



# An Equations Tutorial

Matthieu Sozeau  
Inria

CoqPL 2020  
New Orleans, LA, USA  
January 25th 2020

# Equations

- ▶ High-level function definitions by dependent pattern-matching and recursion.
- ▶ Supports where clauses and pattern-matching lambdas.
- ▶ Improved no-confusion principle for « forced » arguments

```
Inductive fin : nat → Set :=  
| fz : ∀ n : nat, fin (S n)  
| fs : ∀ n : nat, fin n → fin (S n).
```

- ▶ Mutual and well-founded recursive definitions

# Equations

- ▶ Compiled down to axiom-free Coq terms (not relying on UIP/K)
- ▶ Compatible with equality in **Prop** or **Type** (e.g. HoTT/Coq)
- ▶ Functional elimination principle synthesizing the recursion and pattern-matching behavior of the function(s).
- ▶ Also provides a saner dependent elimination tactic

# Equations

[mattam82.github.io/Coq-Equations/](https://mattam82.github.io/Coq-Equations/)

```
# opam install coq-equations
```

Gallery of programs:

- ▶ Nested and mutual well-founded recursion on inductive families
- ▶ Using views
- ▶ A reflexive polynomial solving tactic (« mini-ring »)
- ▶ Port of CPDT (Chlipala) chapters
- ▶ Using proven instances of UIP in proofs

# Coming up

[mattam82.github.io/Coq-Equations/](https://mattam82.github.io/Coq-Equations/)

- ▶ Copattern-matching
- ▶ Interactive Refinement Mode in IDEs
- ▶ Internal Size-Change Termination Principle
- ▶ A cleaner whitespace-sensitive syntax?
- ▶ Integration of no-confusion/dependent elimination in intro-patterns
- ▶ Replacing for the Function tool