

Leivant, Daniel M.

Global semantic typing for inductive and coinductive computing. (English) Zbl 1356.03073
[Log. Methods Comput. Sci. 10, No. 4, Paper No. 18, 23 p. \(2014\).](#)

Summary: Inductive and coinductive types are commonly construed as ontological (Church-style) types, denoting canonical data-sets such as natural numbers, lists, and streams. For various purposes, notably the study of programs in the context of global semantics, it is preferable to think of types as semantical properties (Curry-style). Intrinsic theories were introduced in the late 1990s to provide a purely logical framework for reasoning about programs and their semantic types. We extend them here to data given by any combination of inductive and coinductive definitions. This approach is of interest because it fits tightly with syntactic, semantic, and proof theoretic fundamentals of formal logic, with potential applications in implicit computational complexity as well as extraction of programs from proofs.

We prove a Canonicity Theorem, showing that the global definition of program typing, via the usual (Tarskian) semantics of first-order logic, agrees with their operational semantics in the intended model.

Finally, we show that every intrinsic theory is interpretable in a conservative extension of first-order arithmetic. This means that quantification over infinite data objects does not lead, on its own, to proof-theoretic strength beyond that of Peano Arithmetic. Intrinsic theories are perfectly amenable to formulas-as-types Curry-Howard morphisms, and were used to characterize major computational complexity classes. Their extensions described here have similar potential which has already been applied.

MSC:

[03B70](#) Logic in computer science

[68N30](#) Mathematical aspects of software engineering (specification, verification, metrics, requirements, etc.)

Keywords:

[inductive types](#); [coinductive types](#); [equational programs](#); [intrinsic theories](#); [global model theory](#)

Software:

[CASL](#); [CCSL](#); [CoCasl](#)

Full Text: [DOI](#)