Deep specification and verification of a SQL query planner

Véronique Benzaken Évelyne Contejean Chantal Keller

Envisioned Research

Query languages like SQL are declarative: they specify what the user wants to retrieve and not how to retrieve it. The compilation process consists in two phases: the parsing and planning phases. The first phase translates, as much as possible, the query in a relational algebra expression (algebraic AST). The second phase consists in logical and physical optimisation. The logical optimisation step exploits algebraic equivalences to perform sound query rewritings. The physical optimisation is in charge of producing query evaluation plans which are trees whose nodes are concrete, system-provided, implementations of algebraic operators. This last step is *data dependent* and is achieved based on auxiliary data structures, system maintained statistics and cost functions. Query evaluation plans are then evaluated by the runtime system.

Given an algebraic operator, the underlying system provides several different algorithm implementations for it. For instance to the relational join correspond at least four such different algorithms: nested loop join, index nested loop join, sort-merge join and hash join. The goal of this intern is to formally specify, using Coq, those algorithms, and to verify, using the Why(3) verification tool chain, that they conform to their specification.

Prerequisite

The candidate should have a strong background in theoretical computer science with emphasis in logic as well as a concrete practical experience of functional programming in OCaml-like programming languages. A **good** knowledge of Coq is a plus.

Funding

This intern will be funded in the context of the ANR grant DATACERT.

Location and duration

This intern will take place within the VALS research group of LRI lab in Orsay (bâtiment 650). The VALS group is a world-wide recognised research team in the area of Verification and Validation of Algorithms, Languages and Systems, right in the heart of the scientific field called "Formal Methods" (`https://vals.lri.fr/`). The applicant will be co-advised by Prof., V. Benzaken, Dr., É. Contejean and Dr., Ch. Keller.

Opportunities

This intern could lead to a three years PhD if the candidate demonstrates the expected skills to enroll in a PhD programme. In terms of academic skills, this internship will immerge the candidate in the realm of interactive theorem proving and offer her/him the opportunity to acquire or improve a first experience with Coq. In terms of professional skills, this internship will allow her/him to acquire a first experience with the emerging profession of “proof engineer” as witnessed by many job offers such as the one found at:
`http://ssrg.nicta.com.au/jobs/proof-engineers2015`.

References

[1] V. Benzaken and É. Contejean. SQLCert: Coq mechanisation of SQL’s compilation (formally reconciling SQL and (relational) algebra). Submitted for publication, 2016.

[2] V. Benzaken, É. Contejean, and S. Dumbrava. A Coq Formalization of the Relational Data Model. In *23rd European Symposium on Programming (ESOP)*, 2014.

[3] Véronique Benzaken and Évelyne Contejean. The datacert library (`http://datacert.lri.fr/`), 2012.

[4] Jean-Christophe Filliâtre and Andrei Paskevich. Why3 - where programs meet provers. In *ESOP*, pages 125–128, 2013.

[5] Xavier Leroy. A formally verified compiler back-end. *J. Autom. Reasoning*, 43(4):363–446, 2009.

[6] Gregory Malecha, Greg Morrisett, Avraham Shinnar, and Ryan Wisnesky. Toward a verified relational database management system. In *ACM Int. Conf. POPL*, 2010.

[7] The Coq Development Team. *The Coq Proof Assistant Reference Manual*, 2010. `http://coq.inria.fr`.