Inductive-Coinductive Reasoning

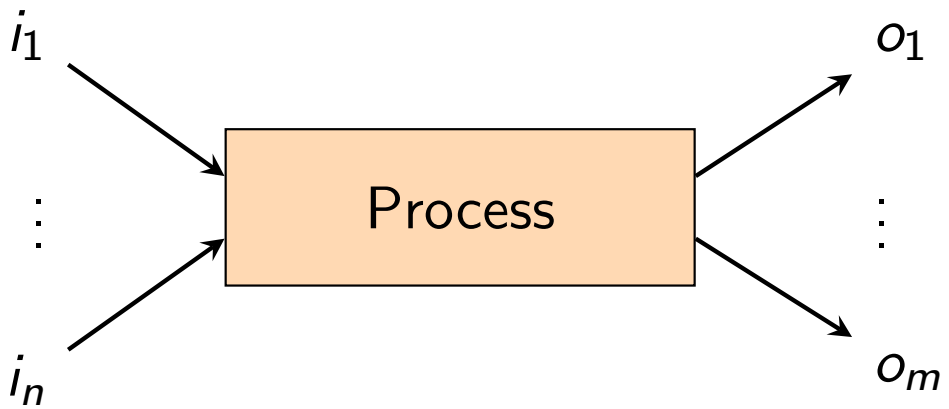
With an Eye Towards Reactive Systems

Henning Basold
Radboud University, Nijmegen and CWI, Amsterdam

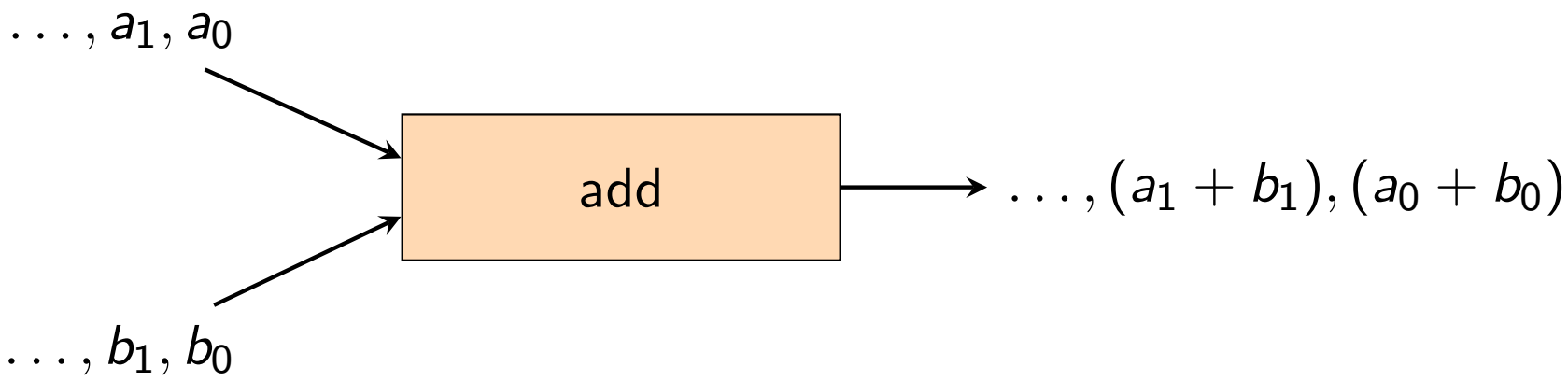
IPA Fall Days, Hoog Soeren
7 November 2016

