

Transforming Functional Logic Programs into Monadic Functional Programs

Bernd Braßel Sebastian Fischer Michael Hanus Fabian Reck
Institut für Informatik, CAU Kiel, D-24098 Kiel, Germany
`{bbr|sebf|mh|fre}@informatik.uni-kiel.de`

We present a high-level transformation scheme to translate lazy functional logic programs into pure Haskell programs. This transformation is based on a recent proposal to efficiently implement lazy non-deterministic computations in Haskell into monadic style. We build on this work and define a systematic method to transform lazy functional logic programs into monadic programs with explicit sharing. This results in a transformation scheme which produces high-level and flexible target code. For instance, the target code is parametric w.r.t. the concrete evaluation monad. Thus, different monad instances could, for example, define different search strategies (e.g., depth-first, breadth-first, parallel).