POPL24

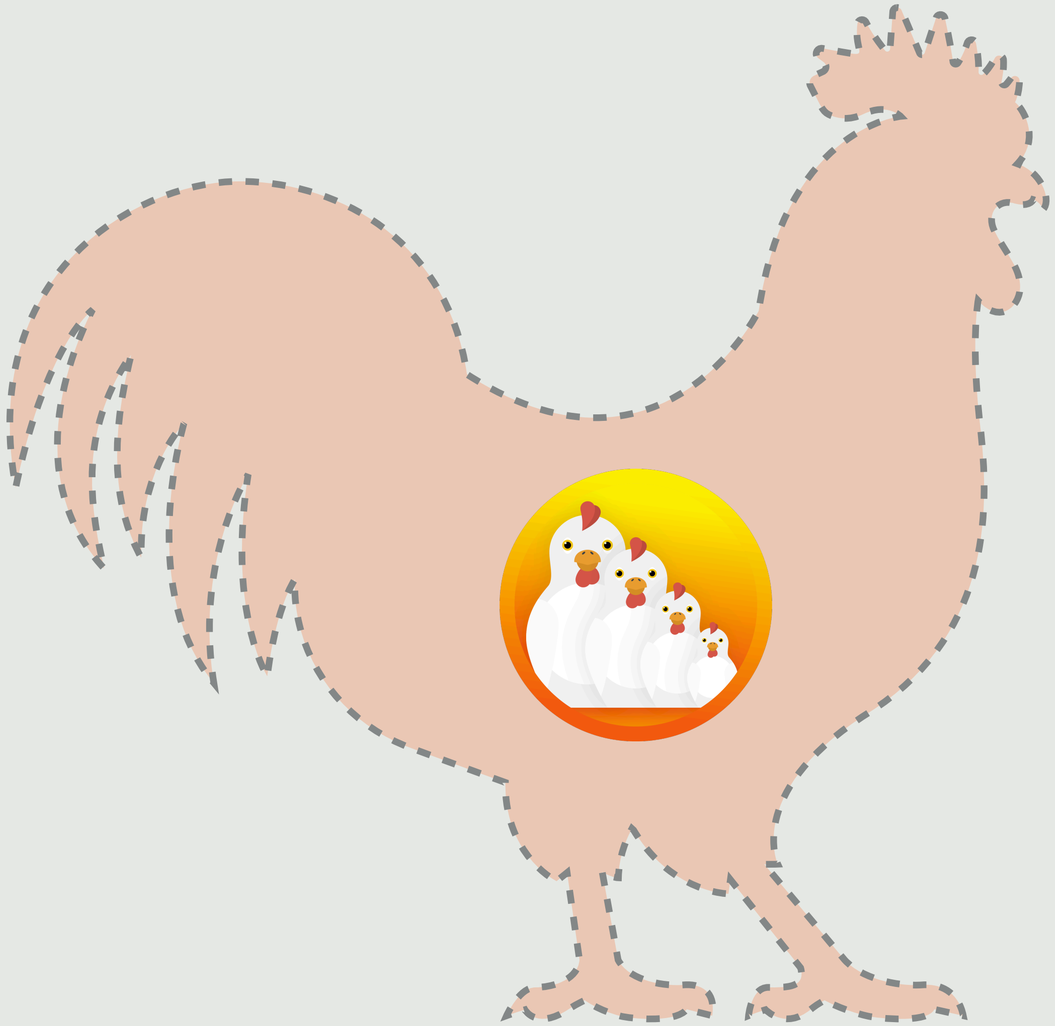
MetaCoq Tutorial

Yannick **Forster** Meven **Lennon-Bertrand**

