Problem #50

Originator: Jean-Pierre Jouannaud
Date: December 1991

Summary: Investigate confluence and termination of combinations of typed lambda-calculi with term rewriting systems.

Combinations of typed \( \lambda \)-calculi with term-rewriting systems have been studied extensively in the past few years [Bar90][BTG89][DO90][Dou91]. The strongest termination result allows first-order rules as well as higher-order rules defined by a generalization of primitive recursion. Suppose all rules for functional constant \( F \) follow the schema:

\[
F(\bar{l}[\bar{X}], \bar{Y}) \rightarrow v[F(\bar{r}_1[\bar{X}], \bar{Y}), ..., F(\bar{r}_m[\bar{X}], \bar{Y}), \bar{Y}]]
\]

where the (not necessarily disjoint) variables in \( \bar{X} \) and \( \bar{Y} \) are of arbitrary order, each of \( \bar{l}, \bar{r}_1, ..., \bar{r}_m \) is in \( T(\mathcal{F}, \{\bar{X}\}) \), \( v[\bar{z}, \bar{Y}] \) is in \( T(\mathcal{F}, \{\bar{Y}, \bar{z}\}) \), for new variables \( \bar{z} \) of appropriate types, and \( \bar{r}_1, ..., \bar{r}_m \) are each less than \( \bar{l} \) in the multiset extension of the strict subterm ordering. If \( T(\mathcal{F}, \mathcal{X}') \) is the term-algebra which includes only previously defined functional constants— forbidding the use of mutually recursive functional constants—termination is ensured [JO91]. Does termination also hold when there are mutually recursive definitions? Does this also hold when the subterm assumption is unfulfilled? (In [JO91] an alternative schema is proposed, with the subterm assumption weakened at the price of having only first-order variables in \( \bar{X} \).) Questions of confluence of combinations of typed \( \lambda \)-calculi and higher-order systems also merit investigation. These results have been extended to combinations with more expressive type systems [BF93b][BF93a].

Remark

An extension to the Calculus of Constructions has been reported in [BFG94]. One can also allow the use of lexicographic and other “statutes” for the higher-order constants when comparing the subterms of \( F \) in left and right hand sides [Jouannaud and Okada, unpublished]. Finally, this can also be done when the rewrite rules follow from the induction schema in the initial algebra of the constructors [Wer94].

Important improvements of the previous works have been achieved in [Bla03] and [WC03].

http://www.win.tue.nl/rtaloop/
Bibliography


[Dou91] Daniel Dougherty. Adding algebraic rewriting to the untyped lambda calculus (extended abstract). In Ronald. V. Book, ed-

