

McCracken, Nancy

The typechecking of programs with implicit type structure. (English) Zbl 0579.68009
Semantics of data types, Proc. int. Symp., Sophia-Antipolis/France 1984, Lect. Notes Comput. Sci. 173, 301-315 (1984).

[For the entire collection see [Zbl 0534.00019](#).]

Since the specification of complete type information would make a syntactically cumbersome language, a solution, discussed in the present paper, is to regard the typechecking part of a compiler as a translator from a program with implicit type information to a program with explicit type information.

After a short review on polymorphic λ -calculus, section II presents a typechecker that takes an implicitly typed program and generates the computationally equivalent explicitly typed program, approach mainly based on [*R. Milner*, J. Comput. Syst. Sci. 17, 348- 375 (1978; [Zbl 0388.68003](#))]. In section III the author extends the Leivant's typing algorithm [*D. Leivant*, Polymorphic type inference in "Tenth ann. Symp. on principles of programming languages" (Austin/Texas 1983)], for the conjunctive type system, restricted to rank 2, by constructing explicitly typed expressions.

These two typecheckers are "combined" (section IV) for keeping a well- balanced (thus useful) syntax that makes the mixture of implicit and explicit types, deducing types for programs from the combined set of implicit typing inference rules.

Reviewer: [N. Curteanu](#)

MSC:

[68Q60](#) Specification and verification (program logics, model checking, etc.)

Cited in **5** Documents

Keywords:

[semantics of data types](#); [type inference for polymorphic functions](#); [typechecker construction](#)