

# The Metalanguage $\lambda$ Prolog and Its Implementation

## Référence

- Titre de l'article : "The Metalanguage  $\lambda$ Prolog and Its Implementation"
- Auteur : Gopalan Nadathur
- Date de publication : 2001

## Résumé

Cet article est un article de définition et de description du métalangage  $\lambda$ Prolog. L'objectif de l'article est de montrer que les nouvelles fonctionnalités présentes dans le  $\lambda$ Prolog peuvent simplifier la programmation des manipulations syntaxiques, et qu'elles peuvent être mises en oeuvre avec une efficacité suffisante pour être des outils pratiques dans ce domaine. Le  $\lambda$ Prolog est une extension de la programmation logique traditionnelle qui permet de tenir compte des "structures de liaisons" (binding structures) dans le codage d'objets syntaxiques. La syntaxe du  $\lambda$ Prolog et son implémentation sont expliqués en détail dans l'article.

## Contexte scientifique

Trois articles principaux précèdent celui-ci :

- un article qui fait référence à la théorie d'ordre supérieur des formules de Harrop héréditaires. L'auteur fait référence à cet article en disant que le  $\lambda$ Prolog est basé sur cette théorie. C'est une théorie qui incarne une riche interprétation de l'idée abstraite de la programmation logique :

*D. Miller, G. Nadathur, F. Pfenning, and A. Scedrov. Uniform proofs as a foundation for logic programming. Annals of Pure and Applied Logic , 51 :125-157, 1991.*

- un article qui introduit la liaison et son importance. C'est un article sur le développement d'un traitement explicite de la liaison dans la représentation syntaxique, aboutissant à ce que l'on appelle désormais la syntaxe abstraite d'ordre supérieur (higher-order abstract syntax) :

*F. Pfenning and C. Elliott. Higher-order abstract syntax. In Proceedings of the ACM-SIGPLAN Conference on Programming Language Design and Implementation , pages 199-208. ACM Press, June 1988.*

- un article sur *Teyjus*, une implémentation récente du  $\lambda$ Prolog au moment de la publication qui utilise certaines idées développée par l'auteur :

*G. Nadathur and D.J. Mitchell. System description : Teyjus/a compiler and abstract machine based implementation of  $\lambda$ Prolog. In H. Ganzinger, editor, Automated Deduction-CADE-16 , number 1632 in Lecture Notes in Artificial Intelligence, pages 287-291. Springer-Verlag, July 1999.*

Cet article est cité dans quatre publications :

- Byrd, William. (2009). Relational Programming in miniKanren : Techniques, Applications, and Implementations.

- Momigliano, Alberto & Ornaghi, Mario. (2010). Proof-Theoretic and Higher-Order Extensions of Logic Programming. 254-270. 10.1007/978-3-642-14309-0\_12.
- Abel, Andreas & Kraus, Nicolai. (2011). A Lambda Term Representation Inspired by Linear Ordered Logic. Electronic Proceedings in Theoretical Computer Science. 71. 10.4204/EPTCS.71.1.
- Cervesato, Iliano & Pfenning, Frank. (2004). A Linear Spine Calculus. Journal of Logic and Computation. 13. 10.1093/logcom/13.5.639.

## **Pertinence pour lecture approfondie**

Cet article est intéressant si on a envie de comprendre le  $\lambda$ Prolog et son usage en tant que méta-langage. Les règles y sont expliquées en détail, donc l'article peut être une référence pour comprendre d'autres documents. La partie sur l'implémentation est plus technique, mais elle est bien détaillée donc utile pour le travail d'implémentation.