Matthieu **Sozeau** Théo **Winterhalter**

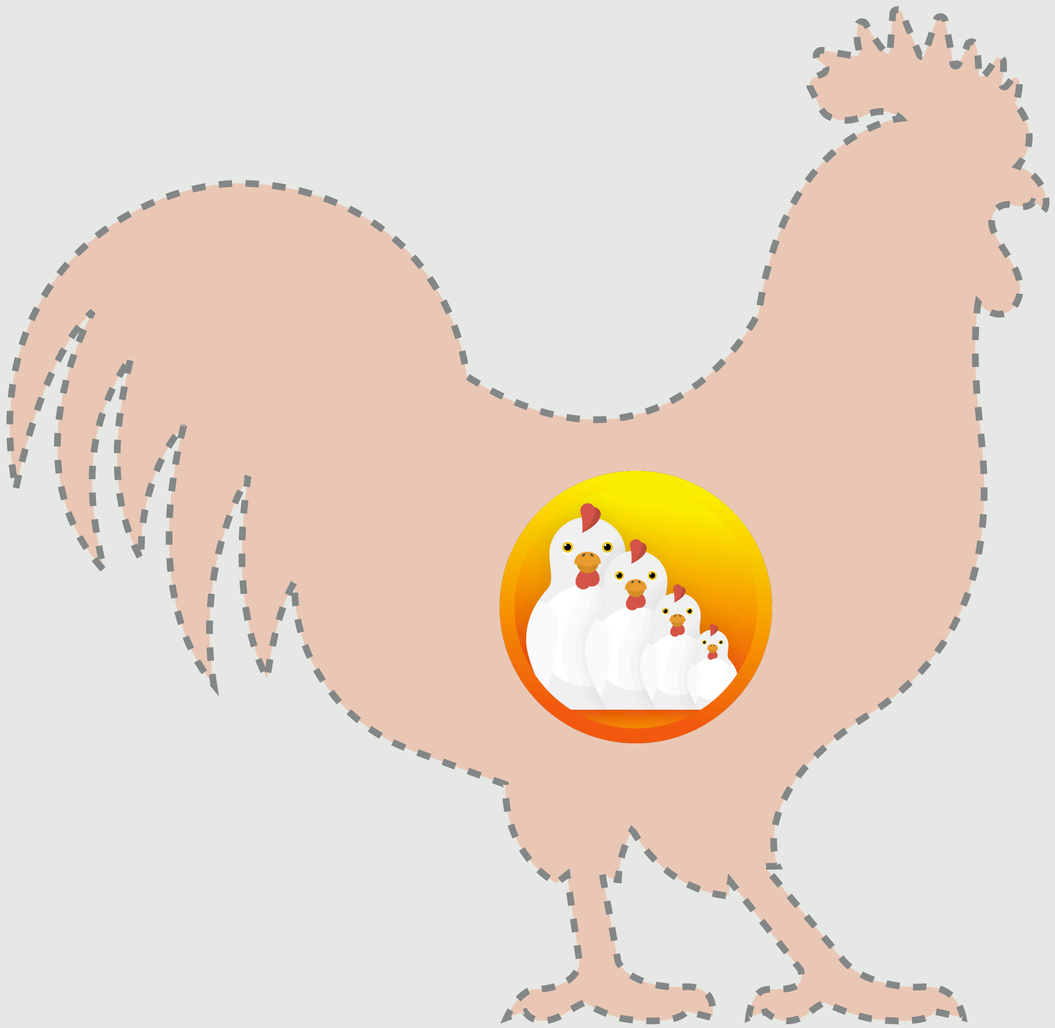
14 January 2024

What is MetaCoq?