Reactive Systems

- Reactive system: process that reacts with outputs to events from environment



Example: Addition



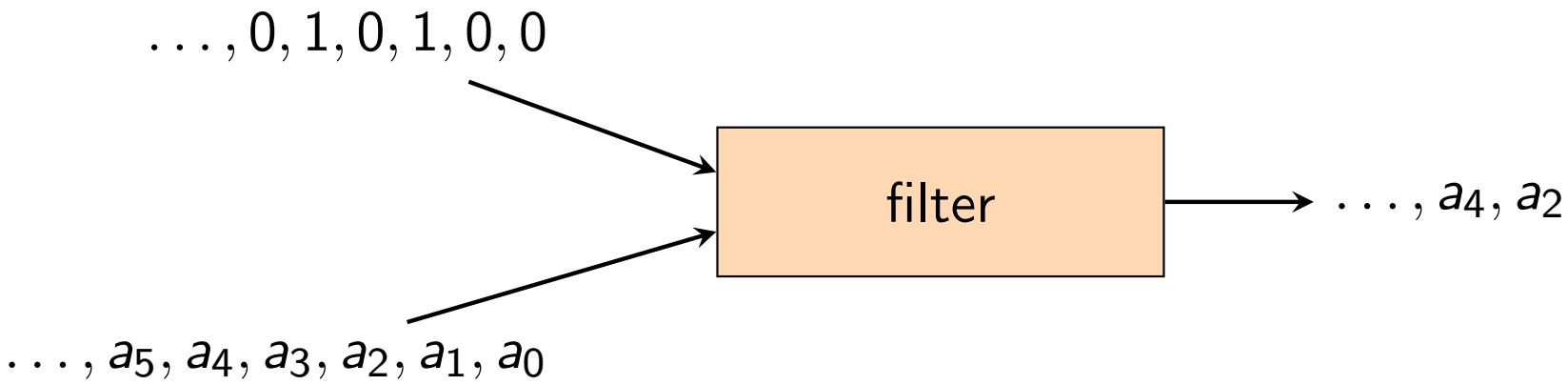
Programming Reactive Systems

- Lustre/Scade
  - Chaining of processes
  - Conservative: Iteration depth must be statically known
  - Thus: Many properties guaranteed by construction
- Functional Reactive Programming (e.g., in Haskell)
  - Process chaining again primitive construction
  - Allows the use of functional programming
  - Includes recursion: anything can happen

