

Cambridge University Press

978-1-107-04801-0 - Program Logics for Certified Compilers

Andrew W. Appel, Robert Dockins, Aquinas Hobor, Lennart Beringer, Josiah Dodds,  
Gordon Stewart, Sandrine Blazy and Xavier Leroy

Frontmatter

[More information](#)

---

## PROGRAM LOGICS FOR CERTIFIED COMPILERS

Separation logic is the twenty-first-century variant of Hoare logic that permits verification of pointer-manipulating programs. This book covers practical and theoretical aspects of separation logic at a level accessible to beginning graduate students interested in software verification. On the practical side it offers an introduction to verification in Hoare and separation logics, simple case studies for toy languages, and the Verifiable C program logic for the C programming language. On the theoretical side it presents separation algebras as models of separation logics; step-indexed models of higher-order logical features for higher-order programs; indirection theory for constructing step-indexed separation algebras; tree-shares as models for shared ownership; and the semantic construction (and soundness proof) of Verifiable C. In addition, the book covers several aspects of the CompCert verified C compiler, and its connection to foundationally verified software analysis tools. All constructions and proofs are made rigorous and accessible in the Coq developments of the open-source Verified Software Toolchain.

Andrew W. Appel is the Eugene Higgins Professor and Chairman of the Department of Computer Science at Princeton University, where he has been on the faculty since 1986. His research is in software verification, computer security, programming languages and compilers, automated theorem proving, and technology policy. He is known for his work on Standard ML of New Jersey and on Foundational Proof-Carrying Code. He is a Fellow of the Association for Computing Machinery, recipient of the ACM SIGPLAN Distinguished Service Award, and has served as Editor-in-Chief of *ACM Transactions on Programming Languages and Systems*. His previous books include *Compiling with Continuations* (1992), the *Modern Compiler Implementation* series (1998 and 2002), and *Alan Turing's Systems of Logic* (2012).

Cambridge University Press

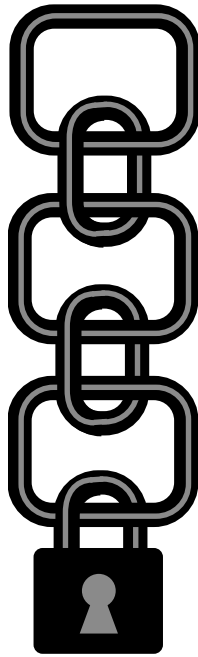
978-1-107-04801-0 - Program Logics for Certified Compilers

Andrew W. Appel, Robert Dockins, Aquinas Hobor, Lennart Beringer, Josiah Dodds,  
Gordon Stewart, Sandrine Blazy and Xavier Leroy

Frontmatter

[More information](#)

---



Cambridge University Press

978-1-107-04801-0 - Program Logics for Certified Compilers

Andrew W. Appel, Robert Dockins, Aquinas Hobor, Lennart Beringer, Josiah Dodds,

Gordon Stewart, Sandrine Blazy and Xavier Leroy

Frontmatter

[More information](#)

# PROGRAM LOGICS FOR CERTIFIED COMPILERS

ANDREW W. APPEL

*Princeton University, Princeton, New Jersey*

ROBERT DOCKINS

*Portland State University, Portland, Oregon*

AQUINAS HOBOR

*National University of Singapore and Yale/NUS College, Singapore*

LENNART BERINGER

*Princeton University, Princeton, New Jersey*

JOSIAH DODDS

*Princeton University, Princeton, New Jersey*

GORDON STEWART

*Princeton University, Princeton, New Jersey*

SANDRINE BLAZY

*Université de Rennes 1*

XAVIER LEROY

*INRIA Paris-Rocquencourt*



CAMBRIDGE  
UNIVERSITY PRESS

Cambridge University Press  
978-1-107-04801-0 - Program Logics for Certified Compilers  
Andrew W. Appel, Robert Dockins, Aquinas Hobor, Lennart Beringer, Josiah Dodds,  
Gordon Stewart, Sandrine Blazy and Xavier Leroy  
Frontmatter  
[More information](#)

---

**CAMBRIDGE**  
UNIVERSITY PRESS

32 Avenue of the Americas, New York, NY 10013-2473, USA

Cambridge University Press is part of the University of Cambridge  
It furthers the University's mission by disseminating knowledge in the pursuit of  
education, learning, and research at the highest international levels of excellence.

[www.cambridge.org](http://www.cambridge.org)  
Information on this title: [www.cambridge.org/9781107048010](http://www.cambridge.org/9781107048010)

© Andrew W. Appel 2014

This publication is in copyright. Subject to statutory exception  
and to the provisions of relevant collective licensing agreements,  
no reproduction of any part may take place without the written  
permission of Cambridge University Press.

First published 2014

Printed in the United States of America

*A catalog record for this publication is available from the British Library.*

ISBN 978-1-107-04801-0 Hardback

Cambridge University Press has no responsibility for the persistence  
or accuracy of URLs for external or third-party Internet Web sites referred  
to in this publication and does not guarantee that any content on such  
Web sites is, or will remain, accurate or appropriate.

This book is typeset in the Bitstream Charter font.  
Font Copyright ©1989–1992, Bitstream Inc., Cambridge, MA.

