
Programmation fonctionnelle

Énoncé du Travail

Année 2014-2015

1 Dérivation symbolique

Le calcul symbolique est un procédé de transformation d'expressions mathématiques. On entend par dérivation symbolique d'une expression analytique, le calcul de l'expression analytique exacte correspondant à la dérivée de l'expression donnée en entrée.

1.1 Fonctions de base

Dans ce travail, on se contentera d'un sous-ensemble restreint de fonctions de base permettant de décrire les expressions à traiter.

Les fonctions seront données selon leur forme **Scheme** habituelle :

- Addition : $a_0 + a_1 + a_2 + \dots$
(+ a0 a1 a2 ...)
- Multiplication : $a_0 \cdot a_1 \cdot a_2 \cdot \dots$
(* a0 a1 a2 ...)
- Soustraction : $a_0 - (a_1 + a_2 + \dots)$
(- a0 a1 a2 ...)
- Division : $\frac{a_0}{a_1 \cdot a_2 \cdot \dots}$
(/ a0 a1 a2 ...)
- Exponentielle : a^b
(expt a b)
- Logarithme : $\ln(x)$
(log x)
- Sinus : $\sin(x)$
(sin x)
- Cosinus : $\cos(x)$
(cos x)
- Tangente : $\tan(x)$
(tan x)

1.2 Simplifications

Par soucis de lisibilité, de compréhension et de concision, il est souhaitable de simplifier le plus possible l'expression correspondant au résultat.

Il est par exemple évident que malgré l'exactitude mathématique d'une solution qui serait $(\ast (\cos x) (\sin x) 0 x)$, on préférerait avoir comme solution directement 0.

De la même manière, 17 sera préféré à $(+ 5 (\sin 0) 12)$.

La difficulté sera de choisir un ensemble de règles de simplifications pertinentes pour le travail qui nous occupe.

1.3 Exemples

Remarque : Vos résultats peuvent bien entendu être différents mais doivent être mathématiquement équivalents.

```
(deriv '(* 3 x) 'x)
⇒ 3
```

```
(deriv '(+ 3 (log x) x) 'x)
⇒ (+ (/ 1 x) 1)
```

```
(deriv '(expt a b) 'a)
⇒ (* b (expt a (- b 1)))
```

```
(deriv '(expt a b) 'b)
⇒ (* (expt a b) (log a))
```

```
(deriv '(+ 1 (expt (cos x) 2) 'x)
⇒ (* -2 (cos x) (sin x))
```

```
(deriv '(/ (cos (* 5 x)) (* 3 (expt x a))) 'x)
⇒ (- (/ (* -5 (sin (* 5 x))) (* 3 (expt x a)))
      (/ (* a (cos (* 5 x))) (* 3 (expt x (+ a 1)))))
```

2 Travail

Il est demandé d'écrire une fonction Scheme (`deriv exp var`), `exp` une expression symbolique à dériver et `var` la variable par rapport à laquelle effectuer la dérivation. La fonction devra renvoyer comme résultat votre meilleure simplification possible de la dérivation symbolique de l'expression donnée en argument.

Le nombre, l'efficacité et la pertinence des différentes simplifications seront pris en compte lors de l'évaluation.

Remarques importantes :

- Ce travail est à réaliser par groupe de deux étudiants.
- Votre travail doit pouvoir fonctionner sous Linux (*DrRacket*).
(Les machines ms800 seront utilisées comme référence.)
- Ce travail doit être envoyé sous la forme d'une archive tar compressée nommée `projet_scheme_2015_nom1_nom2.tar.gz`.
- Votre archive doit contenir au minimum un rapport au format PDF et le programme Scheme réalisé.
- Votre rapport expliquera tous vos choix non triviaux ainsi que tout ce vous jugerez utile à la correction.
- Votre travail doit être envoyé avant le 19 avril 2015 à 23h59 à l'adresse `gribomont@montefiore.ulg.ac.be`. L'email doit être envoyé depuis votre adresse ULg, avec [INF0-0054 Projet 2015] comme sujet et avec l'adresse de votre co-auteur en copie.
- N'oubliez pas de tester votre programme avec d'autres exemples !
- Au maximum un groupe de 3 personnes sera toléré dans le cas d'un nombre impair d'étudiants.
- La liste complète des groupes (centralisée par un délégué) doit être envoyée avant le 25/02/2014 à 23h59 à l'adresse `gribomont@montefiore.ulg.ac.be`.
- Toute personne non présente sur cette liste (sans raison valable) obtiendra une note de 0 au projet.