

Explicit Substitutions in the Reduction of Lambda Terms

Référence

- Titre de l'article : "Explicit Substitutions in the Reduction of Lambda Terms"
- Auteurs : Gopalan Nadathur, Xiaochu Qi
- Date de publication : 2003
- Revue : Proceedings of the ACM SIGPLAN Conference on Principles and Practice of Declarative Programming
- Pages : 195–206

Résumé

Cet article propose des méthodes de substitution dans l'implémentation des beta reductions de lambda termes, qui utilisent la récursivité dans l'application des règles de réécriture, et qui suspendent les opérations de substitution lorsque c'est profitable. Il présente d'abord un calcul explicite de substitution déjà existant, puis il présente trois différentes nouvelles approches de réduction, avant de les comparer quantitativement. Les méthodes de substitution présentées sont plus efficaces en temps et en espace que ce qui existait.

Contexte scientifique

Les principaux articles cités qui appuient le travail effectués :

- Un article sur la réduction forte, qui est la réduction utilisée utilisée par les auteurs
B. Grégoire and X. Leroy. A compiled implementation of strong reduction. In Proceedings of the Seventh ACM SIGPLAN International Conference on Functional Programming, pages 235–246, Pittsburgh, October 2002.
- Un article sur le schéma de de Bruijn qui permet de comparer des termes modulo alpha réduction ou renommage de variables liées
N. Bruijn. Lambda calculus notation with nameless dummies, a tool for automatic formula manipulation, with application to the Church-Rosser Theorem. Indag. Math., 34(5) :381–392, 1972.
- Un article sur *Teyjus*, une implemantation de λ Prolog qui est utilisée par les auteurs pour faire des évaluations expérimentales
G. Nadathur and D. J. Mitchell. System description : Teyjus—a compiler and abstract machine based implementation of λ Prolog. In H. Ganzinger, editor, Automated Deduction–CADE-16, number 1632 in Lecture Notes in Artificial Intelligence, pages 287–291. Springer-Verlag, July 1999.
- Un article sur la substitution qu'ils décrivent pour comparer avec leurs différentes approche : la "suspension notation"
G. Nadathur and D. S. Wilson. A notation for lambda terms : A generalization of environments. Theoretical Computer Science, 198(1-2) :49–98, 1998.

Cet article a été cité dans cinq autres articles de recherche :

- Nadathur, Gopalan. (2003). A Treatment of Higher-Order Features in Logic Programming. Theory and Practice of Logic Programming. 5. 10.1017/S1471068404002297.
- Pottier, François. (2006). An Overview of Caml. Electr. Notes Theor. Comput. Sci.. 148. 27-52. 10.1016/j.entcs.2005.11.039.
- Qi, Xiaochu. (2004). Reduction Strategies in Lambda Term Normalization and their Effects on Heap Usage.
- Ferreira, Francisco & Monnier, Stefan & Pientka, Brigitte. (2013). Compiling contextual objects : bringing higher-order abstract syntax to programmers. 13-24. 10.1145/2428116.2428121.
- Moura, Flávio & Ayala-Rincon, Mauricio & Kamareddine, Fairouz. (2006). SUBSEXPL :A tool for simulating and comparing explicit substitutions calculi*. Journal of Applied Non-Classical Logics. 16. 119-150. 10.3166/jancl.16.119-150.

Pertinence pour lecture approfondie

Cet article est un article technique d'implémentation de méthode de substitution. Il n'établit pas de résultat théorique important. Il est intéressant de le lire seulement s'il y a besoin d'utiliser ce genre d'implémentation. Le code y est bien détaillé et expliqué.