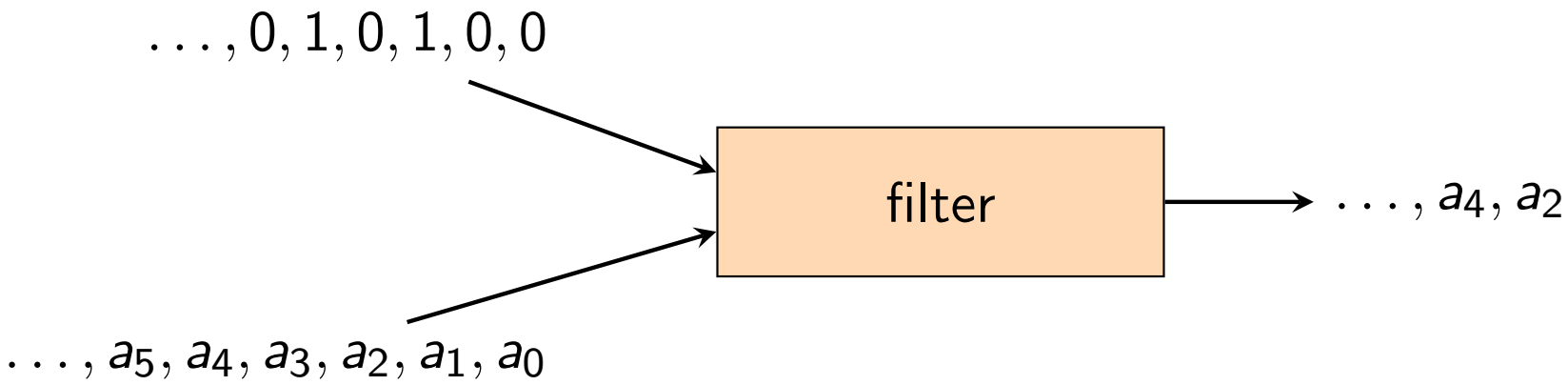
Here

We want iteration, but still guarantee some (liveness) properties

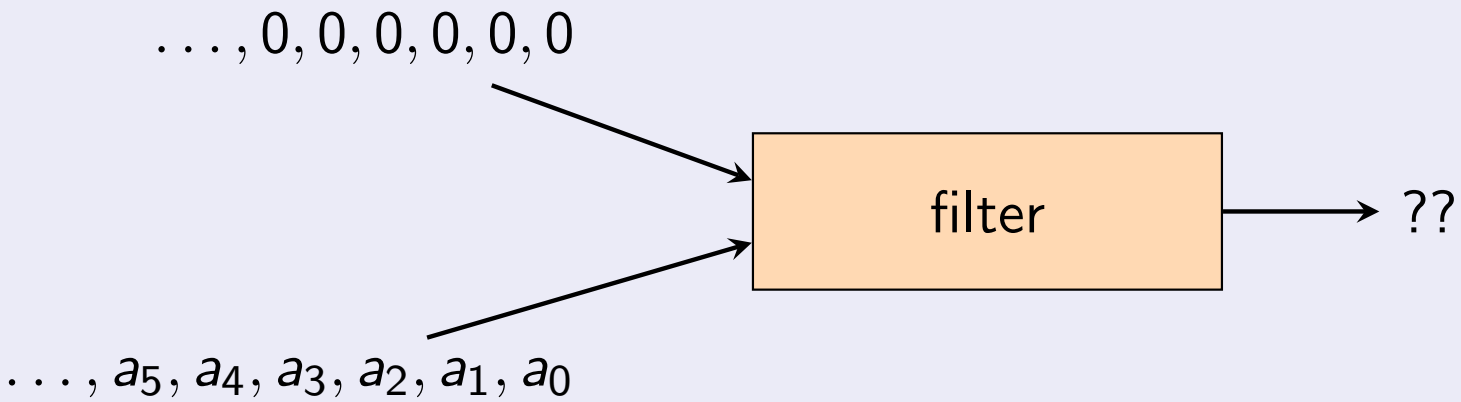
Example: Dynamic Filter



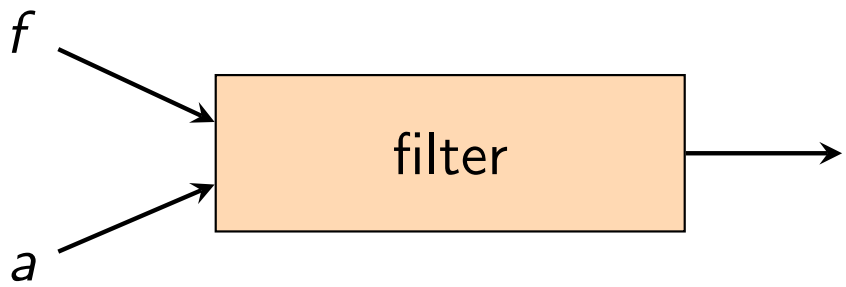
Example: Dynamic Filter



Problem



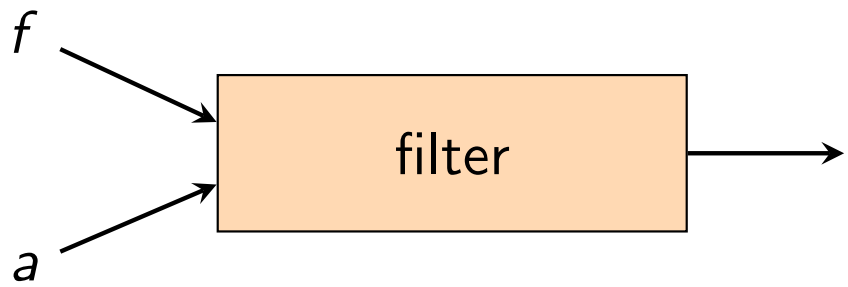
Productive Processes



Solution: Restrict input

- Requirement: f contains infinitely many 1s
- Equivalent: Every consecutive sequence of 0s in f is finite

Productive Processes



Solution: Restrict input

- Requirement: f contains infinitely many 1s
- Equivalent: Every consecutive sequence of 0s in f is finite

