

Type System for complexity analysis of Java programs.

Team : INRIA Project Carte

Laboratory : LORIA, Nancy

Advisors :

- Romain Péchoux (Romain.Pechoux@loria.fr), LORIA, Université de Lorraine, Nancy, +33 3 83 59 20 41
- Emmanuel Hainry (Emmanuel.Hainry@loria.fr), LORIA, Université de Lorraine, Nancy, +33 3 54 95 84 18

Skills :

We are looking for master 2 students with good knowledge of **programming languages (type systems)** and skills in **Object Oriented programming** including Java. We also expect the student to have a good skills (and interest) in **complexity theory** and type systems.

Background :

The aim of Implicit Complexity is to design criteria (type systems, semantic interpretations) to prove that programs belong to a given complexity class. The goal is to obtain certificates providing upper bounds on the memory and time needed by a program for a correct execution. A new implicit complexity analysis based on a type system for imperative and object oriented languages was proposed in articles [1], [2] and [7]. This analysis is inspired by Data Ramification techniques [3, 4] and by non-interference (control flow analysis) [5]. It ensures that if a program can be typed and terminates, it will run in polynomial time (or in a different context, polynomial space).

Objectives :

The main objectives of the project are the following :

- Increase the number of programs that can be analyzed using program transformation techniques.
- Combine the complexity analysis with tools for showing the termination of imperative and OO programs (for example, [6]).
- Increase the expressivity of the analyzed language (forks, threads, ...).
- Explore the common cases in real world programs for which the analysis fails and correct (or extend) the type system to capture them.

References :

1. J.-Y. Marion: A Type System for Complexity Flow Analysis. LICS 2011: 123-132
2. J.-Y. Marion, R. Péchoux: Complexity Information Flow in a Multi-threaded Imperative Language, TAMC 2014: 280-299
3. S. Bellantoni et S. Cook - « A new recursion-theoretic characterization of the poly-time functions », Computational Complexity 2 (1992), p. 97-110.
4. D. Leivant et J.-Y. Marion - « Lambda calculus characterizations of poly-time », Fundamenta Informaticae 19 (1993), no. 1,2, p. 167,184.
5. D. M. Volpano, C. E. Irvine, Geoffrey Smith: A Sound Type System for Secure Flow Analysis. Journal of Computer Security 4(2/3): 167-188 (1996)
6. B. Cook, A. Podelski, A. Rybalchenko: Terminator: Beyond Safety. CAV 2006: 415-418
7. E. Hainry, R. Péchoux: Objects In Polynomial Time, APLAS 2015: 387-404