Basically… Coq in Coq

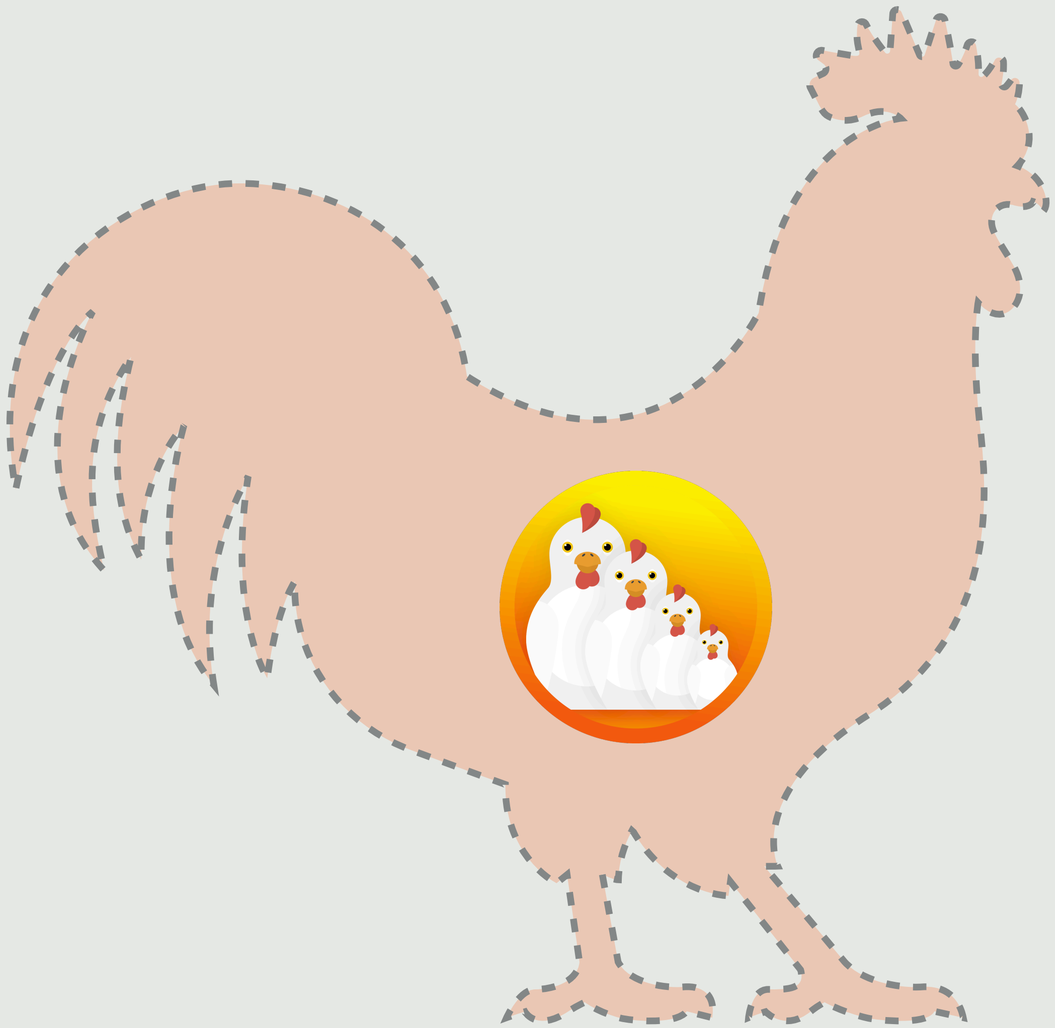
What is MetaCoq?



Basically… Coq in Coq

Manipulate Coq syntax… meta-programming
(main focus of this tutorial)

What is MetaCoq?



Basically… Coq in Coq

Manipulate Coq syntax… meta-programming
(main focus of this tutorial)

Reason about Coq typing…
certified meta-programs, plugins…

Meta-programming with Coq

MetaCoq is both a library

Representation of Coq terms in Coq

```
Inductive term :=
| tRel : nat → term
| tApp : term → list term → term
| …
```

Meta-programming with Coq

MetaCoq is both a library and a plugin

Representation of Coq terms in Coq

```
Inductive term :=
| tRel : nat → term
| tApp : term → list term → term
| …
```

Coq terms

```
(λ x. x) 0
```

quote

unquote

```
tApp (tLambda _ _ _) [tConstruct _ _ _]
```

Meta-programming with Coq

MetaCoq is both a library and a plugin

Coq terms

```
(λ x. x) 0
```

quote

unquote

Representation of Coq terms in Coq

```
Inductive term :=
| tRel : nat → term
| tApp : term → list term → term
| …
```

```
tApp (tLambda _ _ _) [tConstruct _ _ _]
```

Representation of commands

```
tmDefinition “foo” 24
```

Meta-programming with Coq

MetaCoq is both a library and a plugin

Coq terms

```
(λ x. x) 0
```

quote

unquote

Representation of Coq terms in Coq

```
Inductive term :=
| tRel : nat → term
| tApp : term → list term → term
| …
```

```
tApp (tLambda _ _ _) [tConstruct _ _ _]
```

Representation of commands

```
tmDefinition “foo” 24
```

run

```
Definition foo := 24
```

Meta-programming with MetaCoq

Examples

Meta-programming with MetaCoq

Examples

Autosubst 2

Meta-programming with MetaCoq

Examples

Autosubst 2

Induction principle and subterm relation generation

Meta-programming with MetaCoq

Examples

Autosubst 2

Induction principle and subterm relation generation

Parametricity translation

Meta-programming with MetaCoq

Examples

Autosubst 2

Induction principle and subterm relation generation

Parametricity translation

Other meta-programming tools exist

Coq ELPI, Ltac2, OCaml

Meta-programming with MetaCoq

Examples

Autosubst 2

Induction principle and subterm relation generation

Parametricity translation

MetaCoq is just Coq!

