

Entiers de Church

Jean-Baptiste Rouquier

Novembre 2005

Un TP court, prévoir une demi-heure à une heure. Une bonne gymnastique fonctionnelle pour être sûr d'avoir vu pire que ce que l'on trouvera en concours. Un bon moyen aussi de voir comment on réagit face à des concepts déroutants.

On appelle *entier de Church*, et l'on note \widehat{n} , la *fonction* qui à f associe f^n , où f^n désigne la fonction f itérée n fois. Par exemple $\widehat{2}(\cos)(\frac{\pi}{2}) = \cos(\cos(\frac{\pi}{2})) = \cos(0) = 1$. En d'autres termes,

```
let rec church_int n f x =  
  if n=0 then x  
  else church_int (n-1) f (f x)
```

Ainsi `church_int n` renvoie \widehat{n} . Si x est de type $'a$, f est de type $'a \rightarrow 'a$, donc \widehat{n} est de type $('a \rightarrow 'a) \rightarrow ('a \rightarrow 'a)$, ce que l'on notera en raccourci par le type $'a \widehat{\text{int}}$.

Question 0.1. Écrire directement (sans utiliser `church_int`) les entiers de Church $\widehat{0}$, $\widehat{1}$ et $\widehat{2}$.

Question 0.2. Écrire `eval` : $\widehat{\text{int}} \rightarrow \text{int}$ qui à l'entier de Church \widehat{n} associe l'entier n .

Question 0.3. Écrire `succ_ci` : $\widehat{\text{int}} \rightarrow \widehat{\text{int}}$ qui à l'entier de Church \widehat{n} associe l'entier de Church $\widehat{n+1}$.

Question 0.4. Écrire les fonctions `add`, `mult` et `exp`, de type $\widehat{\text{int}} \rightarrow \widehat{\text{int}} \rightarrow \widehat{\text{int}}$.

Question 0.5. Tout tester. Il est normal que le vérificateur de type refuse

```
let dix_huit = add neuf neuf in  
dix_huit succ 0,  
dix_huit sqrt 10.
```

Question 0.6 (bonus). Chaque fonction ne prend qu'une ligne (sans avoir besoin de tasser).