Application BASTRI
Fiches Equipes

GALLIUM (SR0093AR)

Langages de programmation, types, compilation et preuves
CRISTAL (SR0224GR) GALLIUM CAMBIUM (SR0881NR)

Statut: Terminée

Responsable : Xavier Leroy

Mots-clés de "A - Thèmes de recherche en Sciences du numérique - 2024" : *Aucun mot-clé.*

Mots-clés de "B - Autres sciences et domaines d'application - 2024" : *Aucun mot-clé.*

Domaine : Algorithmique, programmation, logiciels et architectures
Thème : Preuves et vérification

Période : 01/05/2006 -> 31/07/2019
Dates d'évaluation : 22/03/2011 , 17/03/2015 , 20/03/2019

Etablissement(s) de rattachement : <sans>
Laboratoire(s) partenaire(s) : <sans UMR>

CRI : Centre Inria de Paris
Localisation : Centre de recherche Inria de Paris
Code structure Inria : 021071-1

Numéro RNSR : 200618331D
N° de structure Inria: SR0093AR

Présentation

Les travaux de recherche de l'équipe-projet GALLIUM portent sur la conception, la formalisation et l'implémentation de langages et systèmes de programmation. Notre objectif est d'améliorer la fiabilité (sûreté de fonctionnement et sécurité) des logiciels en utilisant :

- des langages de programmation de plus haut niveau, plus sûrs et plus expressifs, basés sur le paradigme de la programmation fonctionnelle ;
- la détection automatique d'erreurs de programmation à l'aide de systèmes de types et autres analyses statiques ;
- une meilleure intégration de la programmation et des méthodes formelles, en particulier la preuve sur machine de programmes.

Axes de recherche

- Systèmes de types, inférence de types et analyses statiques reliées. Le typage statique (la vérification d'un système de types pendant la compilation) est un outil très efficace pour détecter très tôt de nombreuses erreurs de programmation. De plus, les types aident à structurer non seulement les programmes eux-mêmes mais aussi les designs de langages de programmation. Parmi nos travaux en cours sur les systèmes de types, citons: le typage fin de structure de données, reflétant des invariants structurels à l'aide de types algébriques généralisés; l'inférence partielle de type pour des systèmes de types très expressifs, incluant du polymorphisme de première classe et d'ordre supérieur; et les systèmes de types pour les mécanismes de programmation à grande échelle (modules, classes, mixins, etc).

- Vérification formelle de compilateurs et autres outils de programmation. Lorsqu'on applique les méthodes formelles à un logiciel critique, comme exigé par les plus hauts niveaux de certification, il devient nécessaire de certifier formellement les outils qui servent à produire ce logiciel: compilateurs, analyseurs statiques, etc. Nous travaillons actuellement sur le développement et la vérification formelle, à l'aide de l'assistant de preuve Coq, d'un compilateur optimisant pour le langage C. La preuve Coq garantit que le code assembleur produit par ce compilateur a le même comportement sémantique que le programme source. Nous nous intéressons également aux spécifications et vérifications sur machine d'autres outils de programmation, comme les systèmes de types.

Contact

- **Responsable :** Xavier Leroy
- **Tél :** 01.39.63.55.61
- **Secrétariat Tél :** 01.39.63.52.07

En savoir plus

- Site de l'équipe
- Site sur inria.fr
- Derniers Rapports d'Activité : 2016 , 2017 , 2018

Documents sur la structure

- Intranet
- Privés

Décisions

- 4989 (01/06/2006) : création
- 8339 (19/01/2012) : prolongation
- 11317 (14/12/2015) : prolongation
- 13730 (15/07/2019) : prolongation
- 13803 (12/08/2019) : fermeture

Localisation

- **Adresse postale :** Centre Inria de Paris 48, rue Barrault CS 61534 75647 PARIS CEDEX
- **Coordonnées GPS :** 48.826, 2.346

- Conception et implémentation de langages de programmation de haut niveau.
  Nous concevons et réalisons le langage Objective Caml et son environnement de programmation. Objective Caml est un langage de haut niveau, statiquement typé, qui combine les styles de programmation fonctionnel, impératif, à objets et modulaire. Nos réflexions actuelles autour de Caml et d'un éventuel successeur à ce langage comprennent l'étude de mécanismes unifiés pour la modularisation et la paramétrisation de fragments de code, ainsi que pour l'encapsulation et la délimitation des traits impératifs, ainsi que l'application de résultats récents en compilation et environnements d'exécution.

- Conception conjointe du logiciel et de sa preuve.
  La pratique des méthodes formelles dans l'industrie consiste à vérifier formellement du code déjà écrit, sans avoir le droit de le modifier. Nous défendons et utilisons une approche différente, où le code et ses preuves de correction sont développés en parallèle. Nous étudions divers moyens de faciliter cette approche de conception conjointe logiciel/preuve: enrichir les capacités de programmation de systèmes logiques tels que Coq; étendre des langages de programmation comme Caml avec des assertions logiques; et contribuer à des environnements unifiés pour la programmation et la preuve. Nous réalisons également le démonstrateur automatique Zenon, qui est une brique de base pour de tels environnements.

Logiciels

- Objective Caml

Relations industrielles et internationales

- ANR, action de recherche amont Compcert: compilation certifiée.
- Intel Corporation: implémentation du langage reFLect.
- Projet européen IST EDOS: environnement pour le développement et la distribution de logiciel libre.
- Membres industriels du Consortium Caml.