Programmation fonctionnelle 2023-2024

Feuille 3bis : Exercices complémentaires - Dérivation formelle

Rappels :

$$\begin{array}{ll} \frac{dc}{dx} & = 0, \\ \frac{dx}{dx} & = 1, \\ \frac{d(u+v)}{dx} & = \frac{du}{dx} + \frac{dv}{dx}, \\ \frac{d(uv)}{dx} & = u\frac{dv}{dx} + v\frac{du}{dx}, \\ \frac{d(exp(u))}{dx} & = \frac{du}{dx}exp(u). \end{array}$$

On représente des expressions arithmétiques utilisant les opérations $+, -, \times, exp$ par le type suivant :

```
type expr =
          Var of string
        | Number of float
        | Plus of expr * expr
        | Minus of expr * expr
        | Mult of expr * expr
        | Exp of expr
```

Ainsi, une variable x est représentée par l'expression `OCaML` `Var "x"` et $3x^2 + 2x + 1$ par

```
Plus (Mult (Number 3., Mult (Var "x", Var "x")),
  Plus (Mult (Number 2., Var "x"), Number 1.))
```

Exercice 3.1 1. Définir deux variables `vx` et `vy` contenant respectivement des variables x et y.

2. Définir la variable `e1` contenant l'expression $2x + 1$.
3. Définir la variable `e2` contenant l'expression $3x^2 + 2x + 1$.
4. Implémenter une fonction `derivee var expr` par une traduction directe des cinq règles ci-dessus, i.e. par un simple filtrage avec cinq cas distincts.
   Exemple d'utilisation :

```
utop[83]> vx;;
- : expr = Var "x"
utop[84]> e1;;
- : expr = Plus (Mult (Number 2., Var "x"), Number 1.)
utop[85]> derivee vx e1;;
- : expr =
Plus (Plus (Mult (Number 2., Number 1.), Mult (Number 0., Var "x")),
 Number 0.)
utop[86]> e2;;
- : expr =
Plus (Mult (Number 3., Mult (Var "x", Var "x")),
 Plus (Mult (Number 2., Var "x"), Number 1.))
utop[87]> derivee vx e2;;
- : expr =
Plus
 (Plus
   (Mult (Number 3.,
```

```
      Plus (Mult (Var "x", Number 1.), Mult (Number 1., Var "x"))),
    Mult (Number 0., Mult (Var "x", Var "x"))),
  Plus (Plus (Mult (Number 2., Number 1.), Mult (Number 0., Var "x")),
   Number 0.))
```

5. Écrire la fonction `derivee_n var expr n` qui retourne une expression correspondant à la dérivée f*n*ème de `expr`.
   Exemples :

```
utop[91]> derivee_n vx e2 2;;
- : expr =
Plus
 (Plus
   (Plus
     (Mult (Number 3.,
       Plus (Plus (Mult (Var "x", Number 0.), Mult (Number 1., Number 1.)),
        Plus (Mult (Number 1., Number 1.), Mult (Number 0., Var "x")))),
     Mult (Number 0.,
      Plus (Mult (Var "x", Number 1.), Mult (Number 1., Var "x")))),
   Plus
    (Mult (Number 0.,
      Plus (Mult (Var "x", Number 1.), Mult (Number 1., Var "x"))),
     Mult (Number 0., Mult (Var "x", Var "x")))),
  Plus
   (Plus (Plus (Mult (Number 2., Number 0.), Mult (Number 0., Number 1.)),
     Plus (Mult (Number 0., Number 1.), Mult (Number 0., Var "x"))),
    Number 0.))
    utop[92]>
```

Exercice 3.2 Les expressions auraient bien besoin d'être simplifiées. Proposer des pistes pour simplifier les expressions.