J Autom Reasoning (2013) 50:383–421
DOI 10.1007/s10817-012-9253-6

A Two-Valued Logic for Properties of Strict Functional Programs Allowing Partial Functions

David Sabel · Manfred Schmidt-Schauß

Received: 13 October 2011 / Accepted: 25 May 2012 / Published online: 8 June 2012
© Springer Science+Business Media B.V. 2012

Abstract A typed program logic LMF for recursive specification and verification is presented. It comprises a strict functional programming language with polymorphic and recursively defined partial functions and polymorphic data types. The logic is two-valued with the equality symbol as only predicate. Quantifiers range over the values, which permits inductive proofs of properties. The semantics is based on a contextual (observational) semantics, which gives a consistent presentation of higher-order functions. Our analysis also sheds new light on the the role of partial functions and loose specifications. It is also an analysis of influence of extensions of programs on the tautologies. The main result is that universally quantified equations are conservative, which is also the base for several other conservative classes of formulas.

Keywords Verification · Functional programming · Logics · Semantics · Contextual equivalence

1 Introduction

Clearly, programming and reasoning about the properties of programs is a core task of computer science. A great variety of program logics for this purpose are proposed in the computer science literature.

Electronic supplementary material The online version of this article (doi:10.1007/s10817-012-9253-6) contains supplementary material, which is available
to authorized users.

D. Sabel (✉) · M. Schmidt-Schauß
Institut für Informatik, Goethe-Universität, Postfach 11 19 32, 60054 Frankfurt, Germany
e-mail: sabel@ki.informatik.uni-frankfurt.de

M. Schmidt-Schauß
e-mail: schauss@ki.informatik.uni-frankfurt.de