Devoir “Maison”

3 mars 1998

On considère le contexte suivant (dans le lambda calcul simplement typé avec un type de base : ι) :

$$
\begin{aligned}
0 &: \iota && (1)\\
S &: \iota \to \iota && (2)\\
nil &: \iota && (3)\\
cons &: \iota \to \iota \to \iota && (4)
\end{aligned}
$$

Il s’agit de représenter des listes d’entiers naturels. Par exemple, la liste `(1 2 3)` est représentée par le λ−terme

```
cons(S(0),cons(S(S(0)),cons(S(S(S(0))),nil)))
```

1

Les fonctions `tail` et `append` sont-elles β−définissables ? Pourquoi ?

1.1 Rappel

$$
\begin{aligned}
tail(nil) &= nil && (5)\\
tail((x_1\ x_2\ \ldots\ x_n)) &= (x_2\ \ldots\ x_n) && (6)\\
append((x_1\ x_2\ \ldots\ x_n),(y_1\ y_2\ \ldots\ y_p)) &= (x_1\ x_2\ \ldots\ x_n\ y_1\ y_2\ \ldots\ y_p) && (7)
\end{aligned}
$$

2

Adapter la démarche vue pour les entiers (théorie des types simples avec axiomes) pour donner un cadre permettant de définir `tail` et `append`

2.1 Remarque

Le type ι permet de représenter entiers et listes. Comme dans le Dowek, on définit des prédicats de types

$$\mathbb{N} = \lambda n \forall (X : \iota \to o)\ X(0) \wedge (\forall (i : \iota) X(i) \Rightarrow X(S(i))) \Rightarrow X(n)$$

Dont on déduit un principe de récurrence (Dowek 5.3.9).

Vous devrez certainement faire la même démarche avec un prédicat $\mathbb{L} : \iota \to o$ ($\mathbb{L}(x)$ signifiant: “x est une liste d’entiers”.)