

Vérification de compilation de requêtes SQL à base de traces

Véronique Benzaken

Évelyne Contejean

Chantal Keller

Description

Les langages de requêtes tels que SQL sont déclaratifs : ils décrivent ce que l'on souhaite obtenir et non comment l'obtenir. Leur compilation s'effectue en deux phases : le "parsing" et l'optimisation. La première phase consiste à traduire la requête en une expression d'une algèbre permettant la manipulation symbolique des requêtes. Dû au très haut degré de déclarativité, la phase d'optimisation est particulièrement complexe car exploitant de nombreuses réécritures logiques mais également des informations système relatives à la manière dont sont implantés les opérateurs algébriques, dont sont stockées les données, au maintien de statistiques d'utilisation de ces dernières et de fonctions de coût sophistiquées. Cette phase conduit à l'obtention d'un plan d'exécution de la requête qui est un arbre syntaxique dont les feuilles sont des méthodes d'accès aux données (parcours séquentiel de fichiers, utilisation d'index) et les noeuds des implantations de l'opérateur de jointure de l'algèbre (boucles imbriquées, tri-fusion ... etc).

Tout SGBD offre la possibilité de consulter le plan qui sera exécuté au moyen de la commande SQL: EXPLAIN. Ce plan peut être vu comme une trace d'exécution qui permet de certifier a posteriori la compilation de la requête.

Le but du stage est de réaliser, en OCaml, un outil permettant de relier un plan donné à la sémantique algébrique de la requête. Il s'inscrit dans le cadre du projet DataCert visant à l'obtention d'un compilateur certifié de requêtes SQL en Coq.

Prérequis

Coq n'est pas un prérequis. En revanche, une connaissance de la programmation fonctionnelle est souhaitée.

Financement

Le stage peut être financé dans le cadre du projet ANR DataCert.

Lieu du stage

Le stage déroulera au sein de l'équipe VALS (Verification and validation of Algorithms, Languages and Systems) du LRI à Paris-Saclay (<https://vals.lri.fr/>).

Opportunités

Ce stage permet de se plonger au coeur du domaine des méthodes formelles ainsi que de découvrir les aspects systèmes de la compilation de SQL.

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>.