Nantes Université

HABILITATION À DIRIGER DES RECHERCHES
HDR

NANTES UNIVERSITE

ECOLE DOCTORALE N° 601
Mathématiques et Sciences et Technologies de l'Information et de la Communication
Spécialité : *Informatique*

Par
Matthieu SOZEAU

Mécanisation de la Métathéorie et Compilation Certifiée pour le prouveur Rocq

Travaux présentés et soutenus à Nantes, le 12 Février 2026
Unité de recherche : LS2N (UMR 6004)

Rapporteurs avant soutenance :

Sylvain	BOULMÉ	Maître de Conférences, Université Grenoble Alpes
François	POTTIER	Directeur de Recherche, Centre Inria de Paris
Alan	SCHMITT	Directeur de Recherche, Centre Inria de l'Université de Rennes

Composition du Jury :

Président :			
Examinateurs :	Assia	MAHBOUBI	Directrice de Recherche, Inria & Université de Nantes
	David	MONNIAUX	Directeur de Recherche, CNRS, Université Grenoble Alpes

Nantes Université

Titre : Mécanisation de la Métathéorie et Compilation Certifiée pour le prouveur Rocq

Mots clés : théorie des types, métathéorie, vérification

Résumé : Les travaux présentés dans cette habilitation à diriger les recherches (HDR) ont pour objet la vérification du noyau logique de l'assistant de preuve Rocq ainsi que de son mécanisme original de compilation vers des langages fonctionnels usuels. L'objectif de ces travaux est d'obtenir des garanties formelles de sûreté sur la théorie sous-jacente du prouveur Rocq, sur l'implémentation du vérificateur de preuves pour cette théorie et sur la procédure d'extraction de programmes permettant l'exécution de code vérifié par Rocq.

Nous présentons MetaRocq, un développement formel de spécifications et programmes vérifiés, représentant un large fragment de Rocq dans Rocq même, combinant des techniques de méta-programmation et de preuve de métathéorie des langages à grande échelle ainsi que des techniques de compilation certifiée.

Nous développons formellement dans Rocq une spécification formelle de la théorie des types de Rocq et sa variante algorithmique équivalente, un algorithme de typage correct et complet vis-à-vis de ces spécifications, un algorithme d'effacemen des preuves correct et une chaine de compilation de Rocq vers OCaml préservant la sémantique du langage source.

Les résultats obtenus nous permettent de changer de paradigme quand à la sûreté des résultats obtenus dans l'assistant de preuve et produits par l'extraction, passant d'une notion de base de code fiable - le noyau non vérifié actuel - à une notion de base de théorie fiable - la spécification formelle de la théorie de Rocq en Rocq que nous avons introduit.

Nous discutons l'impact et les limites de nos travaux pour l'obtention de garanties maximales sur les développements réalisés dans Rocq.

Title : Mechanized metatheory and certified compilation for The Rocq Prover

Keywords : type theory, metatheory, verification

Abstract : The work presented in this habilitation thesis focuses on the verification of the logical core of the Rocq Prover, as well as its original mechanism for compiling to usual functional programming languages. The objective of this work is to obtain formal safety guarantees for the underlying theory of the Rocq prover, for the implementation of the proof verifier for this theory, and for the procedure of program extraction that enables the efficient execution of code verified by Rocq. This work lies at the intersection of the meta-theory of type theory and the formal verification of type inference and compilation algorithms.

We present MetaRocq, a formal development of specifications and verified programs, representing a large fragment of Rocq within Rocq itself. This combines techniques for meta-programming and large-scale language meta-theory proofs, as well as certified compilation techniques.

We formally develop in Rocq a formal specification of Rocq's type theory and its equivalent algorithmic variant, a typing algorithm that is correct and complete with respect to these specifications, a correct proof erasure algorithm, and a compilation chain from Rocq to OCaml that preserves the semantics of the source language.

The results obtained here allow us to shift the paradigm regarding the safety of the results obtained in the proof assistant and produced by extraction, moving from a notion of a trusted code base — the current unverified kernel — to a notion of trusted theory base — the formal specification of Rocq's theory in Rocq that we have introduced.

We discuss the impact and limitations of our work to obtain maximal correctness guarantees for the formal developments carried out in the proof assistant.