we can prove things

Other meta-programming tools exist

Coq ELPI, Ltac2, OCaml

Certified Meta-programming with MetaCoq

$$\Gamma \vdash t : A$$

Typing

$$\Gamma \vdash u \twoheadrightarrow v$$

Reduction

$$\Gamma \vdash u \equiv v$$

Conversion

all defined / specified within MetaCoq

Example: proving parametricity preserves typing becomes possible (no more de Bruijn bugs)

Certified Meta-programming with MetaCoq

$$\Gamma \vdash t : A$$

Typing

$$\Gamma \vdash u \twoheadrightarrow v$$

Reduction

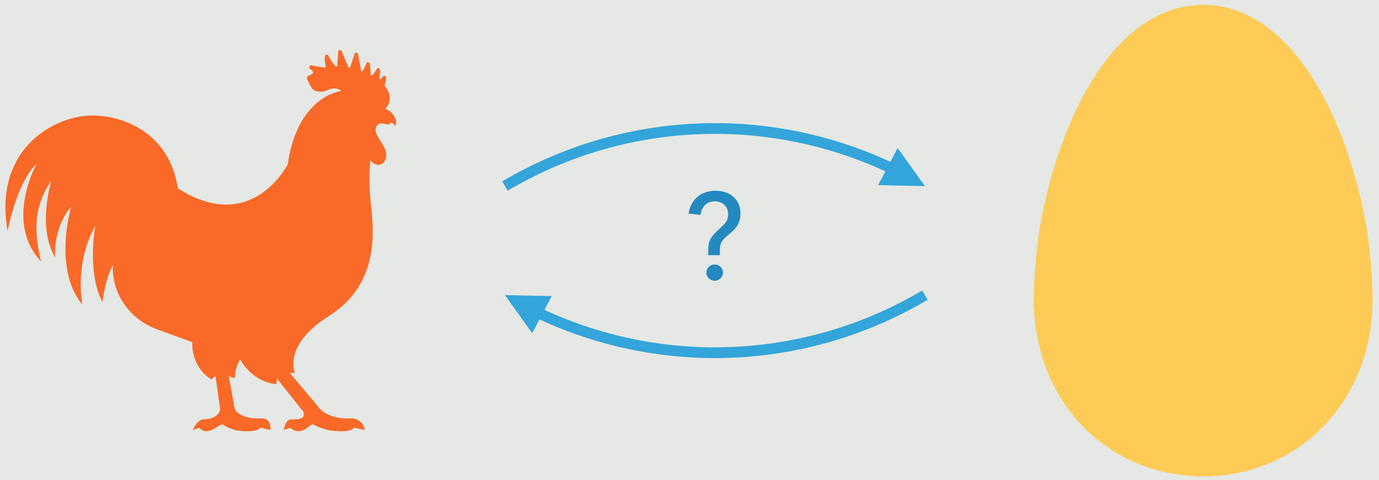
$$\Gamma \vdash u \equiv v$$

Conversion

all defined / specified within MetaCoq

Example: proving parametricity preserves typing becomes possible (no more de Bruijn bugs)

We can certify notorious meta-programs like the type checker!

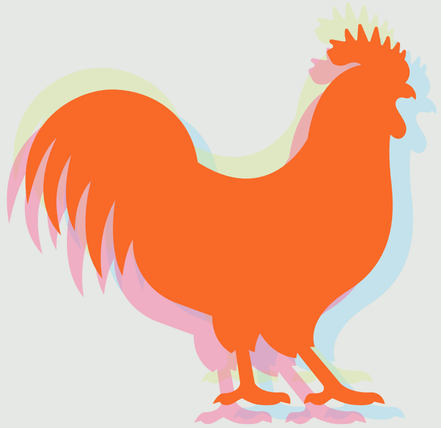


Ideal Coq

Coq kernel

Reference manual

Papers + Theses



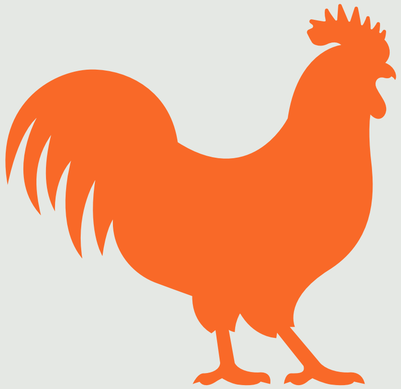
Ideal Coq

underspecified

Coq kernel

Reference manual

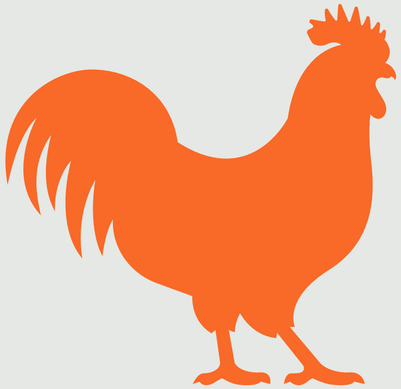
Papers + Theses



?

