

Inférence et visualisation de boucles structurées et déstructurées sur le langage C

Contexte

L'analyse statique du pire temps d'exécution d'un programme requiert la connaissance de limites sur le nombre d'itérations de chacune des **boucles** du programme. Les codes visés par ce type d'analyse sont généralement écrits en langage C. Ce langage offre à la fois des boucles **structurées** (for, while) et des boucles **déstructurées** (goto). Les premières se retrouvent naturellement dans l'**arbre de syntaxe abstrait** alors que les deuxièmes doivent être identifiées à l'aide d'un algorithme sur les graphes [1] appliqué à un **graphe de transition**.

Objectifs du stage

Les objectifs du stages sont les suivants :

- (a) Définir formellement la notion de boucle structurée et déstructurée en terme d'arbre de syntaxe abstrait et de graphe de transition. L'arbre de syntaxe utilisé sera celui fourni par le module OCaml FontC, développé dans l'équipe. Le système de transition s'appuiera sur la sémantique du langage C et peut être fourni par l'encadrant au besoin.
- (b) Développer une application Ocaml qui transforme un code source C en visuel contenant le code C original, une imbrication reflétant l'arbre de syntaxe abstrait et le graphe de transition avec une identification des boucles et de leur nature.
- (c) Adapter l'algorithme présenté en dans la section 4.3 de [1] de sorte qu'il favorise l'identification de boucles structurées. Déterminer ce qui pourrait être prouvable ou non à ce sujet.

Références

[1] François Bourdoncle, Efficient Chaotic Iteration Strategies With Widenings

Qualités et compétences attendues

Goût pour la programmation, pour la compilation, pour la sémantique des langages de programmation, rigueur, maîtrise des mathématiques discrètes (ensembles, graphes, relations), maîtrise du langage C, connaissance du langage OCaml.

Contact et localisation

Pascal Sotin – pascal.sotin@irit.fr

Le stage s'effectuera dans les locaux de l'équipe TRACES à l'IRIT (Toulouse).