TD 1 : Types abstraits & fonctions recursives

Matthieu Sozeau (matthieu.sozeau@inria.fr)

September 29, 2014

Ce TD porte sur la definition de fonctions recursives et leur verification. Les notes de cours sont disponibles a l'adresse:

http://www.pps.univ-paris-diderot.fr/~sozeau/teaching/MVF-2014-lecture1.pdf

Multiplication

On rappelle le type et la definition des fonctions d'addition et de multiplication sur les entiers de Peano:

$$
\begin{aligned}
+ &: Nat \times Nat \to Nat \\
0 + x &= x \\
s(x_1) + x_2 &= s(x_1 + x_2)
\end{aligned}
$$

$$
\begin{aligned}
\times &: Nat \times Nat \to Nat \\
0 \times x &= 0 \\
s(x_1) \times x_2 &= (x_1 \times x_2) + x_2
\end{aligned}
$$

Exercice 1 (Absorption a droite). *Montrer par induction que* $m \times 0 = 0$.

Solution Par induction sur m. Cas $m = 0$: $0 \times 0 = 0$ par definition. Cas $m = s(m')$: Par induction, $m' \times 0 = 0$. $s(m') \times 0 = 0 + m' \times 0 = 0 + 0 = 0$.

Exercice 2 (Associativite de l'addition). *Monter que* $m + (n + p) = (m + n) + p$.

Solution Par induction sur m:

- Cas $m = 0$, $0 + (n + p) = n + p = (0 + n) + p$ par definition de $+$.
- Cas $m = s(m')$. Hypothese d'induction $m' + (n + p) = m' + n + p$. On doit montrer: $s(m') + (n + p) = s(m') + n + p$. Par simplifications : $s(m' + (n + p)) = s(m' + n + p)$. Par l'hypothese d'induction.

On suppose la commutativite de l'addition demontree $m + n = n + m$, ainsi que $n + 0 = 0$ et $n + s(m) = s(n + m)$.

Exercice 3 (Successeur a droite). *Montrer par induction que* $m \quad s(n) = m + m \quad n$.

Solution Par induction sur m. Cas $m = 0$: $0 \quad s(n) = 0 = 0 + 0 \quad n$ par de nition de et $+$. Cas $m = s(m')$: Par induction, $m' \quad s(n) = m' + m' \quad n$. $s(m') \quad s(n) = s(n) + m' \quad s(n) = s(n + m' \quad s(n)) = s(n + (m' + m' \quad n)$ par induction. $s(m') + s(m') \quad n = s(m' + s(m') \quad n) = s(m' + (n + m' \quad n))$ par de nition.

Par associativite et commutativite de $+$: $s(m' + (n + m' \quad n)) = s(n + (m' + m' \quad n))$

Exercice 4 (Commutativite). *Montrer par induction que* $m \quad n = n \quad m$, *en utilisant les lemmes déjà démontrés.*

Solution Par induction sur m.

- Cas $m = 0$, on doit montrer $0 \quad n = n \quad 0$. $0 \quad n = 0$ par de nition de la multiplication. $n \quad 0 = 0$ par le lemme d'absorption.
- Case $m = s(m')$. Hypothese d'induction: $m' \quad n = n \quad m'$. On doit montrer $s(m') \quad n = n \quad s(m')$ $s(m') \quad n = n + (m' \quad n)$ par de nition. $n \quad s(m') = n + n \quad m'$ par le lemme precedent. Par induction $n + (m' \quad n) = n + (n \quad m')$

Exercice 5 (Distributivite). *Montrer par induction que pour tout* m, n, p, $(m + n) \quad p = m \quad p + n \quad p$.

Solution Par induction sur p.

- Cas $p = 0$, on doit montrer $(m + n) \quad 0 = m \quad 0 + n \quad 0$. En utilisant l'aborsorpition a droite pour le multiplication on se ramene a $0 = 0 + 0$, trivial.
- Case $p = s(p')$. Hypothese d'induction:

  $$(m + n) \quad p' = m \quad p' + n \quad p'$$

  On doit montrer

  $$(m + n) \quad s(p') = m \quad s(p') + n \quad s(p')$$

  .

  Par l'application repetee lemme de multiplication du successeur a droite, on obtient:

  $$(m + n) + (m + n) \quad p' = (m + m \quad p') + (n + n \quad p')$$

.

Par l'hypothese d'induction, il su t de montrer:

$$(m + n) + (m \quad p' + n \quad p') = (m + m \quad p') + (n + n \quad p')$$

.

