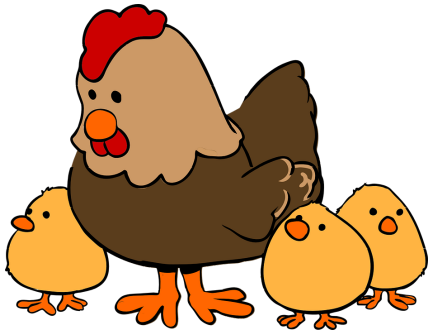


Agda,
a *beautiful* proof assistant

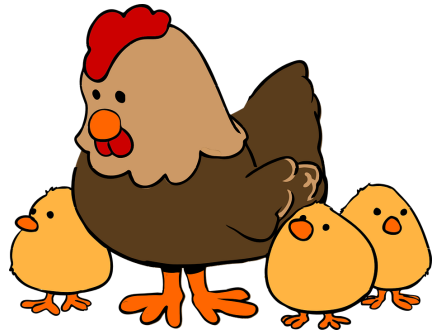
second part of the course TEORIA DEI TIPI • begin April 29th, 2020 • by Ingo Blechschmidt

Agda is …

1. a programming language
2. a **proof language**

```
File  Edit  Options  Buffers  Tools  Agda  Help
Save  Undo
data ℕ : Set where
  zero : ℕ
  succ : ℕ → ℕ

add : ℕ → ℕ → ℕ
add n zero     = n
add n (succ k) = succ (add n k)
-U:**-  bz.agda       All L7    <I>   (Agda:Checked +4 Undo-Tree)
```



Agda, a *beautiful* proof assistant

second part of the course TEORIA DEI TIPI • begin April 29th, 2020 • by Ingo Blechschmidt

Agda is …

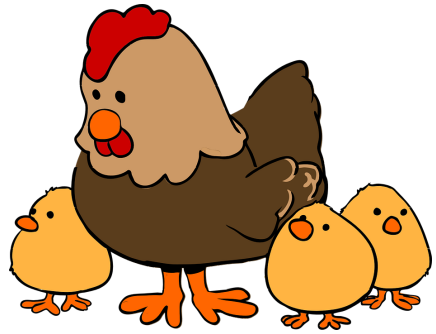
1. a programming language
2. a **proof language**

With Agda you can …

1. ensure **correctness** of proofs
2. **practically explore** type theory
3. appreciate mathematics **from a new point of view**
4. **verify correctness** of programs

Three mottos:

1. "proving = programming"
2. "conjecturing = specifying"
3. "induction = recursion"



Agda, a *beautiful* proof assistant

second part of the course TEORIA DEI TIPI • begin April 29th, 2020 • by Ingo Blechschmidt

Agda is …

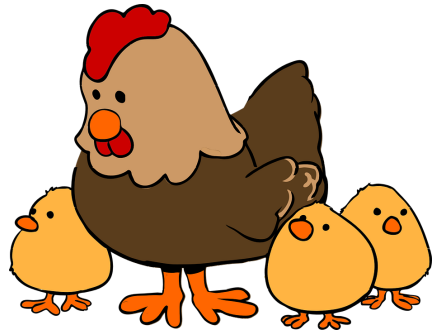
1. a programming language
2. a **proof language**

With Agda you can …

1. ensure **correctness** of proofs
2. **practically explore** type theory
3. appreciate mathematics **from a new point of view**
4. **verify correctness** of programs

Three mottos:

1. "proving = programming"
2. "conjecturing = specifying"
3. "induction = recursion"



Agda, a *beautiful* proof assistant

second part of the course TEORIA DEI TIPI • begin April 29th, 2020 • by Ingo Blechschmidt

Agda is …

1. a programming language
2. a **proof language**
3. somewhat hard to learn on one's own
4. **fun and easy** to learn as part of a course

With Agda you can …

1. ensure **correctness** of proofs
2. **practically explore** type theory
3. appreciate mathematics **from a new point of view**
4. **verify correctness** of programs

University Director

The University Director is the head of the University

University Director Catarina Coquand

Administrative Support at Halmstad University – all administration, support and service to the University's management, education and research.

