



**HAL**  
open science

# MetaCoq: de la métaprogrammation à l'extraction certifiée pour Coq

Matthieu Sozeau

► **To cite this version:**

Matthieu Sozeau. MetaCoq: de la métaprogrammation à l'extraction certifiée pour Coq. 35es Journées Francophones des Langages Applicatifs (JFLA 2024), Jan 2024, Saint-Jacut-de-la-Mer, France. hal-04407164

**HAL Id: hal-04407164**

**<https://inria.hal.science/hal-04407164v1>**

Submitted on 20 Jan 2024

**HAL** is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.



Distributed under a Creative Commons Attribution 4.0 International License

# MetaCoq : de la métaprogrammation à l'extraction certifiée pour Coq

Matthieu Sozeau

Inria Rennes – Bretagne Atlantique, LS2N

MetaCoq est une librairie qui a pour but d'étudier la théorie et l'implémentation de Coq lui-même, en Coq. MetaCoq comprend quatre composants principaux. Le premier, nommé Template-Coq, est une réification en Coq des termes et environnements manipulés par le noyau de Coq (en OCaml), qui permet d'interopérer directement avec le système en développant des métaprogrammes à l'aide du langage Gallina et d'une monade représentant l'interface du noyau de Coq. Le second, PCUIC, est une formalisation d'une variante du système de types du calcul de constructions inductives et de ses propriétés métathéoriques essentielles, qui permet de garantir la correction du système et d'étudier ses extensions possibles. Le troisième composant est une version certifiée d'un vérificateur de types et d'environnements pour ce calcul, garantissant la décidabilité de l'inférence de types et sa correction et complétude vis-à-vis de la spécification de PCUIC. Enfin, le quatrième composant est un développement certifié de la procédure d'effacement des preuves et des types de PCUIC vers un lambda-calcul pur, qui forme le coeur du mécanisme d'extraction de Coq, permettant en particulier d'extraire le vérificateur de types vers des langages exécutables de façon sûre. Au cours de cet exposé, je présenterai un tour d'horizon de ces quatre composants et leur utilisation pour la production de programmes certifiés en Coq avec des garanties de sûreté maximales.

## Références

[Tea] The MetaCoq TEAM : The MetaCoq Project. <https://metacoq.github.io/>.