Cambridge University Press

978-1-107-04801-0 - Program Logics for Certified Compilers

Andrew W. Appel, Robert Dockins, Aquinas Hobor, Lennart Beringer, Josiah Dodds,

Gordon Stewart, Sandrine Blazy and Xavier Leroy

Frontmatter

[More information](#)

---

*in memory of*

*Kenneth I. Appel*

*1932–2013*

*a pioneer in computer proof*

Cambridge University Press

978-1-107-04801-0 - Program Logics for Certified Compilers

Andrew W. Appel, Robert Dockins, Aquinas Hobor, Lennart Beringer, Josiah Dodds,  
Gordon Stewart, Sandrine Blazy and Xavier Leroy

Frontmatter

[More information](#)

---

Cambridge University Press

978-1-107-04801-0 - Program Logics for Certified Compilers

Andrew W. Appel, Robert Dockins, Aquinas Hobor, Lennart Beringer, Josiah Dodds,

Gordon Stewart, Sandrine Blazy and Xavier Leroy

Frontmatter

[More information](#)

vii

## *Contents*

Road map	ix
Acknowledgments	x
1 Introduction	1
<b>I Generic separation logic</b>	<b>9</b>
2 Hoare logic	10
3 Separation logic	16
4 Soundness of Hoare logic	25
5 Mechanized Semantic Library	33
6 Separation algebras	35
7 Operators on separation algebras	44
8 First-order separation logic	49
9 A little case study	55
10 Covariant recursive predicates	63
11 Share accounting	69
<b>II Higher order separation logic</b>	<b>75</b>
12 Separation logic as a logic	76
13 From separation algebras to separation logic	84
14 Simplification by rewriting	89
15 Introduction to step-indexing	94
16 Predicate implication and subtyping	99
17 General recursive predicates	104
18 Case study: Separation logic with first-class functions	111

Cambridge University Press

978-1-107-04801-0 - Program Logics for Certified Compilers

Andrew W. Appel, Robert Dockins, Aquinas Hobor, Lennart Beringer, Josiah Dodds,

Gordon Stewart, Sandrine Blazy and Xavier Leroy

Frontmatter

[More information](#)

CONTENTS	viii
19 Data structures in indirection theory	123
20 Applying higher-order separation logic	130
21 Lifted separation logics	134
<b>III Separation logic for CompCert</b>	<b>141</b>
22 Verifiable C	142
23 Expressions, values, and assertions	148
24 The VST separation logic for C light	153
25 Typechecking for Verifiable C	173
26 Derived rules and proof automation for C light	184
27 Proof of a program	195
28 More C programs	208
29 Dependently typed C programs	217
30 Concurrent separation logic	222
<b>IV Operational semantics of CompCert</b>	<b>232</b>
31 CompCert	233
32 The CompCert memory model	237
33 How to specify a compiler	272
34 C light operational semantics	288
<b>V Higher-order semantic models</b>	<b>294</b>
35 Indirection theory	295
36 Case study: Lambda-calculus with references	316
37 Higher-order Hoare logic	340
38 Higher-order separation logic	347
39 Semantic models of predicates-in-the-heap	351
<b>VI Semantic model and soundness of Verifiable C</b>	<b>362</b>
40 Separation algebra for CompCert	363
41 Share models	374
42 Juicy memories	385
43 Modeling the Hoare judgment	392
44 Semantic model of CSL	401



Cambridge University Press

978-1-107-04801-0 - Program Logics for Certified Compilers

Andrew W. Appel, Robert Dockins, Aquinas Hobor, Lennart Beringer, Josiah Dodds,

Gordon Stewart, Sandrine Blazy and Xavier Leroy

Frontmatter

[More information](#)

CONTENTS	ix
45 Modular structure of the development	406
<b>VII Applications</b>	<b>410</b>
46 Foundational static analysis	411
47 Heap theorem prover	426
<b>Bibliography</b>	<b>442</b>
<b>Index</b>	<b>452</b>

## *Road map*

Readers interested in **the theory of separation logic** (with some example applications) should read Chapters 1–21. Readers interested in **the use of separation logic to verify C programs** should read Chapters 1–6 and 8–30. Those interested in **the theory of step-indexing** and **indirection theory** should read Chapters 35–39. Those interested in building models of **program logics** proved sound for **certified compilers** should read Chapters 40–47, though it would be helpful to read Chapters 1–39 as a warm-up.

Cambridge University Press

978-1-107-04801-0 - Program Logics for Certified Compilers

Andrew W. Appel, Robert Dockins, Aquinas Hobor, Lennart Beringer, Josiah Dodds,  
Gordon Stewart, Sandrine Blazy and Xavier Leroy

Frontmatter

[More information](#)

---

x

## *Acknowledgments*

I thank Jean-Jacques Lévy for hosting my visit to INRIA Rocquencourt 2005–06, during which time I started thinking about the research described in this book. I enjoyed research collaborations during that time with Francesco Zappa Nardelli, Sandrine Blazy, Paul-André Melliès, and Jérôme Vouillon.

I thank the scientific team that built and maintains the Coq proof assistant, and I thank INRIA and the research funding establishment of France for supporting the development of Coq over more than two decades.

Mario Alvarez and Margo Flynn provided useful feedback on the usability of VST 0.9.

Research funding for some of the scientific results described in this book was provided by the Air Force Office of Scientific Research (agreement FA9550-09-1-0138), the National Science Foundation (grant CNS-0910448), and the Defense Advanced Research Projects Agency (agreement FA8750-12-2-0293). The views and conclusions contained herein are those of the authors and should not be interpreted as necessarily representing the official policies or endorsements, either expressed or implied, of AFOSR, NSF, DARPA, or the U.S. government.