

Adding Plural Arguments to Curry Programs

Michael Hanus

Institut für Informatik, CAU Kiel, D-24098 Kiel, Germany.
`mh@informatik.uni-kiel.de`

Abstract. Functional logic languages combine lazy (demand-driven) evaluation strategies from functional programming with non-deterministic computations from logic programming. To provide a strategy-independent semantics, most languages are based on the call-time choice semantics where parameters are passed as values. From an implementation point of view, the call-time choice semantics fits well with sharing performed by lazy languages. On the other hand, there are also situations where it is intended to pass non-deterministic arguments as sets of values in order to exploit the power of non-deterministic programming. This alternative parameter passing model is known under the name “plural” arguments. In this talk, we show how both mechanisms can be integrated in a single language. In particular, we present a novel technique to implement plural arguments in a call-time choice language so that existing implementations of contemporary functional logic languages can be easily re-used to implement plural parameter passing.