Département d'Informatique
LICENCE 2ème année
Mardi 22 janvier 2008

Corrigé du contrôle terminal de Programmation Fonctionnelle

I Récursivité et schémas de programmes

1.
```
let rec ajoute_occ repartition l = match(repartition) with
      [] -> [(l,1)]
 |    (l',occ)::r -> if l=l' then (l,occ+1)::r
                             else (l',occ)::(ajoute_occ r l);;
```

2.
```
let rec repartition_occ t =
      List.fold_left ajoute_occ []
         (List.fold_left (@) [] t);;
```

3.
```
let repartition_freq r =
   let tot = List.fold_left
       (function sum -> function (l,occ) -> sum+occ) 0 r in
          List.map (function (l,occ) ->
            (l,float_of_int(occ)/.float_of_int(tot))) r;;
```

4.
```
let trouve_cle texte =
      let e = lettre_E (reaprtition_freq (calcul_occ [] texte)) in
         (Char.code 'E') - (Char.code e);;
```

5.
```
let decale cle l =
      Char.chr ((Char.code l) + cle);;

let dechiffre texte cle =
      List.map (function m -> List.map (decale cle) m) texte;;
```

II Lambda-calcul

1. $\lambda f.\lambda g.\lambda x.((f\ x)\ (g\ x))_{\curlywedge}\lambda x.\lambda y.x\ \lambda x.\lambda y.x$
$\xrightarrow[\beta]{} \lambda g.\lambda x.((\lambda x.\lambda y.x\ x)\ (g\ x))_{\curlywedge}\lambda x.\lambda y.x$
$\xrightarrow[\beta]{} \lambda x.(\lambda x.\lambda y.x_{\curlywedge}x\ (\lambda x.\lambda y.x\ x))$
$\xrightarrow[\beta]{} \lambda x.(\lambda y.x_{\curlywedge}(\lambda x.\lambda y.x\ x))$
$\xrightarrow[\beta]{} \lambda x.x$

2. $\lambda f.\lambda g.\lambda x.((f\ x)\ (g\ x))_{\curlywedge}\lambda x.\lambda a.(x\ a)\ \lambda a.\lambda x.(a\ x)$

$$
\begin{aligned}
&\xrightarrow[\beta]{} \lambda g.\lambda x.(\lambda x.\lambda a.(x\ a)\ x\ (g\ x))_{\curlywedge}\lambda a.\lambda x.(a\ x)\\
&\xrightarrow[\beta]{} \lambda x.(\lambda x.\lambda a.(x\ a)_{\curlywedge}x\ (\lambda a.\lambda x.(a\ x)\ x))\\
&\xrightarrow[\beta]{} \lambda x.(\lambda a.(x\ a)_{\curlywedge}(\lambda a.\lambda x.(a\ x)\ x))\\
&\xrightarrow[\beta]{} \lambda x.(x\ (\lambda a.\lambda x.(a\ x)_{\curlywedge}x))\\
&\xrightarrow[\alpha]{} \lambda x.(x\ (\lambda a.\lambda b.(a\ b)_{\curlywedge}x))\\
&\xrightarrow[\beta]{} \lambda x.(x\ \lambda b.(x\ b))
\end{aligned}
$$

3. $\lambda f.\lambda g.\lambda x.((f\ x)\ (g\ x))_{\curlywedge}\lambda a.\lambda x.(a\ x)\ \lambda x.\lambda a.(x\ a)$

$$
\begin{aligned}
&\xrightarrow[\beta]{} \lambda g.\lambda x.(\lambda a.\lambda x.(a\ x)\ x\ (g\ x))_{\curlywedge}\lambda x.\lambda a.(x\ a)\\
&\xrightarrow[\beta]{} \lambda x.(\lambda a.\lambda x.(a\ x)_{\curlywedge}x\ (\lambda x.\lambda a.(x\ a)\ x))\\
&\xrightarrow[\alpha]{} \lambda x.(\lambda a.\lambda b.(a\ b)_{\curlywedge}x\ (\lambda x.\lambda a.(x\ a)\ x))\\
&\xrightarrow[\beta]{} \lambda x.(\lambda b.(x\ b)_{\curlywedge}(\lambda x.\lambda a.(x\ a)\ x))\\
&\xrightarrow[\beta]{} \lambda x.(x\ (\lambda x.\lambda a.(x\ a)_{\curlywedge}x))\\
&\xrightarrow[\beta]{} \lambda x.(x\ \lambda a.(x\ a))
\end{aligned}
$$