2- Types, modularity, and program transformations

Our programming languages should encourage modularity, or “programming in the large”. That is, they should make it easy to write small re-usable components and to combine these components while ensuring static type safety and keeping verbosity under control. Unfortunately, many real-world programming languages fail to achieve these properties. Java is very weak. The module systems of Standard ML and Ocaml are quite heavyweight, remain surprisingly difficult to formalize, and do not offer an economy of concepts: in Ocaml, records, objects and modules are three distinct concepts. We would like to investigate calculi and programming languages that are theoretically more economical, practically easier to use, and more scalable. Rossberg’s 1ML and Amin and Rompf’s DOT could serve as sources of inspiration. We are interested not only in the theoretical foundations of these calculi, but also in language design and usability issues, including type inference. In the presence of universal / existential / dependent types (and other advanced type-theoretic features), it is a challenge to develop (partial) type inference algorithms whose behavior seems predictable to a programmer and is robust in the face of local program transformations. Finally, we are interested in type-directed program synthesis and program transformation, including program evolution and refactoring.

3- Formalization of weak memory models, mixed size extension

We are interested in type-directed development (partial) type inference algorithms whose behavior seems predictable to a programmer and existential / dependent types (and other advanced type-theoretic features), it is a challenge to also in language design and usability issues, including type inference. In the presence of universal / existential / dependent types (and other advanced type-theoretic features), it is a challenge to develop (partial) type inference algorithms whose behavior seems predictable to a programmer and is robust in the face of local program transformations. Finally, we are interested in type-directed program synthesis and program transformation, including program evolution and refactoring.
Recent work on weak (relaxed) memory models as implemented by contemporary multicore processors addresses “mixed size accesses”. “Mixed size” improves over the simple view of processors atomically accessing independent memory locations: memory accesses are decomposed into several sub-accesses (typically byte accesses) and accesses may be unaligned or overlap. Recent work [1] addresses the issue in the context of operational memory model, where a transition system acts as a model of actual hardware or execution engine.

The DIY suite (http://diy.inria.fr/) provides a formal language (CAT) to describe and implement weak memory models, and tools to verify conformance between these models and actual hardware. By contrast with [1], the models are given in axiomatic style, that is, as conditions that executions has to fulfill to be valid. We are looking forward to extending the modeling machinery and tools so as to handle mixed size accesses. A first objective is to design (and test) a model of “mixed size” x86 machine, with possible extensions to more complex machines and languages (such as C11).


Main activities

Conduct research on one of the topics of interest above.

Skills

Prior research experience is required on:

- For topic 1: mechanized proof, program logics, formal semantics.
- For topic 2: type systems, language design, formal semantics.
- For topic 3: shared-memory concurrency, processor architectures.

Benefits package

- Subsidised catering service
- Partially-reimbursed public transport

Remuneration

- Location: Paris 12ème
- Gross Salary per month: 2 653€ brut/mensuel

Security and defense procedure:

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.