



A First Order Forward Chaining Approach for Answer Set Computing

Submitted by Emmanuel Lemoine on Mon, 10/06/2014 - 18:25

Titre	A First Order Forward Chaining Approach for Answer Set Computing
Type de publication	Communication
Type	Communication avec actes dans un congrès
Année	2009
Langue	Anglais
Date du colloque	2009/01/01
Titre du colloque	10th International Conference, LPNMR 2009
Titre des actes ou de la revue	Logic Programming and Nonmonotonic Reasoning
Volume	5753
Pagination	196 - 208
Auteur	Lefevre, Claire [1], Nicolas, Pascal [2]
Pays	Allemagne
Editeur	Springer
Ville	Potsdam
ISBN	978-3-642-04237-9 / 978-3-642-04238-6
Mots-clés	Artificial Intelligence (incl. Robotics) [3], Discrete Mathematics in Computer Science [4], Logics and Meanings of Programs [5], Mathematical Logic and Formal Languages [6], Programming Techniques [7], Software Engineering [8]
Résumé en anglais	<p>The natural way to use Answer Set Programming (ASP) to represent knowledge in Artificial Intelligence or to solve a Constraint Satisfaction Problem is to elaborate a first order logic program with default negation. In a preliminary step this program, with variables, is translated in an equivalent propositional one by a first tool: the grounder. Then, the propositional program is given to a second tool: the solver. This last one computes (if they exist) one or many answer sets (models) of the program, each answer set encoding one solution of the initial problem. Until today, almost all ASP systems apply this two steps computation. In this work, our major contribution is to introduce a new approach of answer set computing that escapes the preliminary phase of rule instantiation by integrating it in the search process. Our methodology applies a forward chaining of first order rules that are grounded on the fly by means of previously produced constants. We have implemented this strategy in our new ASP solver ASPeRiX. The first benefit of our work is to avoid the bottleneck of instantiation phase arising for some problems because of the huge amount of memory needed to ground all rules of a program, even if these rules are not really useful in certain cases. The second benefit is to make the treatment of function symbols easier and without syntactic restriction provided that rules are safe.</p>
Notes	Date du colloque : 09/2009

URL de la notice <http://okina.univ-angers.fr/publications/ua4487> [9]
DOI [10.1007/978-3-642-04238-6_18](https://doi.org/10.1007/978-3-642-04238-6_18) [10]
Lien vers le document en ligne http://dx.doi.org/10.1007/978-3-642-04238-6_18 [10]

Liens

- [1] <http://okina.univ-angers.fr/cl.le/publications>
- [2] [http://okina.univ-angers.fr/publications?f\[author\]=7495](http://okina.univ-angers.fr/publications?f[author]=7495)
- [3] [http://okina.univ-angers.fr/publications?f\[keyword\]=6729](http://okina.univ-angers.fr/publications?f[keyword]=6729)
- [4] [http://okina.univ-angers.fr/publications?f\[keyword\]=8862](http://okina.univ-angers.fr/publications?f[keyword]=8862)
- [5] [http://okina.univ-angers.fr/publications?f\[keyword\]=8934](http://okina.univ-angers.fr/publications?f[keyword]=8934)
- [6] [http://okina.univ-angers.fr/publications?f\[keyword\]=8919](http://okina.univ-angers.fr/publications?f[keyword]=8919)
- [7] [http://okina.univ-angers.fr/publications?f\[keyword\]=8990](http://okina.univ-angers.fr/publications?f[keyword]=8990)
- [8] [http://okina.univ-angers.fr/publications?f\[keyword\]=8901](http://okina.univ-angers.fr/publications?f[keyword]=8901)
- [9] <http://okina.univ-angers.fr/publications/ua4487>
- [10] http://dx.doi.org/10.1007/978-3-642-04238-6_18

Publié sur *Okina* (<http://okina.univ-angers.fr>)