Trivial par associativite et commutativite de l'addition.

Fonction puissance

Exercice 6 (Type et de nition). *Définir le type de la fonction puissance x^n sur les entiers, écrite* pow x n *ainsi que sa définition.*

Solution

$$\begin{array}{rcl} \texttt{pow} & : & Nat \quad Nat \; ! \; Nat \\ \texttt{pow}\; x\; 0 & = & 1 \\ \texttt{pow}\; x\; s(n) & = & x \quad \texttt{pow}\; x\; n \end{array}$$

Exercice 7 (Calcul). *Déplier la calcul de* pow 2 3*. On utilisera désormais la notation n pour $s^n(0)$ et $n + 1$ pour $s(n)$. Donner directement les résultats intermédiaires de l'appel à la multiplication.*

Solution

$$\begin{array}{rcl} \texttt{pow}\; 2\; 3 & = & 2 \quad \texttt{pow}\; 2\; 2 \\ & = & 2 \quad 2 \quad \texttt{pow}\; 2\; 1 \\ & = & 2 \quad 2 \quad 2 \quad \texttt{pow}\; 2\; 0 \\ & = & 2 \quad 2 \quad 2 \quad 1 \\ & = & 8 \end{array}$$

Exercice 8 (Puissances de 1). *Montrer par induction que* pow 1 n = 1

Solution Par induction sur n:

- Cas $n = 0$: pow 1 0 = 1 par de nition.
- Cas $n = n' + 1$: pow 1 $(n' + 1) = 1$ pow n' Par induction pow 1 $n' = 1$, donc 1 pow $n' = 1 \quad 1 = 1$.

Exercice 9 (Puissance et addition). *Montrer par induction que* pow x $(m + n)$ = pow x m pow x n.

Solution Par induction sur m:

- Cas $m = 0$: $\texttt{pow}\ x\ (0 + n) = \texttt{pow}\ x\ n = 1\ \ \texttt{pow}\ x\ n = \texttt{pow}\ x\ 0\ \ \texttt{pow}\ x\ n$ par de nition.
- Cas $m = m' + 1$: Par induction $\texttt{pow}\ x\ (m' + n) = \texttt{pow}\ x\ m'\ \ \texttt{pow}\ x\ n$.

  $$\texttt{pow}\ x\ (s(m') + n) = \texttt{pow}\ x\ (s(m' + n)) = x\ \ \texttt{pow}\ x\ (m' + n)$$

  par simpli cation. Par l'hypothese d'induction:

  $$x\ \ \texttt{pow}\ x\ (m' + n) = x\ \ \texttt{pow}\ x\ m'\ \ \texttt{pow}\ x\ n$$

  Par associativite et de nition de $\texttt{pow}\ x\ s(m')$:

  $$x\ \ \texttt{pow}\ x\ m'\ \ \texttt{pow}\ x\ n = \texttt{pow}\ x\ s(m')\ \ \texttt{pow}\ x\ n$$

Listes

On rappelle que les listes sont de nies par deux constructeurs:

$$\begin{array}{rcl} [] & : & List[\ ] \\ & : & \quad List[\ ]\ !\ List[\ ] \end{array}$$

Exercice 10 (Longueur d'une liste). *Définir la fonction* `length` l *calculant la longueur d'une liste et* `rev` l *qui inverse la liste.*

Exercice 11 (Longueur d'une concatenation). *Démontrer que* `length` $(l@l')$ = `length` l + `length` l'.

Exercice 12 (Longueur d'une liste renverse). *En utilisant le lemme précédent, démontrer que* `length` (`rev` l) = `length` l.

Exercice 13 (`rev` est involutive). *Démontrer que* `rev` (`rev` l) = l. *Cela nécessite des lemmes sur la concaténation à vous de les découvrir et démontrer.*

Solution Par induction sur l.

- Cas $l = []$: $\texttt{rev}\ (\texttt{rev}\ []) = []$ par de nition.
- Cas $l = x\ \ l'$: On doit montrer $\texttt{rev}\ ((\texttt{rev}\ l')@[x]) = x\ \ l'$. Par induction: $\texttt{rev}\ (\texttt{rev}\ l') = l'$.

