Université de Lorraine — IDMC
Masters SC & TAL — M1
2018–2019

Fiche de projet tutoré

Optimisation d'analyse syntaxique par réécriture de programme

Encadrement :
— Équipe Sémagramme du LORIA
— Sylvain Pogodalla, sylvain.pogodalla@inria.fr

Description

1. **Projet global** Le formalisme des grammaires catégorielles abstraites (ACG, de Groote (2001)) est un cadre grammatical qui permet l'encodage de divers formalismes grammaticaux, notamment les grammaires d'arbres adjoints (TAG, Joshi et Schabes (1997)). Il se caractérise par la définition de langages de λ-termes, qui généralisent les langages de chaînes de caractères ou d'arbres.

   L'analyse avec les ACG peut se ramener à de la recherche de démonstrations pour une requête en Datalog, un langage de programmation logique pour la gestion de bases de données (Abiteboul, Hull et Vianu 1995). L'optimisation de l'exécution de requêtes pour les bases de données a conduit à proposer des réécritures de programmes Datalog, par exemple la réécriture *Magic set*. L'objet du stage est d'étudier et d'implanter la réécriture de tels programmes dans le cadre de l'analyse syntaxique avec les ACG.
2. **Bibliographie** Cette partie du projet consistera en l'étude des différents formalismes utilisés et des algorithmes mis en œuvre dans ce cadre : analyse, recherche de démonstrations Datalog, réécriture de programme. La réécriture de programme modifiant les arbres de démonstrations obtenus, une première réflexion sur la manière de retrouver les arbres originaux sera conduite avec une étude bibliographique étendue aux grammaires et automates d'arbres (Comon et al. 2007). De premiers tests d'efficacité, sur des grammaires très simples permettant de s'approprier les algorithmes de réécriture, pourront être menés à l'aide du logiciel ACGtk, qui comprend notamment un prouveur Datalog.
3. **Réalisation** Cette deuxième partie consistera à implanter les algorithmes de réécriture de programme. L'étude sur la reconstruction des arbres de démonstration originaux sera complétée et, si possible, implantée comme extension dans ACGtk. Des grammaires de différentes tailles permettront de tester les réécritures. Enfin, un travail sur la possibilité de généraliser la description des réécritures pourra être conduit afin de tester différents types de réécritures.

Université de Lorraine — IDMC
Masters SC & TAL — M1
2018–2019

Informations diverses : De nombreuses notions seront abordées, ce qui nécessitera une forte interaction. Pour la partie implantation, celle-ci se fera en OCaml, langage dans lequel est développé et maintenu ACGtk.

Livrables et échéancier : La première partie du travail sera jalonnée de diverses présentations.

- t0+1mois : présentation de Datalog et de l'algorithme de recherche de preuve
- t0+2mois : présentation de la réécriture *Magic set*
- t0+3mois : présentation sur les automates d'arbres et rapport sur la partie bibliographique

Pour la deuxième partie :

- t0+4mois : Tests de réécriture sur des toutes petites grammaires
- t0+5mois : Prototypage de la réécriture *Magic set*
- t0+6mois : Intégration de la réécriture dans ACGtk, analyse de la transformation des arbres de démonstration
- t0+7mois : Proposition pour la transformation des arbres de démonstration, rapport, préparation de la soutenance.

Abiteboul, Serge, Richard Hull et Victor Vianu (1995). *Foundations of Databases*. Assison-Wesley. URL : http://webdam.inria.fr/Alice/pdfs/all.pdf.

Comon, Hubert et al. (2007). *Tree Automata Techniques and Applications*. release October 12th, 2007. URL : http://www.grappa.univ-lille3.fr/tata.

de Groote, Philippe (2001). "Towards Abstract Categorial Grammars". In : *Proceedings of 39th Annual Meeting of the Association for Computational Linguistics*, p. 148–155. Anthologie ACL : P01-1033.

Joshi, Aravind K. et Yves Schabes (1997). "Tree-adjoining grammars". In : *Handbook of formal languages*. Sous la dir. de Grzegorz Rozenberg et Arto K. Salomaa. T. 3. Springer. Chap. 2.