Introduction to Coq

Coq Andes Summer School 2020

Assia Mahboubi, Inria – VU Amsterdam

A Versatile Prover

A Versatile Prover

- Program Verification
  - Compcert compiler [Leroy et al. 2009]
  - Synthesis of cryptographic primitives [Ebsen et al. 2019]
  - ...

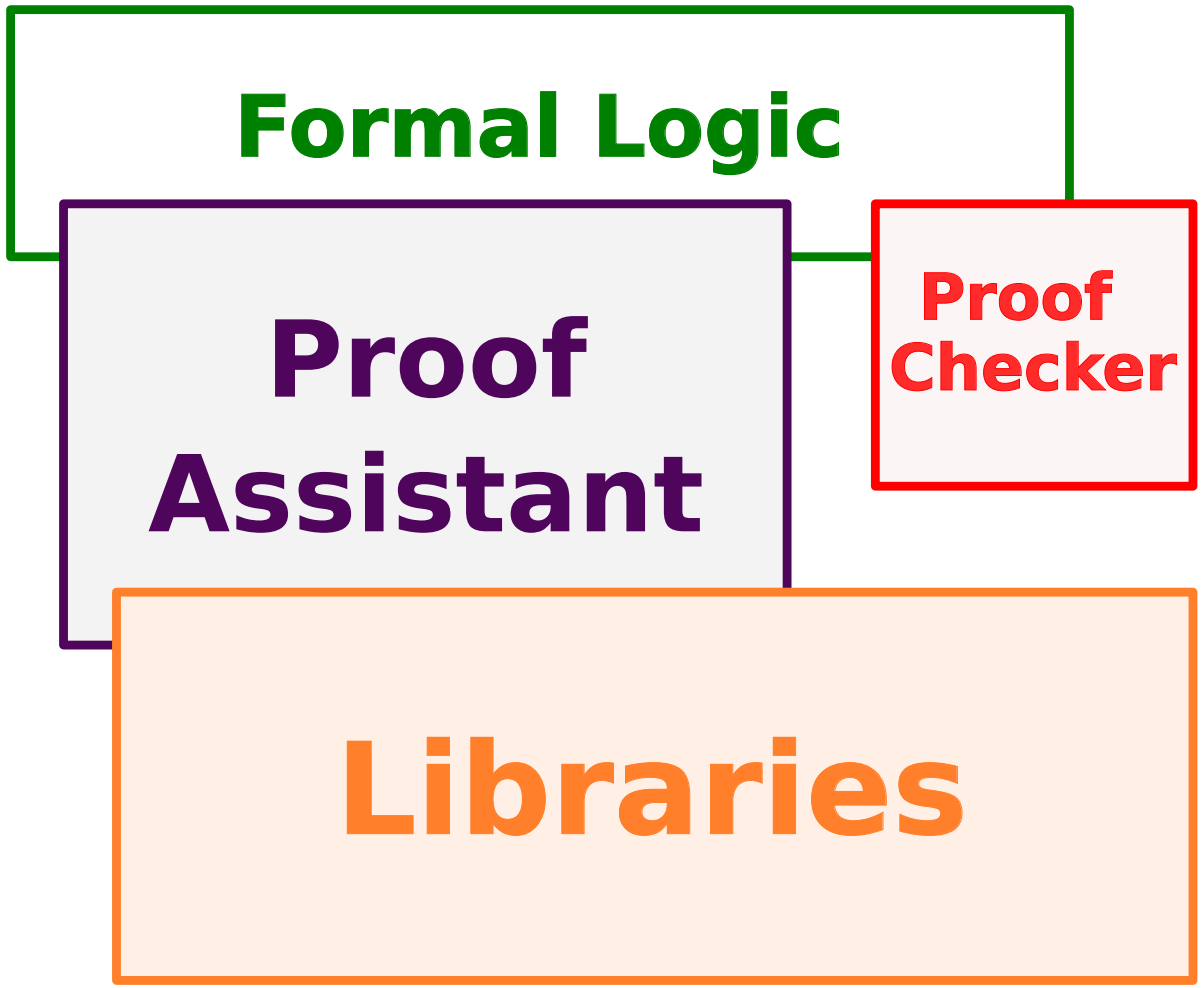
A Versatile Prover

- Program Verification
  - Compcert compiler [Leroy et al. 2009]
  - Synthesis of cryptographic primitives [Ebsen et al. 2019]
  - ...
- Formalized Mathematics
  - Feit-Thompson theorem [Gonthier et al. 2012]
  - Homotopy Type Theory [Univalent Foundations Program 2013]
  - ...

A Versatile Prover

- Program Verification
  - Compcert compiler [Leroy et al. 2009]
  - Synthesis of cryptographic primitives [Ebsen et al. 2019]
  - ...
- Formalized Mathematics
  - Feit-Thompson theorem [Gonthier et al. 2012]
  - Homotopy Type Theory [Univalent Foundations Program 2013]
  - ...
- Teaching
  - Software Foundations
  - ...

Interactive Theorem Provers



Dependent Type Theory

Coq, Agda, Matita, Lean, NuPRL, etc.

- First prototype by Thierry Coquand and Gérard Huet
  circa 1984

Dependent Type Theory

Coq, Agda, Matita, Lean, NuPRL, etc.

- First prototype by Thierry Coquand and Gérard Huet
  circa 1984
- Intuitionistic higher-order logic, dependent types, inductives
  inspired by de Bruijn, Martin Löf, Girard

Dependent Type Theory

Coq, Agda, Matita, Lean, NuPRL, etc.

- First prototype by Thierry Coquand and Gérard Huet
  circa 1984
- Intuitionistic higher-order logic, dependent types, inductives
  inspired by de Bruijn, Martin Löf, Girard
- Pure functional programs in the logic

Dependent Type Theory

Coq, Agda, Matita, Lean, NuPRL, etc.

- First prototype by Thierry Coquand and Gérard Huet
  circa 1984
- Intuitionistic higher-order logic, dependent types, inductives
  inspired by de Bruijn, Martin Löf, Girard
- Pure functional programs in the logic
- Extraction

Dependent Type Theory

Coq, Agda, Matita, Lean, NuPRL, etc.

- First prototype by Thierry Coquand and Gérard Huet
  circa 1984
- Intuitionistic higher-order logic, dependent types, inductives
  inspired by de Bruijn, Martin Löf, Girard
- Pure functional programs in the logic
- Extraction

Languages

- Logical foundations
- Libraries
- Meta-Language(s)

Two Sides of Automation

Filling gaps:

Two Sides of Automation

Filling gaps:

- in proofs

Two Sides of Automation

Filling gaps:

- in proofs
- in statements.