  En utilisant $\texttt{rev}\ (l@l') = \texttt{rev}\ l'@\texttt{rev}\ l$, il su t de montrer $\texttt{rev}\ [x]@\texttt{rev}\ (\texttt{rev}\ l') = x\ \ l'$. Par induction et de nition de `rev`,

  $$\texttt{rev}\ [x]@\texttt{rev}\ (\texttt{rev}\ l') = [x]@l' = x\ \ l'$$

Preuve de $\texttt{rev}\ (l@l') = \texttt{rev}\ l'@\texttt{rev}\ l$: Par induction sur l:

- Cas $l = []$: `rev (rev [])` $= []$ par definition.
- Cas $l = x :: m$, par induction: `rev` $(m@l')$ = `rev` l'@`rev` m. On montre `rev` $(x :: m@l')$ = `rev` l'@`rev` $(x :: m)$. Par definition:

  $$\texttt{rev}\ (x :: m@l') = \texttt{rev}\ (m@l')@[x]$$

  Par induction:

  $$\texttt{rev}\ (m@l')@[x] = \texttt{rev}\ l'@\texttt{rev}\ m@[x]$$

  Par definition de `rev`:

  $$\texttt{rev}\ l'@\texttt{rev}\ m@[x] = \texttt{rev}\ l'@\texttt{rev}\ (x :: m)$$

  .

Arbres binaires

Exercice 14 (Definition des arbres)**.** *Donner les constructeurs pour un type de donnée d'arbres binaires $Tree$ dont les feuilles sont étiquetées par des entiers.*

Solution

$$\begin{array}{lcl} \texttt{Leaf} & : & Tree \\ \texttt{Node} & : & Tree \times Tree \rightarrow Tree \end{array}$$

Exercice 15 (Mirroir)**.** *Définir la fonction* `mirror` *sur les arbres binaires qui inverse les branches gauche et droite de chaque noeud de l'arbre.*

Solution

$$\begin{array}{rcl} \texttt{mirror Leaf}(n) & = & \texttt{Leaf}(n) \\ \texttt{mirror }(\texttt{Node}(l,r)) & = & \texttt{Node}(\texttt{mirror } r, \texttt{mirror } l) \end{array}$$

Exercice 16 (Involution de `mirror`)**.** *Montrer que* `mirror` *est involutive.*

Exercice 17 (Elements d'un arbre)**.** *Définir une fonction* `elements` t *de collecte des éléments d'un arbre binaire retournant une liste des éléments aux feuilles de l'arbre dans l'ordre préfixe.*

Solution

$$\begin{array}{rcl} \texttt{elements Leaf}(n) & = & [n] \\ \texttt{elements }(\texttt{Node}(l,r)) & = & \texttt{elements } l\ @\ \texttt{elements } r \end{array}$$

Exercice 18 (Elements et mirroir)**.** *Montrer que pour tout arbre,* `elements` t = `rev` (`elements` (`mirror` t))*. On utilisera le lemme démontré dans l'exercice sur l'involution de* `rev`*.*

Solution Par induction sur t:

- Cas t = $\texttt{Leaf}(n)$: $\texttt{elements Leaf}(n)$ = $[n]$ et $\texttt{rev (elements (mirror Leaf}(n)))$ = $\texttt{rev (elements Leaf}(n))$ = $\texttt{rev}\ [n]$ = $[n]$ par de nition de `mirror`, `elements` et `rev`.

- Cas t = $\texttt{Node}(l, r)$: On doit montrer $\texttt{elements (Node}(l, r))$ = $\texttt{rev (elements (mirror (Node}(l, r))))$ Par induction: $\texttt{elements}\ l$ = $\texttt{rev (elements (mirror}\ l))$ et $\texttt{elements}\ r$ = $\texttt{rev (elements (mirror}\ r))$.

  Par de nition $\texttt{elements (Node}(l, r))$ = $\texttt{elements}\ l$ @ $\texttt{elements}\ r$. On utilise les hypotheses d'induction pour obtenir:

  $\texttt{elements}\ l$ @ $\texttt{elements}\ r$ = $\texttt{rev (elements (mirror}\ l))$ @ $\texttt{rev (elements (mirror}\ r))$

  D'autre part, par de nition de `mirror` et `elements`:

  $$\begin{aligned} \texttt{rev (elements (mirror (Node}(l, r)))) &= \texttt{rev (elements (Node(mirror}\ r, \texttt{mirror}\ l))) \\ &= \texttt{rev (elements (mirror}\ r)\ @\ \texttt{elements (mirror}\ l)) \end{aligned}$$

  On conclut en utilisant le lemme demontre dans l'exercice 13 ($\texttt{rev}\ (l@l')$ = $\texttt{rev}\ l'@\texttt{rev}\ l$).