UNIVERSITE BRETAGNE LOIRE / MATHSTIC

UNIVERSITÉ DE NANTES

THÈSE DE DOCTORAT DE

L'UNIVERSITÉ DE NANTES
COMUE UNIVERSITÉ BRETAGNE LOIRE

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

Par
Théo WINTERHALTER

Formalisation and Meta-Theory of Type Theory

Thèse présentée et soutenue à l'Université de Nantes, le 18 septembre 2020
Unité de recherche : LS2N

Rapporteurs avant soutenance :

Andrej BAUER — Professor, University of Ljubljana
Herman GEUVERS — Professor, Radboud University Nijmegen

Composition du Jury :

Président :	Gilles DOWEK	Directeur de recherche, Inria
Examinateurs :	Hugo HERBELIN	Directeur de recherche, Inria
	Assia MAHBOUBI	Chargée de recherche, Inria Rennes
	Bas SPITTERS	Associate professor, Aarhus University
Dir. de thèse :	Nicolas TABAREAU	Directeur de recherche, Inria Rennes
Co-dir. de thèse :	Matthieu SOZEAU	Chargé de recherche, Inria Rennes

Invité(s) :
Andreas ABEL — Professor, Chalmers / Göteborg University

UNIVERSITE BRETAGNE LOIRE / MATHSTIC

UNIVERSITÉ DE NANTES

Titre : Formalisation et Méta-Théorie de la Théorie des Types

Mot clés : théorie des types, formalisation, traduction de programme, vérification de type

Résumé : Dans cette thèse, je parle de la méta-théorie de la théorie des types et de la façon de la formaliser dans un assistant de preuve.

Je me concentre d'abord sur une traduction *conservative* de la théorie des types extensionnelle vers la théorie des types intensionnelle ou faible, entièrement écrite en Coq. La première traduction consiste en une suppression de la règle de *reflection* de l'égalité, tandis que la deuxième traduction produit quelque chose de plus fort : la théorie des types faibles est une théorie des types sans notion de conversion. Le résultat de conservativité implique que la conversion n'augmente pas la puissance logique de la théorie des types.

Ensuite, je montre ma contribution au projet MetaCoq de formalisation et de spécification de Coq au sein de Coq. En particulier, j'ai travaillé sur l'implantation d'un vérificateur de type pour Coq, en Coq. Ce vérificateur de type est prouvé correct vis à vis de la spécification et peut être extrait en code OCaml et exécuté indépendamment du vérificateur de type du noyau de Coq. Pour que cela fonctionne, nous devons nous appuyer sur la méta-théorie de Coq que nous développons, en partie, dans le projet MetaCoq. Cependant, en raison des théorèmes d'incomplétude de Gödel, nous ne pouvons pas prouver la cohérence de Coq dans Coq, ce qui signifie que certaines propriétés — principalement la forte normalisation — doivent être supposées, c'est-à-dire prises comme axiomes.

Title: Formalisation and Meta-Theory of Type Theory

Keywords: type theory, extensionality, program translation, type checking

Abstract: In this thesis, I talk about the meta-theory of type theory and about how to formalise it in a proof assistant.

I first focus on a conservative translation between extensional type theory and either intensional or weak type theory, entierely written in Coq. The first translation consists in a removal of the reflection of equality rule, whereas the second translation produces something stronger: weak type theory is a type theory with no notion of conversion. The conservativity result implies that conversion doesn't increase the logical power of type theories.

Then, I show my work for the Meta-Coq project of formalising and specifying Coq within Coq. In particular I worked on writing a type-checker for Coq, in Coq. This type checker is proven sound with respect to the specification and can be extracted to OCaml code and run independently of Coq's kernel type-checker. For this to work we have to rely on the meta-theory of Coq which we develop, in part, in the MetaCoq project. However, because of Gödel's incompleteness theorems, we cannot prove consistency of Coq within Coq, and this means that some properties—mainly strong normalisation—have to be assumed, *i.e.* taken as axioms.