LOGIQUE ET PREUVE DE PROGRAMMES
IF107

Filière : Informatique, Année : 1
Web: `http://herbrete.vvv.enseirb-matmeca.fr/IF107`

Exercice 1 (Arbres binaires)

Un arbre binaire x sur un alphabet Σ est soit l'arbre vide, noté nil, soit l'arbre $a(g, d)$ de racine $a \in \Sigma$, de fils gauche g et de fils droit d, où g et d sont des arbres binaires.

1. Donner une définition inductive de la fonction n qui à tout arbre binaire x associe le nombre de nœuds dans x.
2. Definir la fonction inductive nbv qui à tout arbre binaire x associe le nombre de sous-arbres vides dans l'arbre x.
3. Puis, prouver par induction que le nombre de sous-arbres vides dans un arbre à $n \geq 0$ nœuds est égal à $n + 1$. Formellement, le prédicat à prouver $P(x)$ est défini pour tout arbre binaire x par $nbv(x) = n(x) + 1$.

♦

Exercice 2 (Un jeu de piles)

Le débute avec une pile de K boîtes. À chaque tour de jeu, on divise une pile de hauteur $n_1 + n_2$ en deux piles de hauteurs n_1 et n_2 telles que $n_1, n_2 \geq 1$. Cette division rapporte $n_1 n_2$ points. Le jeu s'arrête lorsque l'on a K piles de hauteur 1. Le score est alors la somme des points obtenus à chaque division.

Démontrer par induction que le score final est $K(K-1)/2$ quelle que soit la stratégie utilisée.

♦

Exercice 3 (Palindromes)

Un palindrome est un mot qui se lit de la même façon de gauche à droite et de droite à gauche, par exemple : laval, radar, été, ressasser,. . .. Dans la suite, on note L l'ensemble des palindromes sur l'alphabet $\{0, 1\}$.

1. Quels sont les mots palindromes de longueur 2, et ceux de longueur 3 ?
2. En déduire une définition inductive de L et démontrer qu'elle est correcte, c'est à dire que tout palindrome est généré par votre définition, et que tout mot généré est un plaindrome.
3. Donner une expression du nombre de palindromes de longueur n. Quel est le nombre de palindromes de longueur $n = 351$? Généraliser à un alphabet à k symboles.

♦

Exercice 4 (Une preuve fausse)

Un chercheur (fou) est parvenu à montrer le théorème suivant : “tous les chevaux ont la même couleur”. Il a proposé la preuve par induction sur le nombre de chevaux retranscrite ci-dessous.

(Base) Dans un groupe ne contenant qu’un seul cheval, tous les chevaux ont la même couleur.

(Induction) Supposons le théorème vrai pour N chevaux. Considérons un groupe G de $N+1$ chevaux. Observons qu’en retirant un cheval quelconque de G, on obtient un groupe de N chevaux qui ont donc tous la même couleur par hypothèse d’induction. De même, en retirant maintenant *un autre* cheval de G, on obtient a nouveau un groupe de N chevaux qui ont tous la même couleur. On en déduit alors que tous les chevaux ont la même couleur dans le groupe de $N+1$ chevaux.

Cette preuve est évidemment fausse. Où se trouve l’erreur ? ♦

Exercice 5 (Preuve d’algorithme récursif)

La fonction `length` suivante calcule la longueur d’une liste, le nombre d’éléments qu’elle contient. Une liste est soit la liste vide `[]`, soit une liste non vide `hd::tl` constituée d’un élément de tête `hd` et d’une liste `tl`.

```
length(l) = match l with
              [] -> 0
            | hd::tl -> length(tl)+1
```

Prouver la correction de la fonction `length` par induction. ♦

Exercice 6 (Preuve d’algorithme récursif (2))

La fonction `find` suivante indique si un élément `x` apparient à une liste. Une liste est soit la liste vide `[]`, soit une liste non vide `hd::tl` constituée d’un élément de tête `hd` et d’une liste `tl`.

```
find(x,l) = match l with
              [] -> false
            | hd::tl -> if (hd=x) then
                          true
                        else
                          find(x,tl)
```

Prouver la correction de la fonction `find` par induction. ♦

Exercice 7 (TAD et induction)

Une file FIFO est une structure de données qui stocke une séquence d'éléments, et telle que l'ajout se fait en fin de séquence et le retrait se fait en tête de séquence [1]. L'ensemble des files (FIFO) est inductivement défini par :

(Base) `empty` est une file

(Induction) pour toute file `f` et pour tout élément `x`, `push(f,x)` est une file

Par exemple, $\texttt{push}(\texttt{push}(\texttt{empty}, \texttt{a}), \texttt{b})$ est la file FIFO correspondant à la séquence d'éléments ab. La fonction θ ci-dessous associe à une file `f` la séquence correspondante.

$$\begin{aligned} \theta(\texttt{empty}) &= \varepsilon \\ \theta(\texttt{push}(\texttt{f}, \texttt{x})) &= \theta(\texttt{f}) \cdot x \end{aligned}$$

La séquence vide est notée ε et "$\cdot$" représente la concaténation. Ainsi :

$$\begin{aligned} \theta(\texttt{push}(\texttt{push}(\texttt{empty}, \texttt{a}), \texttt{b})) &= \theta(\texttt{push}(\texttt{empty}, \texttt{a})) \cdot b \\ &= \theta(\texttt{empty}) \cdot a \cdot b \\ &= \varepsilon \cdot a \cdot b \\ &= ab \end{aligned}$$

1. Par quelle file la séquence abc est-elle représentée ? Plus généralement, par quelle file la séquence $x_1 \cdots x_n$ est-elle représentée ?
2. Donner une définition inductive de `top(f)` qui retourne l'élément de tête d'une file `f` supposée non vide. Démontrez par induction que votre définition est correcte, c'est à dire que $\texttt{top}(\texttt{f}) = x_1$ si et seulement si $\theta(\texttt{f}) = x_1 \cdots x_n$ avec $n \geq 1$.
3. Donnez une définition inductive de `pop(f)` qui retourne la file `f` privée de son élément de tête, `f` étant supposée non vide. Démontrez par induction que votre définition est correcte, c'est à dire que $\theta(\texttt{pop}(\texttt{f})) = x_2 \cdots x_n$ si et seulement si $\theta(\texttt{f}) = x_1 x_2 \cdots x_n$ avec $n \geq 1$ (remarque : la séquence $x_2 \cdots x_n$ peut être vide).
4. L'opération `reverse(f)`, qui inverse le contenu d'une file `f`, est définie par :

```
reverse(empty) = empty
reverse(f)     = push( reverse( pop(f) ), top(f) )    [f != empty]
```

   Démontrez par induction que la définition de `reverse` est correcte : $\theta(\texttt{reverse}(f)) = x_n \cdots x_1$ si et seulement si $\theta(\texttt{f}) = x_1 \cdots x_n$ (la séquence vide est obtenue pour $n = 0$).

♦

1. Une file FIFO fonctionne exactement comme une file d'attente à un guichet.