—— INFORMATIQUE — MIAS/SM 1 — TP 4 ——

Encore quelques récursivités

Exercice 1 : Points fixes de fonctions

Un nombre x est appelé *point fixe* d'une fonction f s'il satisfait l'équation $f(x) = x$. Pour certaines fonctions (la fonction cosinus par exemple), il est possible de déterminer un point fixe à partir d'une estimation initiale x_0, en calculant la suite x_0,$f(x_0)$,$f(f(x_0)),\ldots$,$f(\ldots(f(x_0))\ldots),\ldots$, jusqu'à ce que la valeur n'évolue presque plus, c'est-à-dire que la valeur absolue de la différence entre deux termes consécutifs de la suite soit inférieure à une précision donnée.

En utilisant cette idée, concevez une fonction `point_fixe` de type `(float -> float) -> float -> float -> float`, qui prend pour arguments une fonction, une estimation de départ et une précision, et qui rend une valeur approchée d'un point fixe de la fonction. Cette fonction sera de la forme suivante :

```
let point_fixe = function f -> function x -> function precis ->
          let rec cherche =  RES-1-1
          in  cherche x ;;
```

où `cherche` déterminera les termes successifs de la suite jusqu'à la valeur cherchée (on devra utiliser la fonction Caml `abs_float`). Plus précisément, la fonction `cherche` prend comme paramètre un terme de la suite, calcule le terme suivant et, selon la différence entre les deux termes consécutifs, retourne le dernier ou passe au terme suivant.

Tester votre fonction `point_fixe`, en évaluant l'expression `(point_fixe cos 1. 0.001)` pour obtenir le point fixe de la fonction cosinus.

Exercice 2 : Résolution dichotomique d'une équation

Introduction

La méthode dichotomique est une technique simple et puissante pour résoudre une équation de la forme $f(x) = 0$ où f est une fonction continue. En effet, on sait que, si a et b sont deux réels tels que $f(a) < 0 < f(b)$, alors f admet au moins un zéro entre a et b.

La méthode que nous considérons ici consiste à choisir deux valeurs initiales a et b, telles que $f(a) < 0 < f(b)$, puis à réduire l'intervalle d'extrémités a et b de la façon suivante : soit $m = (a + b)/2$,

- si $f(m) < 0$, alors on a $f(m) < 0 < f(b)$ et f admet un zéro dans l'intervalle d'extrémités m et b ;
- si $f(m) > 0$, alors on a $f(a) < 0 < f(m)$ et f admet un zéro dans l'intervalle d'extrémités a et m ;

.../...

– si $f(m) = 0$, on a résolu notre problème.

Dans les deux premiers cas, l'intervalle dans lequel on recherche une solution a été réduit de moitié. Le processus s'arrête lorsque l'on est arrivé à un intervalle assez petit, c'est-à-dire dont la taille est inférieure à une précision donnée. On rend alors la valeur médiane de cet intervalle.

Questions

1. Écrire une fonction `cherche` mettant en œuvre la stratégie décrite ci-dessus, en complétant la fonction ci-dessous :

```
let rec cherche = function f -> function precis ->
           function a -> function b ->
                let m =   [RES-2-1]
                in  if    [RES-2-2]       (* condition d'arrêt *)
                    then  [RES-2-3]       (* valeur terminale *)
                    else if    [RES-2-4]  (* appels récursifs *)
                         then  [RES-2-5]
                         else  [RES-2-6];;
```

   Tester cette fonction pour des équations dont les solutions sont connues. Donner le résultat de ces tests.

2. La fonction précédente ne fonctionne que lorsque les paramètres a et b vérifient la condition $f(a) < 0 < f(b)$.
   Écrire une fonction `dichotomie` qui prend les mêmes arguments que la fonction `cherche` et détermine dans quel ordre on doit placer les deux paramètres a et b pour que cette condition soit vérifiée (ou signale une erreur grâce à `failwith` si ce n'est pas possible). Ne pas hésiter, comme dans la fonction précédente, à utiliser des variables locales afin d'éviter d'effectuer plusieurs fois le même appel de fonction.

3. **Application :**
   – trouver une solution de $x^2 = 2$;
   – trouver une solution de $\cos x = x$ dans l'intervalle $[0,\pi]$, et comparer cette valeur à celle trouvée dans l'exercice 1.