Étape récursive : on pose $t = (a, t_1, t_2)$ et on suppose que $f(t_1) \leq 2^{h(t_1)-1}$ et que $f(t_2) \leq 2^{h(t_2)-1}$.
  En utilisant les inégalités remarquées ci-dessus et les hypothèses de récurrence, on obtient que $f(t_1) \leq 2^{max(h(t_1),h(t_2))-1}$ et $f(t_2) \leq 2^{max(h(t_1),h(t_2))-1}$. Par sommation, définition de f et simplification, on a $f(t) \leq 2^{max(h(t_1),h(t_2))}$. Comme $max(h(t_1), h(t_2)) = h(t) - 1$, on a bien que $f(t) \leq 2^{h(t)-1}$.

EXERCICE 7 Un arbre binaire est *strict* si chacune de ses branche se termine par une feuille. On n'a donc pas de sous arbre de la forme $(a, \emptyset, t_2)$ avec $t_2 \neq \emptyset$ ni $(a, t_1, \emptyset)$ avec $t_1 \neq \emptyset$.

1) Inductivement, cela signifie que le cas de base des arbres strict sont les feuilles. D'où la définition :

- pour toute étiquette a, l'arbre $(a, \emptyset, \emptyset)$ est strict.

– si t_1 et t_2 sont des arbres binaires stricts alors (a, t_1, t_2) l'est aussi, quelque soit a.

Remarquons que les arbres binaires stricts ainsi définis ne sont jamais vides, au sens des arbres binaires. Du coup, il faut amender un peu les définitions des fonctions n, a et f de l'exercice 5. Pour éviter toute équivoque, on notera (a) pour $(a, \emptyset, \emptyset)$. Ainsi, le symbole $\emptyset$ ne fait plus partie de notre langage. Et le cas de base des récurrence est $t = (a)$.

$$\left\{ \begin{array}{rcl} n(a) & = & 1 \\ n(a, t_1, t_2) & = & 1 + n(t_1) + n(t_2) \end{array} \right.$$

$$\left\{ \begin{array}{rcl} ar(a) & = & 0 \\ ar(a) & = & 2 + ar(t_1) + ar(t_2) \end{array} \right.$$

$$\left\{ \begin{array}{rcl} h(a) & = & 1 \\ h(a, t_1, t_2) & = & 1 + max(h(t_1), h(t_2)) \end{array} \right.$$

$$\left\{ \begin{array}{rcl} f(a) & = & 1 \\ f(a, t_1, t_2) & = & f(t_1) + f(t_2) \end{array} \right.$$

2) Montrons que $n(t) = ar(t) + 1$ où n. On procède selon le principe de récurrence induit par la définition.

– Case de base : $t = (a)$. On vérifie facilement que $n(a) = ar(a) - 1$.
– On suppose que $n(t_1) = ar(t_1) + 1$ et $n(t_2) = ar(t_2) + 1$. On en déduit que $n(t) = ar(t_1) + ar(t_2) + 3 = ar(t) + 1$.

3) Montrons que $n(t) = 2.f(t) - 1$.

– Case de base : $t = (a)$. On vérifie facilement que $n(a) = 1 = 2 - 1 = 2.f(a) - 1$.
– On pose $t = (a, t_1, t_2)$ et on suppose que $n(t_1) = 2.f(t_1) - 1$ et $n(t_2) = 2.f(t_2) - 1$. On a alors $n(t) = n(t_1) + n(t_2) + 1 = 2.f(t_1) + 2.f(t_2) - 1 = 2.f(t) - 1$.

EXERCICE 9 Les arbres équilibrés restreignent à 1 au maximum la différrence de hauteurs entre les sous-arbres immédiats (les arbres t_1 et t_2 sont les *sous-arbres immédiats* de (a, t_1, t_2)).

1) On définit inductivement l'ensemble AVL des arbres biniares équilibrés par les deux clause suivantes :

– $\emptyset \in AVL$.
– si $t_1, t_2 \in AVL$ **et** $|h(t_1) - h(t_2)| \leq 1$ alors $(a, t_1, t_2) \in AVL$.

Commentaires : la restriction imposée par la définition à t_1 et t_2 doit se retrouver dans le principe de récurrence induit. L'étape de récurrence du raisonnement sur les AVL se formalise ainsi

$$\forall t_1, t_2 \in AVL. |h(t_1) - h(t_2)| \leq 1 \wedge P(t_1) \wedge P(t_2) \Rightarrow P(a, t_1, t_2)$$

2) On définit la suite $(u_n)_{n \in IN}$

$$\begin{array}{l} u_0 = 0 \\ u_1 = 1 \\ u_{n+2} = u_{n+1} + u_n + 1 \end{array}$$

Montrons, par induction sur l'arbre équilibré t, que $n(t) \geq u_{h(t)}$. Comme la suite (u_n) est définie avec deux cas de base, nous considérons les arbres de hauteurs 0 puis les arbres de hauteur 1 avant d'envisager l'étape de récurrence. Pour être précis, nous raisonnerons donc par récurrence sur la hauter de t.

– Si $h(t) = 0$ alors $t = \emptyset$, on a $n(\emptyset) = 0 = u_0 = u_{h(\emptyset)}$.
– Si $h(t) = 1$ alors, nécessairement, par définition des AVL, $t = (a, \emptyset, \emptyset)$; alors, $n(t) = 1 = u_1 = u_{h(t)}$.

