Proofs are Programs Summary

- **Write purely functional programs in Coq**
  - natural numbers, lists, maps, trees, program syntax
- **Verify these programs by proving theorems about them**
  - case analysis, induction, inversion, tactics, ...
- **Curry-Howard correspondence**
  - proofs = typed purely functional programs
- **Simple imperative programming language**
  - syntax and operational semantics

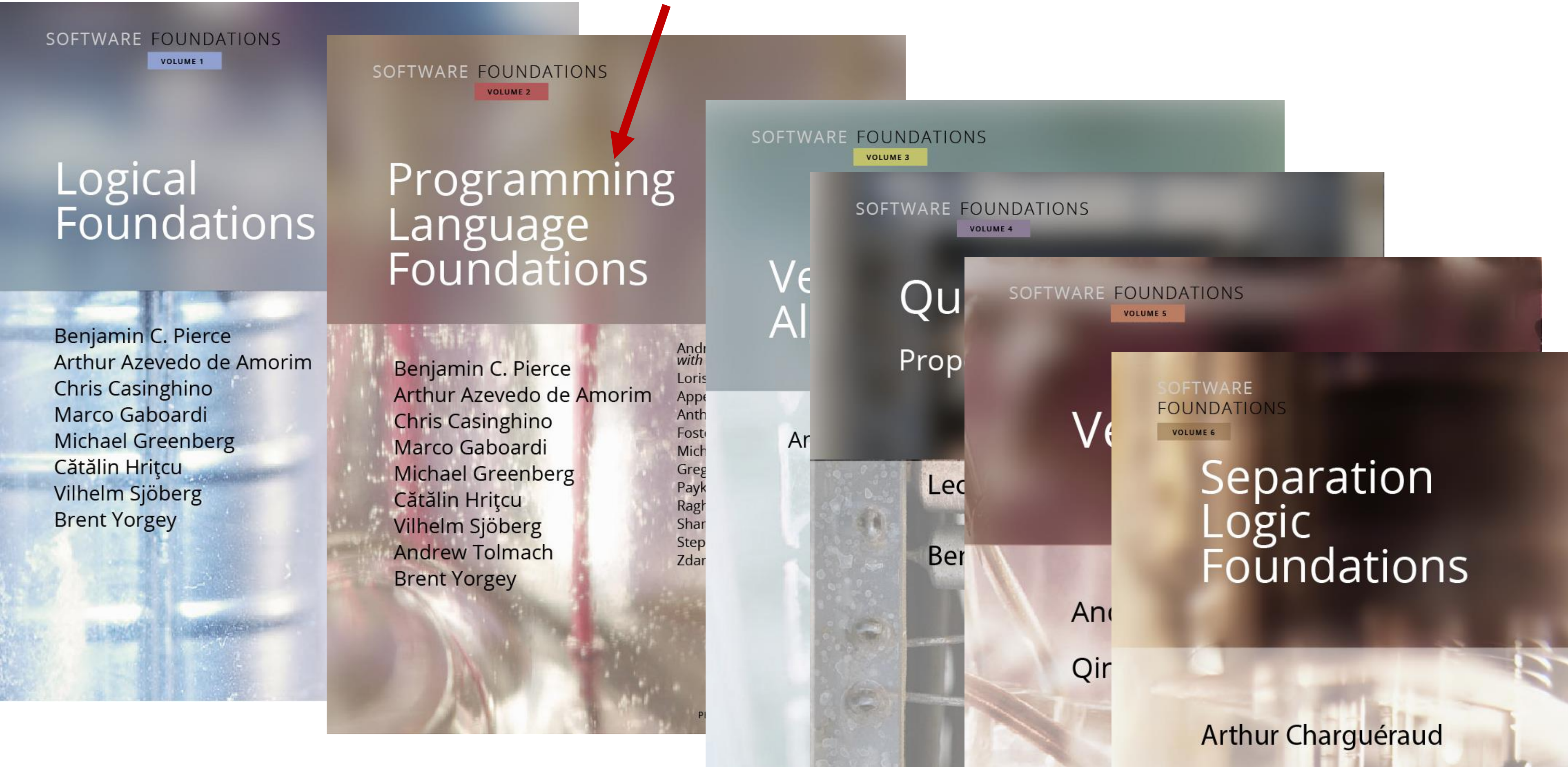
["Le coq mécanisé"
picture by Lilia Anisimova]

Course evaluation starts today

- **Please participate**
  - your feedback is very important to us
  - help us make this better!
- **2 evaluation forms**
  - one standard form from RUB
  - one specific form from us
- **More details via email/Moodle today**

Follow-up course next semester

Lecturers: Cătălin Hrițcu and Rob Blanco



SOFTWARE FOUNDATIONS
VOLUME 1
Logical Foundations
Benjamin C. Pierce
Arthur Azevedo de Amorim
Chris Casinghino
Marco Gaboardi
Michael Greenberg
Cătălin Hriţcu
Vilhelm Sjöberg
Brent Yorgey

SOFTWARE FOUNDATIONS
VOLUME 2
Programming Language Foundations
Benjamin C. Pierce
Arthur Azevedo de Amorim
Chris Casinghino
Marco Gaboardi
Michael Greenberg
Cătălin Hriţcu
Vilhelm Sjöberg
Andrew Tolmach
Brent Yorgey
Andr
with
Loris
Appe
Anth
Fost
Mich
Greg
Payk
Ragh
Shar
Step
Zdar

SOFTWARE FOUNDATIONS
VOLUME 3
Ve
Al
Ar

SOFTWARE FOUNDATIONS
VOLUME 4
Qu
Prop
Lec
Ber

SOFTWARE FOUNDATIONS
VOLUME 5
Ve
An
Qir

SOFTWARE FOUNDATIONS
VOLUME 6
Separation Logic Foundations
Arthur Charguéraud

Foundations of Everything

λ

- **Programming Languages**
  - Imp and Simply Typed Lambda-Calculus (functional)
  - type systems, program equivalence, semantics, metatheory
- **Verification**
  - Hoare Logic: verify Imp programs
  - Relational Hoare Logic
- **Security**
  - Information Flow Control: enforcing noninterference
    - Static enforcement: types, RHL, cryptographic constant time
    - Dynamic enforcement: Secure Multi-Execution, ...

reflect
+
Maps.v