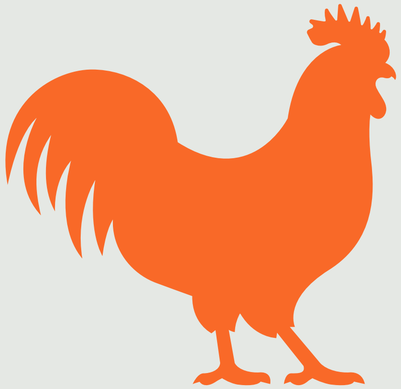
PCUIC

Coq kernel



PCUIC

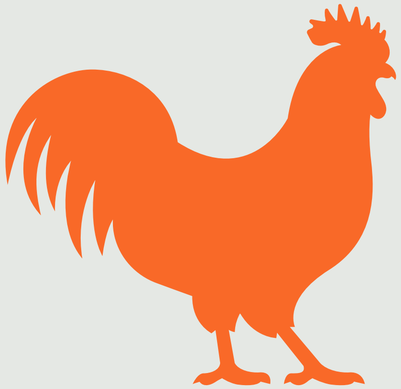
Coq kernel



PCUIC

Specified

Coq kernel



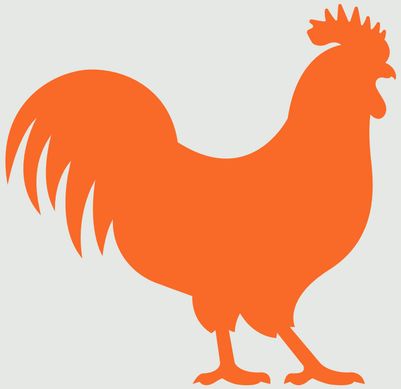
PCUIC

Specified

Coq kernel

✓ Verified

Trusted core



PCUIC

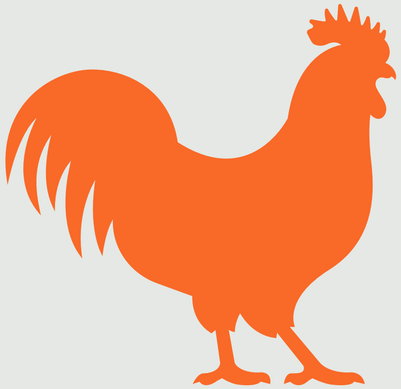
Coq kernel

Specified

Verified

Trusted theory

Trusted core



PCUIC

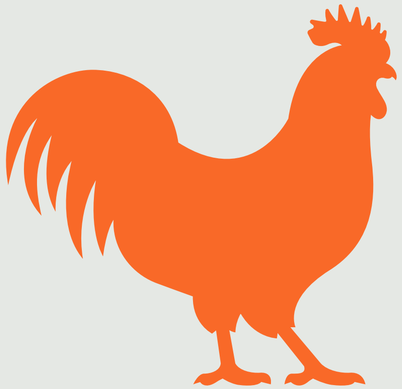
Coq kernel

Specified

Verified

Trusted theory

Trusted core



Type Checker

Verified Erasure + Extraction

External Checker

PCUIC

Coq kernel

Specified

Verified

```
MetaCoq Check foo.
```

Goal of this tutorial

Overview of the MetaCoq project

Teach you how to write simple meta-programs

Plan for today

The MetaCoq formalisations (45 min)

Template monad / meta-programming (45 min)

Exercise session (60 min)

Practical information

Install **Coq 8.17**
coq.inria.fr/download
We recommend **Coq Platform**

In general, get help on **Zulip**:
coq.zulipchat.com
And then use the **MetaCoq stream**

Get **MetaCoq**
github.com/MetaCoq/metacoq
It comes with the Coq Platform

Choose an editor
coq.inria.fr/user-interfaces.html
We suggest **VSCoq legacy** or **CoqIDE**

More info on the MetaCoq wiki
github.com/MetaCoq/metacoq/wiki/
(on the “MetaCoq tutorial” page)