

Negation as Failure with Set Functions

Michael Hanus Björn Peemöller

Fabian Reck

Institut für Informatik, CAU Kiel, D-24098 Kiel, Germany

`{mh|bjp|fre}@informatik.uni-kiel.de`

Set functions [1] were proposed as a novel approach to encapsulate non-deterministic computations in functional logic programs. As the name suggests, a set function returns the set of all the non-deterministic results of a function call. In contrast to other approaches to encapsulated search, the result of a set function does not depend on the evaluation order. This is achieved by separating the non-determinism that is introduced by actual arguments from the non-determinism introduced by the body of the function.

Obviously, it is possible to check if the result of a set function is the empty set. Set functions, therefore, are a means to adapt negation as failure from logic programming.

We will present our approach to integrate set functions into our Curry system *KiCS2* [2]. Then we discuss the properties of failing computations and nesting w.r.t. set functions that were not considered in the original proposal of set functions, but become essential when set functions are used to program with negation as failure.

References

- [1] S. Antoy and M. Hanus, *Set Functions for Functional Logic Programming*, Proc. of the 11th International ACM SIGPLAN Conference on Principle and Practice of Declarative Programming (PPDP'09), ACM Press, 2009, pp. 73–82.
- [2] B. Braßel, M. Hanus, B. Peemöller, and F. Reck, *KiCS2: A New Compiler from Curry to Haskell*, Proc. of the 20th International Workshop on Functional and (Constraint) Logic Programming (WFLP 2011), Springer LNCS 6816, 2011, pp. 1–18.