The Future is Ours: Prophecy Variables in Separation Logic

RALF JUNG, MPI-SWS, Germany
RODOLPHE LEPIGRE, MPI-SWS, Germany
GAURAV PARTHASARATHY, ETH Zurich, Switzerland and MPI-SWS, Germany
MARIANNA RAPOPORT, University of Waterloo, Canada and MPI-SWS, Germany
AMIN TIMANY, imec-DistriNet, KU Leuven, Belgium
DEREK DREYER, MPI-SWS, Germany
BART JACOBS, imec-DistriNet, KU Leuven, Belgium

Early in the development of Hoare logic, Owicki and Gries introduced *auxiliary variables* as a way of encoding information about the *history* of a program's execution that is useful for verifying its correctness. Over a decade later, Abadi and Lamport observed that it is sometimes also necessary to know in advance what a program will do in the *future*. To address this need, they proposed *prophecy variables*, originally as a proof technique for refinement mappings between state machines. However, despite the fact that prophecy variables are a clearly useful reasoning mechanism, there is (surprisingly) almost no work that attempts to integrate them into Hoare logic. In this paper, we present the first account of prophecy variables in a Hoare-style program logic that is flexible enough to verify *logical atomicity* (a relative of linearizability) for classic examples from the concurrency literature like RDCSS and the Herlihy-Wing queue. Our account is formalized in the Iris framework for separation logic in Coq. It makes essential use of *ownership* to encode the exclusive right to resolve a prophecy, which in turn lets us enforce soundness of prophecies with a very simple set of proof rules.

CCS Concepts: • **Theory of computation → Separation logic**; *Programming logic*; Operational semantics.

Additional Key Words and Phrases: Prophecy variables, separation logic, logical atomicity, linearizability, Iris

ACM Reference Format:
Ralf Jung, Rodolphe Lepigre, Gaurav Parthasarathy, Marianna Rapoport, Amin Timany, Derek Dreyer, and Bart Jacobs. 2020. The Future is Ours: Prophecy Variables in Separation Logic. *Proc. ACM Program. Lang.* 4, POPL, Article 45 (January 2020), 32 pages. https://doi.org/10.1145/3371113

1 INTRODUCTION

When proving correctness of a program P, it is often easier and more natural to reason *forward*—that is, to start at the beginning of P's execution and reason about how it behaves as it executes. But sometimes strictly forward reasoning is not good enough: when reasoning about a program step s_0, it may be necessary to "peek into the future" and know ahead of time what will happen at some future program step s_1.

Authors' addresses: Ralf Jung, MPI-SWS, Saarland Informatics Campus, Germany, jung@mpi-sws.org; Rodolphe Lepigre, MPI-SWS, Saarland Informatics Campus, Germany, lepigre@mpi-sws.org; Gaurav Parthasarathy, Department of Computer Science, ETH Zurich, Switzerland and MPI-SWS, Germany, gaurav.parthasarathy@inf.ethz.ch; Marianna Rapoport, University of Waterloo, Canada and MPI-SWS, Germany, mrapoport@uwaterloo.ca; Amin Timany, imec-DistriNet, KU Leuven, Belgium, amin.timany@cs.kuleuven.be; Derek Dreyer, MPI-SWS, Saarland Informatics Campus, Germany, dreyer@mpi-sws.org; Bart Jacobs, imec-DistriNet, KU Leuven, Belgium, bart.jacobs@cs.kuleuven.be.

2475-1421/2020/1-ART45
https://doi.org/10.1145/3371113