UNIVERSITÉ DE RENNES 1

Direction de la Recherche et de l'Innovation

Appel à projets - Année 2021

Contrats doctoraux «Ordinaires»

ECOLE DOCTORALE : MATHSTIC **POLE DE RECHERCHE : MATHSTIC**

Unité de recherche d'accueil

Unité de Recherche : IRISA – UMR 6074

Équipe : CELTIQUE

Encadrement

Directeur de thèse : Sandrine BLAZY **Unité de recherche** : IRISA – UMR 6074

Employeur/Établissement d'origine : Université de Rennes 1

HDR : ☒ oui ☐ non **Nb de thèses en cours** : 2

Mail : sandrine.blazy@irisa.fr **Tél.** : 02 99 84 22 87

Co-directeur de thèse : Delphine DEMANGE **Unité de recherche** : IRISA – UMR 6074

Employeur/Établissement d'origine : Université de Rennes 1

HDR : ☐ oui ☒ non **Nb de thèses en cours** : 0

Mail : delphine.demange@irisa.fr **Tél.** : 02 99 84 22 26

Sujet

Titre : Techniques de Compilation Optimisante basées sur les Dépendances - Formalisation et Vérification

Mots clés : Sémantique ; Compilation Vérifiée ; Assistants de preuve

Cotutelle internationale (oui/non) : NON **Si oui, préciser l'établissement et le pays de rattachement** :

Sollicitation d'un cofinancement EDUC (oui/non) : NON

Descriptif (15-30 lignes) :
Contexte socioéconomique et scientifique ; Les hypothèses et questions posées ; Les grandes étapes de la thèse et démarche ; Approches méthodologiques et techniques envisagées

La dernière décennie a connu des progrès considérables dans le domaine de la compilation vérifiée, à commencer par les travaux de Leroy et al. relatifs au compilateur C vérifié en Coq : CompCert, un véritable tour de force de vérification mécanisée dans un assistant de preuve. CompCert génère du code compact et efficace pour plusieurs architectures cibles e.g. PowerPC, ARM et x86, pour un sous-ensemble très large du langage C. CompCert est programmé et vérifié en Coq, et son théorème de correction assure que le programme assembleur se comporte comme le programme source -- aucun comportement n'a été ajouté au programme durant le processus de compilation.

Les performances du code produit par CompCert rivalisent avec celles des compilateurs de référence, comme GCC. Néanmoins, certaines techniques de compilation optimisantes modernes restent encore mal comprises d'un point de vue sémantique. C'est le cas des techniques basées sur des représentations intermédiaires de programmes basées sur une représentation explicites des dépendances de données et de contrôle. Ces techniques restent donc à ce jour hors d'atteinte des compilateurs vérifiés.

Les travaux de thèse ciblent plus particulièrement une extension de la forme SSA (Static Single Assignment): Monadic Gated SSA, dans laquelle toutes les dépendances de contrôle sont traduites en dépendances de données. Vis-à-vis des représentations classiquement utilisées, la formalisation de Monadic Gated SSA demande de relâcher l'ordre d'exécution intra-bloc de base du graphe de flot de contrôle et l'ordre d'exécution d' instructions appartenant à des blocs de base distincts. Formaliser Monadic Gated SSA permettrait alors de pouvoir intégrer des optimisations proposées dans les compilateurs modernes (LLVM, GCC...), comme Loop Invariant Code Motion, ou Branch Elimination, qui requièrent d'identifier des invariants sémantiques forts de Monadic Gated SSA portant sur la validité d'équations entre variables et valeurs calculées par le programme dans différents de blocs de base. Ces optimisations ne sont actuellement offertes par aucun compilateur vérifié.

A ce jour, aucun modèle sémantique viable n'est disponible pour Monadic Gated SSA dans l'état de l'art. Récemment, des travaux en compilation vérifiée pour les langages synchrones abordent des problématiques similaires, mais sur un paradigme de programmation et une représentation de programmes radicalement différents.

L'objectif de la thèse est de pouvoir fournir un modèle d'exécution simple de cette représentation, qui soit à la fois fidèle au comportement attendu des programmes, et qui se prête bien au raisonnement formel. Idéalement, le travail de développement sera conduit dans l'assistant de preuve Coq, avec potentiellement des implémentations en OCaml validées a posteriori par un validateur prouvé en Coq. En revanche, des preuves rigoureuses sur papier seront également particulièrement appréciées. Intégrer cette technique de compilation dans un compilateur vérifié tel que CompCertSSA est un des objectifs de la thèse, visant la formalisation de toute la chaîne correspondante de construction de la forme Monadic Gated SSA, optimisation, et génération de code.