- Si $h(t) > 1$, soient $t_1, t_2 \in AVL$ tels que$|h(t_1) - h(t_2)| \leq 1$, $n(t_1) \geq u_{h(t_1)}$ et $n(t_2) \geq u_{h(t_2)}$. Soit $t = (a, t_1, t_2)$, il faut montrer que $n(t) gequ_{h(t)}$.
  Par définition de n et hypothèses de récurrence, on a que

$$n(t) = n(t_1) + n(t_2) + 1 \geq u_{h(t_1)} + u_{h(t_2)} + 1 \qquad (1)$$

  On raisonne alors par cas selon que $h(t_1) > h(t_2)$, $h(t_1) = h(t_2)$ ou $h(t_2) > h(t_1)$:
  - si $h(t_1) > h(t_2)$ alors, par définition de h et max, $h(t) = h(t_1) + 1$ et, par définition des AVL, $h(t_2) = h(t_1) - 1$. D'où $h(t_1) = h(t) - 1$ et $h(t_2) = h(t) - 2$. En remplaçant $h(t_1)$ et $h(t_2)$ dans (1), on obtient $n(t) \geq u_{h(t)-1} + u_{h(t)-2} + 1 = u_{h(t)}$.
  - si $h(t_2) > h(t_1)$, on raisonne de façon analogue au cas précédent en inversant les rôles de t_1 et t_2.
  - si $h(t_1) = h(t_2)$ alors, par définition de h et max, on obtient $h(t_1) = h(t_2) = h(t) - 1$. En utilisant (1), on a $n(t) \geq u_{h(t)-1} + u_{h(t)-1} + 1$. Comme la suite (u) est croissante, $u_{h(t)-1} \geq u_{h(t)-2}$. D'où $n(t) \geq u_{h(t)-1} + u_{h(t)-1} + 1 \geq u_{h(t)-1} + u_{h(t)-2} + 1 = u_{h(t)}$

EXERCICE 11 On rappelle que le *monoïde libre* A^* est l'ensemble des mots construits sur un alphabet A par l'opération associative de concaténation (notée $\cdot$) avec le mot vide pour élément neutre (noté ε). On confond une lettre de l'aphabet A avec le mot formé de cette seule lettre a, ce qui autorise les écritures $a \cdot w$ et $w \cdot a$ où $a \in A$ et $w \in A^*$.

Dés lors, la définition de l'opération miroir est triviale :

$$\tilde{w} = \begin{cases} \varepsilon & \text{si } w = \varepsilon \\ \tilde{v} \cdot a & \text{si } w = a \cdot v \end{cases}$$

EXERCICE 12 Définition inductive de l'ensemble L des *listes* d'élements de d'un alphabet A :

- $\varepsilon \in L$.
- si $a \in A$ et $x \in L$ alors $(ax) \in L$

La liste (ax) est obtenue par ajout de a devant x. On dit que l'on *préfixe* x avec la lettre a. Il ne faut pas confondre cette opération avec la concaténation de l'exercice précédent. L'expression (xa) n'a pas de sens ! La liste constituée des lettres a_1, a_2 et a_3, dans cet ordre, s'écrit : $(a_1(a_2(a_3\varepsilon)))$. On pourra écrire (a) simplement en place de $(a\varepsilon)$.
On a sur L le principe de récurrence suivant :

- Cas de base : $P(\varepsilon)$.
- Étape de récurrence : $\forall a \in A \forall x \in L.P(x) \Rightarrow P(ax)$

Soit $g : L \times L \to L$ définie par :

$$\begin{cases} g(\varepsilon, y) &= y \\ g((ax), y) &= g(x, (ay)) \end{cases}$$

1) Montrons pas récurrence sur $x \in L$ que $\forall y \in L.g(x, y) \in L$ (ce qui revient à dire que $g(x, y)$ est partout définie sur L.)

- Si $x = \varepsilon$, soit $y \in L$, on a $g(\varepsilon, y) = y \in L$.
- Si $x = (az)$ avec $z \in L$. Il faut montrer que $\forall y \in L.g((az), y) \in L$. Soit donc $y \in L$, montrons $g((az), y) \in L$. C'est-à-dire $g(z, (ay)) \in L$.
  On a , par hypothèse de récurrence $\forall y \in L.g(z, y)$. On a donc, en particulier que $g(z, (ay)) \in L$.

2) $g((a_1), y) = g(\varepsilon, (a_1 y)) = (a_1 y)$

3) Pour être précis, il faut montrer $\forall y \in L.g((a_n(a_{n-1}(..(a_1)..))), y) = g(\varepsilon, (a_1(..(a_{n-1}(a_n y))..)))$, par récurrence sur $n > 0$.

- Si $n = 1$, voir question précédente.
- Supposons donc $\forall y \in L.g((a_n(a_{n-1}(..(a_1)..))), y) = g(\varepsilon, (a_1(..(a_{n-1}(a_n y))..)))$ il faut montrer que $\forall y \in L.g((a_{n+1}(a_n(..(a_1)..))), y) = g(\varepsilon, (a_1(..(a_n(a_{n+1}y))..)))$.
  Par définition, $g((a_{n+1}(a_n(..(a_1)..))), y) = g((a_n(..(a_1)..)), (a_{n+1}y))$. En particularisant y à $(a_{n+1}y)$ dans l'hypothèse de récurrence, on obtient l'égalité recherchée.

4) Si $rev(x) = g(x, \varepsilon)$, en utilisant le résultat de la question précédente, et la définition de g, on obtient $rev((a_1(..(a_n)..))) = (a_n(..(a_1)..))$.