Mixed Inductive-Coinductive Reasoning
Types, Programs and Logic

Henning Basold

PhD Defence
19 April 2018
Radboud University Nijmegen
Motivation

Well-behaved Programs and Proof Methods
Motivation

Well-behaved Programs and Proof Methods

Crashed Control System
Motivation

Well-behaved Programs and Proof Methods

Crashed Control System

Non-responsive Control System
What are Induction and Coinduction?

- **Coinduction**: Observable systems
  - Alive control systems

- **Induction**: Terminating computations
  - Internal computations finish

- **Induction-Coinduction**: Interleaved control and computations
  - Alive and responsive control systems
## What are Induction and Coinduction?

<table>
<thead>
<tr>
<th>Coinduction: Observable systems</th>
</tr>
</thead>
<tbody>
<tr>
<td>Alive control systems</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Induction: Terminating computations</th>
</tr>
</thead>
<tbody>
<tr>
<td>Internal computations finish</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Induction-Coinduction: Interleaved control and computations</th>
</tr>
</thead>
<tbody>
<tr>
<td>Alive and responsive control systems</td>
</tr>
</tbody>
</table>
Aims of the Thesis

Develop **formal languages** for inductive-coinductive programming and reasoning . . .

that enable **automatic verification** of proofs, . . .

have **formal semantics**, and . . .

are easy to understand **for humans**.
Aims of the Thesis

Develop **formal languages** for inductive-coinductive programming and reasoning . . .

that enable **automatic verification** of proofs, . . .

have **formal semantics**, and . . .

are easy to understand for humans.
Aims of the Thesis

Develop formal languages for inductive-coinductive programming and reasoning . . .

that enable automatic verification of proofs, . . .

have formal semantics, and . . .

are easy to understand for humans.
Aims of the Thesis

Develop **formal languages** for inductive-coinductive programming and reasoning . . .

that enable **automatic verification** of proofs, . . .

have **formal semantics**, and . . .

are easy to understand **for humans**.
Approach in the Thesis

Category Theory
Abstraction of Mathematical Theories

Type Theory
Typed Computations and Constructive Logic

Coalgebraic Methods
Theory of Systems
Approach in the Thesis

Category Theory
Abstraction of Mathematical Theories

Type Theory
Typed Computations and Constructive Logic

Coalgebraic Methods
Theory of Systems
Approach in the Thesis

**Category Theory**
Abstraction of Mathematical Theories

**Type Theory**
Typed Computations and Constructive Logic

**Coalgebraic Methods**
Theory of Systems
Approach in the Thesis

**Category Theory**
Abstraction of Mathematical Theories

**Type Theory**
Typed Computations and Constructive Logic

**Coalgebraic Methods**
Theory of Systems
Results of the Thesis

Well-behaved inductive-coinductive programs

Equivalence of inductive-coinductive programs
- Coalgebraic description of the equivalence
- Coalgebraic up-to techniques to simplify proofs
- Recursive logic to enable computer-verifiable proofs

Constructive logic based on induction and coinduction
- Inspired by category theoretical principles
- Justified computationally as type theory
Results of the Thesis

Well-behaved inductive-coinductive programs

- Coalgebraic description of the equivalence
- Coalgebraic up-to techniques to simplify proofs
- Recursive logic to enable computer-verifiable proofs

Constructive logic based on induction and coinduction

- Inspired by category theoretical principles
- Justified computationally as type theory
Results of the Thesis

Well-behaved inductive-coinductive programs

Equivalence of inductive-coinductive programs
- Coalgebraic description of the equivalence
- Coalgebraic up-to techniques to simplify proofs
- Recursive logic to enable computer-verifiable proofs

Constructive logic based on induction and coinduction
- Inspired by category theoretical principles
- Justified computationally as type theory
Thank you and back to the committee.