
Titre : Différentiation algorithmique et preuve formelle

Sujet :

La différentiation algorithmique est une transformation de programmes qui prend un code source implémentant une fonction F et qui rend un code transformé implémentant des dérivées de F. Calculer ces dérivées est crucial pour tous problèmes d'optimisation, d'études de sensibilité, de problèmes inverses. Il existe deux modes principaux --tangent et adjoint-- qui produisent des dérivées identiques pour des coûts calcul différents. L'objectif de ce stage est d'étudier formellement ces deux modes et à terme de fournir un cadre formel pour la preuve de leur équivalence.

Pour ce faire, on partira d'un langage restreint et des règles de réécriture qui définissent les modestangent et adjoint pour proposer une architecture de preuve extensible. Ce travail se fera dans l'assistant de preuve Coq.

Pour ce stage, on ne présupposera pas de connaissance particulière mais une forte motivation et un intérêtpour le calcul scientifique, la sémantique des langages de programmation, et la preuve formelle sont les bienvenus.

Encadrant: Laurent Hascoët (Laurent.Hascoet@inria.fr)
Laurent Théry (Laurent.Thery@inria.fr)

Title : Algorithmic differentiation and formal proof

Description :

Algorithmic differentiation is a program transformation that takes a source code that implements a functionF and returns a modified code implementing some derivatives of F. Computing these derivatives is crucialfor optimization problems, sensitivity analyses, or inverse problems. There exist two main modes --forward accumulation and backward accumulation-- that produce identical derivatives but at a different computing cost. The goal of this internship is to study these two modes formally and build a formal framework where the equivalence between these two modes can be proved.

We will start with a restricted language and some rewriting rules that express the forward and the backward accumulation modes to then propose a proof architecture that is extensible. This work will be done in the Coq proof assistant.

For this internship, no specific knowledge is required but a strong motivation and a genuine interest for scientific computing, semantics of programming languages and formal proof are welcome.

Contact: Laurent Hascoët (Laurent.Hascoet@inria.fr)
Laurent Théry (Laurent.Thery@inria.fr)