COLLEGE ED DOCTORAL MATHEMATIQUES PAYS DE LA LOIRE STIC

Nantes Université

THÈSE DE DOCTORAT DE

NANTES UNIVERSITÉ

ÉCOLE DOCTORALE N° 641
Mathématiques et Sciences et Technologies de l'Information et de la Communication
Spécialité : *Informatique*

Par
Josselin POIRET

New Sorts for Martin-Löf Type Theory

Thèse présentée et soutenue à Nantes, le 10 novembre 2026
Unité de recherche : Laboratoire des Sciences du Numérique de Nantes

Rapporteurs avant soutenance :

Stephanie WEIRICH — Professor, University of Pennsylvania
Thierry COQUAND — Professor, University of Gothenburg

Composition du Jury :

Attention, en cas d'absence d'un des membres du Jury le jour de la soutenance, la composition du jury doit être revue pour s'assurer qu'elle est conforme et devra être répercutée sur la couverture de thèse

Président :	Prénom NOM	Fonction et établissement d'exercice *(à préciser après la soutenance)*
Examinateurs :	Daniel GRATZER	Associate professor, Aarhus University
	Yves BERTOT	Directeur de recherche, Centre Inria d'Université Côte d'Azur
	Gabriel SCHERER	Chargé de recherche, Inria Paris
	Ambrus KAPOSI	Associate professor, Eötvös Loránd University
Dir. de thèse :	Nicolas TABAREAU	Directeur de recherche, Inria Rennes
Co-dir. de thèse :	Kenji MAILLARD	Inria Starting Faculty Positions, Inria Rennes

COLLEGE ED
DOCTORAL MATHEMATIQUES
PAYS DE LA LOIRE STIC

Nantes Université

Titre : De nouvelles sortes pour la théorie des types de Martin-Löf

Mot clés : Théorie des types dépendants, Logique constructive, Assistants de preuve, Preuve formalisée, Langages de programmation

Résumé : Grâce à la synthèse entre les traditions de la logique constructive et de la théorie de la calculabilité, les assistants de preuve ont permis à leurs utilisateur de vérifier formellement leurs théorie, preuves et code, et commencent même à être utilisés dès l'étape de conception de nouvelles idées. Ces applications ont été rendues possibles grâce aux propriétés fortes que possèdes les languages sous-jacents à ces logiciels, dont l'étude est le cœur du domaine de la théorie des types dépendants. Cependant, avec l'introduction récente de nombreuses nouvelles théories des types, ainsi que la sophistication croissante des implémentations d'assistants de preuves, il est devenu apparent que les preuves sur papier traditionelles de ces invariants ne suffisent plus, et qu'elles devaient ainsi être également formalisée dans un assistant de preuve. De plus, ces formalisations doivent pouvoir passer à l'échelle, afin de pouvoir traiter un grand nombre de variations de théorie des types.

Cette thèse est la tentative d'une telle formalisation, dans laquelle nous commencerons par étudier une version de la théorie des types de Martin-Löf, afin de prouver ses propriétés principales, avant de l'étendre pour accommoder de nouvelles sortes avec des comportements non-standards.

Title: New Sorts for Martin-Löf Type Theory

Keywords: Dependent type theory, Constructive logic, Proof assistants, Formalized mathematics, Programming languages

Abstract: By fusing long-standing traditions in constructive logic and computability, proof assistants have helped users formally verify their theories, proof and code, and are starting to serve as guardrails in the elaboration of new ideas. These applications are possible thanks to the specific properties of the underlying languages these proof assistants implement, which is at the heart of the field of dependent type theory. With a recent proliferation of new type theories, as well as the increasing complexity of proof assistant implementations, it has become clear that pen-and-paper proofs of those guarantees are not enough, and that they should also be verified using a proof assistant. Furthermore, the proof techniques must be scalable, so that it is indeed possible to do so for a large number of variations of type theories.

This thesis is an attempt of such a formalization, in which we will first study a version of Martin-Löf type theory and prove its main properties, before extending it to support new sorts with non-standard behavior.