Offer #2024-07950

Une sémantique de flots dans Coq pour les structures de contrôle

The offer description below is in French

Contract type : Fixed-term contract

Level of qualifications required : Graduate degree or equivalent

Fonction : Temporary scientific engineer

Level of experience : Recently graduated

Context

Contexte et atouts du poste

Le « Model-Based Design » est une approche moderne pour le développement de logiciels embarqués. L'idée est que les dessins dits schémas-blocs, que les ingénieurs utilisent pour modéliser un système et son environnement, peut être des spécifications précises et exécutables. De plus, il est même possible de transformer des parties vers du code de bas niveau pour une cible donnée. Dans le projet Vélus, nous nous donnons comme ambition de formaliser un langage de schémas-blocs et ses algorithmes de compilation dans un assistant de preuve, et de démontrer par preuve formelle que ces derniers préservent la sémantique du langage source jusqu'au code généré.

Principales activités

Travailler en équipe pour bien comprendre la problématique et se mettre d'accord sur les modèles sémantiques et les modifications requises au compilateur. En utilisant l'assistant de preuve Coq, ajouter de nouvelles fonctionnalités au compilateur prototype de Vélus et mettre à jour les preuves formelles.

Assignment

Le compilateur Vélus est développé dans l'équipe Inria PARKAS depuis quelques années. Nous avons étendu progressivement le langage source pour enlever la restriction aux programmes normalisés (Bourke et al. 2021) et ajouter les machines à états(Bourke, Pesin, and Pouzet 2023). Dernièrement, dans le travail de thèse de P. Jeanmaire, nous avons développé une sémantique dénotationnelle pour un sous-ensemble du langage (Bourke, Jeanmaire, and Pouzet 2022) dans le but de prendre en compte les erreurs arithmétiques, comme la division par zéro, venant du modèle sous-jacent fourni par CompCert (Leroy 2009), et de faciliter la vérification interactive des programmes.

Il s'agit maintenant d'étendre le travail de thèse de P. Jeanmaire en traitant les structures de contrôle, telles que la réinitialisation et le choix conditionnel sur les blocs d'équations et les machines à états, tout en développant le lien avec une sémantique à la Kahn dans l'assistant de preuve Coq.

Notre sémantique dénotationnelle est basée sur la bibliothèque de C. Paulin-Mohring(Paulin-Mohring 2009). Cette bibliothèque permet de formaliser les fonctions, les flots et les environnements comme de plus petits points fixes dans un ordre partiel complet. Nous nous en sommes servi pour définir la sémantique synchrone, c'est-à-dire, avec d'absences explicites, d'un sous-ensemble du langage d'entrée du compilateur Vélus. Nous avons démontré que, s'il y n'a pas d'erreur arithmétique, cette sémantique satisfait les prédicats de la sémantique relationnelle qui sert de spécification dans la preuve de correction du compilateur. **Le premier but** est d'étendre ce travail aux constructions syntaxiques définies sur les blocs d'équations. On commencera en traitant les variables locales en utilisant un point fixe sur les environnements qui associent des variables aux flots. Ensuite, on adaptera cette idée aux constructions de merge, case, reset, et enfin les machines à états. Il faudra rétablir les propriétés clés sur le typage, le typage d'horloge, la causalité et le traitement d'erreurs arithmétiques.

Avec ses absences explicites, la sémantique synchrone donne une spécification du schéma de compilation (Biernacki et al. 2008). Une valeur n'est calculée et ne peut être utilisée que quand elle est présente. Par contre, pour la vérification interactive des programmes, les absences ne sont pas utiles (Canovas-Dumas and Caspi 2000), en plus, les définitions des opérateurs synchrones, ayant plus de cas, sont plus pénibles à manipuler dans un assistant de preuve. La solution est donc de définir une sémantique à la Kahn et de raisonner là-dedans. Les opérateurs de base sont déjà formalisés (Paulin-Mohring 2009), mais le lien avec la sémantique synchrone n'a pas été démontré dans Coq.**Le second but**

est d'obtenir ce résultat. Il faudrait prendre en compte la possibilité d'erreurs arithmétiques et assurer le transfert des propriétés du modèle à la chaîne de compilation vérifiée existante. Les constructions sur les blocs d'équations seront traitées ou non selon la progression du travail.

Main activities

Travailler en équipe pour bien comprendre la problématique et se mettre d'accord sur les modèles sémantiques et les modifications requises au compilateur. En utilisant l'assistant de preuve Coq, ajouter de nouvelles fonctionnalités au compilateur prototype de Vélus et mettre à jour les preuves formelles.

Skills

- Une formation solide dans la conception de langages de programmation. (*requis*)
- Expérience de la programmation fonctionnelle. (*souhaité*)
- Expérience de la modélisation et vérification formelle avec un assistant de preuve (Coq, HOL, PVS, Isabelle, Lean, etc.). (*souhaité*)
- Expérience préalable de la programmation synchrone. (*optional*)

Benefits package

- Restauration subventionnée
- Transports publics remboursés partiellement
- Congés: 7 semaines de congés annuels + 10 jours de RTT (base temps plein) + possibilité d'autorisations d'absence exceptionnelle (ex : enfants malades, déménagement)
- Possibilité de télétravail (après 6 mois d'ancienneté) et aménagement du temps de travail
- Équipements professionnels à disposition (visioconférence, prêts de matériels informatiques, etc.)
- Prestations sociales, culturelles et sportives (Association de gestion des œuvres sociales d'Inria)
- Accès à la formation professionnelle
- Sécurité sociale

General Information

- **Theme/Domain :** Proofs and Verification
  Software engineering (BAP E)
- **Town/city :** Paris
- **Inria Center :** Centre Inria de Paris
- **Starting date :** 2024-08-01
- **Duration of contract :** 3 months
- **Deadline to apply :** 2024-07-31

Contacts

- **Inria Team :** PARKAS
- **Recruiter :**
  Bourke Timothy / Timothy.Bourke@inria.fr

About Inria

Inria is the French national research institute dedicated to digital science and technology. It employs 2,600 people. Its 200 agile project teams, generally run jointly with academic partners, include more than 3,500 scientists and engineers working to meet the challenges of digital technology, often at the interface with other disciplines. The Institute also employs numerous talents in over forty different professions. 900 research support staff contribute to the preparation and development of scientific and entrepreneurial projects that have a worldwide impact.

The keys to success

Une forte motivation pour appliquer la théorie des langages de programmation et de la logique formelle à la conception et l'amélioration de systèmes pratiques. La volonté de discuter et de collaborer de manière constructive avec d'autres chercheurs.

Warning : you must enter your e-mail address in order to save your application to Inria. Applications must be submitted online on the Inria website. Processing of applications sent from other channels is not guaranteed.

Instruction to apply

Defence Security :
This position is likely to be situated in a restricted area (ZRR), as defined in Decree No. 2011-1425 relating

to the protection of national scientific and technical potential (PPST).Authorisation to enter an area is granted by the director of the unit, following a favourable Ministerial decision, as defined in the decree of 3 July 2012 relating to the PPST. An unfavourable Ministerial decision in respect of a position situated in a ZRR would result in the cancellation of the appointment.

Recruitment Policy :
As part of its diversity policy, all Inria positions are accessible to people with disabilities.