1. Record Nr.   UNINA9910557461103321

   Autore   Chlipala Adam <1981->

   Titolo   Certified programming with dependent types : a pragmatic introduction
                    to the Coq proof assistant / / Adam Chlipala

   Pubbl/distr/stampa   Cambridge, : The MIT Press, 2013
                        Cambridge, Massachusetts : , : MIT Press, , [2013]
                        [Piscataqay, New Jersey] : , : IEEE Xplore, , [2013]

   Descrizione fisica   1 PDF (xii, 424 pages)

   Collana   The MIT Press

   Disciplina   005.1

   Soggetti   Automatic theorem proving - Computer programs
                    Computer programming

   Lingua di pubblicazione   Inglese

   Formato   Materiale a stampa

   Livello bibliografico   Monografia

   Note generali   Bibliographic Level Mode of Issuance: Monograph

   Nota di bibliografia   Includes bibliographical references (pages [413]-417) and index.

   Sommario/riassunto   The technology of mechanized program verification can play a
                    supporting role in many kinds of research projects in computer science,
                    and related tools for formal proof-checking are seeing increasing
                    adoption in mathematics and engineering. This book provides an
                    introduction to the Coq software for writing and checking mathematical
                    proofs. It takes a practical engineering focus throughout, emphasizing
                    techniques that will help users to build, understand, and maintain large
                    Coq developments and minimize the cost of code change over time.
                    Two topics, rarely discussed elsewhere, are covered in detail: effective
                    dependently typed programming (making productive use of a feature at
                    the heart of the Coq system) and construction of domain-specific proof
                    tactics. Almost every subject covered is also relevant to interactive
                    computer theorem proving in general, not just program verification,
                    demonstrated through examples of verified programs applied in many
                    different sorts of formalizations. The book develops a unique
                    automated proof style and applies it throughout; even experienced Coq
                    users may benefit from reading about basic Coq concepts from this
                    novel perspective. The book also offers a library of tactics, or programs
                    that find proofs, designed for use with examples in the book. Readers

will acquire the necessary skills to reimplement these tactics in other settings by the end of the book. All of the code appearing in the book is freely available online.