

# Projet : dérivation formelle de fonctions

A rendre lors du contrôle continu de la matière “Ingénierie de la preuve”

*Ce projet est à réaliser individuellement. Il est indispensable que vous travailliez de manière autonome et que vous citiez avec précision vos sources si vous utilisez des ressources externes (trouvées sur le web ou ailleurs<sup>1</sup>).*

## 1 Travail demandé

1. Un rapport au format pdf décrivant le projet, les choix techniques effectués et les réalisations.
2. Un développement en Coq comprenant les structures de données, les définitions de fonctions et les preuves des théorèmes demandés.

## 2 Structure de données

On va formaliser dans Coq les fonctions de  $\mathbb{R}$  dans  $\mathbb{R}$ . L'objectif est de construire une opération de dérivation formelle sur ce type des fonctions.

On utilise la bibliothèque standard sur les réels :

`Require Export Reals.`

L'ensemble des théorèmes sur les nombres réels se trouvent dans le répertoire `theories/Reals` de la bibliothèque standard de Coq. On dispose de quelques outils de preuve automatique utiles sur les réels, notamment la tactique `field`, qui permet de prouver des égalités (et donc de faire de la substitution d'expression grâce à la tactique `replace`) et la tactique `fourier` qui permet de démontrer certaines inégalités. La tactique `discrR` pourrait aussi être utile.

**Question 1** Expliquer comment les nombres réels sont définis dans Coq (étudier les fichiers `Rdefinitions` et `Raxioms`). Quels sont les moyens de calcul sur les réels? Justifier pourquoi la commande `Eval compute in (3+3)%R.` retourne `(3+3)%R`.

**Question 2** Proposer un type inductif `fonction` permettant de représenter les fonctions (toutes dérivables en tout point de  $\mathbb{R}$ ) qui sont obtenues à partir de l'identité, des fonctions constantes, de l'exponentielle et des fonctions trigonométriques (sinus et cosinus) et par l'application des opérations d'addition, soustraction, produit, composition de fonctions déjà obtenues.

`Inductive fonction : Set := ...`

**Question 3** Comment définir en Coq les fonctions suivantes  $x \mapsto \sin(x) + \exp(x) * \cos(x)$  et  $x \mapsto x * \sin(x)$  avec le type de données de la question précédente?

**Question 4** Programmer une fonction d'interprétation `interp : fonction -> (R -> R)` interprétant une expression de type `fonction` comme une fonction de  $\mathbb{R}$  dans  $\mathbb{R}$ .

## 3 Fonctions infiniment dérivables en tout point

**Question 5** Définir une fonction `deriv : fonction -> fonction` de dérivation formelle. Pour cela on utilisera les formules usuelles de dérivation pour les fonctions sans jamais interpréter les fonctions considérées.

**Question 6** Montrer la correction de la fonction `deriv`, à savoir

$$\forall f : \text{fonction}, \forall x : \mathbb{R}, \text{dérivable\_pt\_lim} (\text{interp } f) x (\text{interp } (\text{deriv } f) x)$$

Pour cela, on cherchera à utiliser les théorèmes déjà existants de la bibliothèque sur les réels de Coq. On pourra trouver tous ceux traitant de la dérivée d'une fonction en un point au moyen de la commande `SearchPattern (dérivable_pt_lim - - -)`.

<sup>1</sup>Par exemple : ce sujet de projet est librement inspiré d'un sujet posé au MPRI en 2005/2006.