UPDATED

2020-01-31

CONTACT

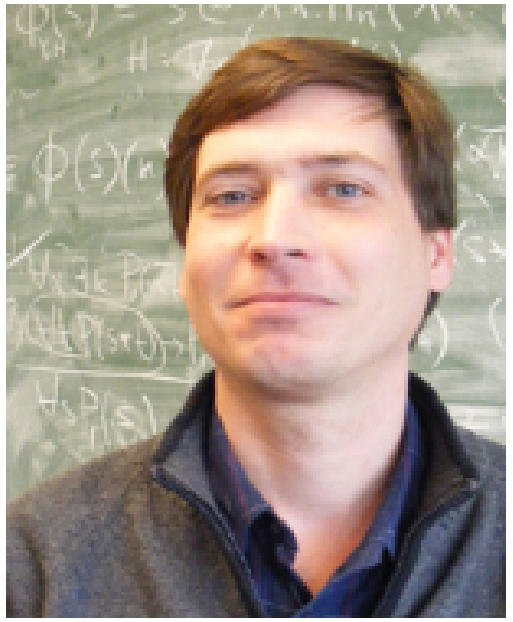
Catarina Coquand

SHARE

CONTACT

Catarina Coquand

Martín Hötzel Escardó



Also known as Martin Escardo.

Reader in Theoretical Computer Science
School of Computer Science, University of Birmingham
Birmingham B15 2TT, UK

New: Introduction to HoTT/UF with Agda.

---

(1) Timetable, (2) teaching, (3) Published and unpublished work, (4) research talks, (5) cv.

Research interests: Topology, topology in higher-type computation, constructive mathematics, dependent type theory, univalent type theory, homotopy type theory, domain theory, locale theory, exact real-number computation. My research often stumbles upon category theory, proof theory and game theory. (Dependent) functional programming is a useful and enjoyable tool for practical manifestations of theoretical ideas in computation.

Autumn school "Proof and Computation", 16th to 22nd September 2018

An international autumn school "Proof and Computation" will be held from 16th to 22nd September 2018 at Aurachhof in Fischbachau near Munich. Its aim is to bring together young researchers in the field of Foundations of Mathematics, Computer Science and Philosophy.

Scope

- Predicative Foundations
- Constructive Mathematics and Type Theory
- Computation in Higher Types
- Extraction of Programs from Proofs

Courses

- Ulrich Berger on Program Extraction from Proofs
- Martin Escardo on Continuity in Constructive Analysis
- Graham Leigh on Truth Theories
- Thomas Powell on Proof Mining
- Michael Rathjen on Constructive Set Theory and Type Theory
- Daniel Wessel on Constructive Algebra


AKTIV SEIN
GENIESSEN
ERLEBEN
TEGERNSEE SCHLIERSEE
Fischbachau
Herzlich Willkommen in Fischbachau
Genießen Sie einen Urlaub im Holler- und Kräutertal rund um Fischbachau

Agda

Agda is a dependently typed functional programming language. It has inductive families, i.e., data types which depend on values, such as the type of vectors of a given length. It also has parametrised modules, mixfix operators, Unicode characters, and an interactive Emacs interface which can assist the programmer in writing the program.

Agda is a proof assistant. It is an interactive system for writing and checking proofs. Agda is based on intuitionistic type theory, a foundational system for constructive mathematics developed by the Swedish logician Per Martin-Löf. It has many similarities with other proof assistants based on dependent types, such as Coq, Epigram, Matita and NuPRL.

Agda is open-source and enjoys contributions from many authors. The center of the Agda development is the Programming Logic group at Chalmers and Gothenburg University. The main developers are Ulf Norell, Nils Anders Danielsson, and Andreas Abel.