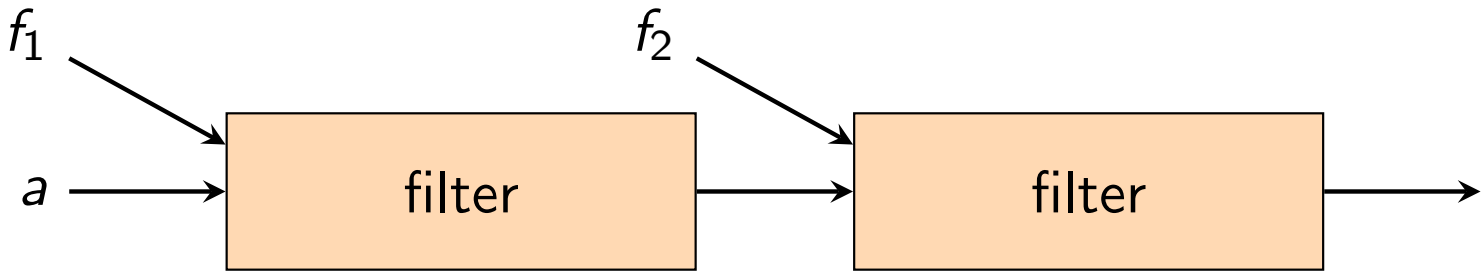
Enter Inductive-Coinductive Types

$$F := \nu X.\mu Y.\, X + Y$$

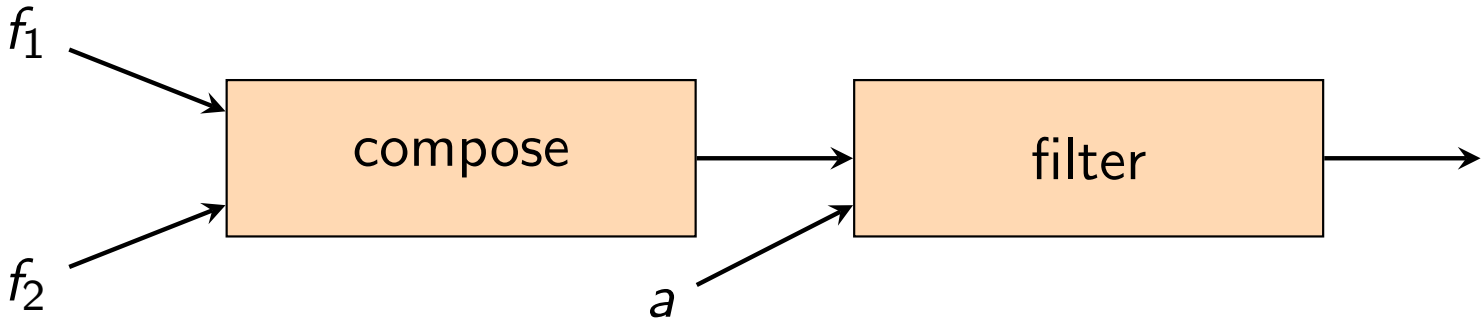
infinitely often "1" interleaved by finitely many "0"

Reasoning about Filter

- Define composition of filter inputs: $\mathrm{compose}\colon F \to F \to F$
- Prove, for example, that



and



have the same behaviour.

Perspectives

Develop

- Programming languages with inductive-coinductive types
- Proof techniques and logics for inductive-coinductive reasoning

Perspectives

Develop

- Programming languages with inductive-coinductive types
- Proof techniques and logics for inductive-coinductive reasoning

A Possible Logic

$$\varphi, \psi ::= \top \mid \bot \mid t \sim_A s \mid \blacktriangleright \varphi \mid \varphi \vee \psi \mid \varphi \wedge \psi \mid \varphi \to \psi$$
$$\mid \forall x : A.\, \varphi \mid \exists x : A.\, \varphi, \mid \mu(\varphi(\vec{x})).\, \psi \mid \nu(\varphi(\vec{x})).\, \psi$$

Perspectives

Develop

- Programming languages with inductive-coinductive types
- Proof techniques and logics for inductive-coinductive reasoning

A Possible Logic

$$\varphi, \psi ::= \top \mid \bot \mid t \sim_A s \mid \blacktriangleright \varphi \mid \varphi \vee \psi \mid \varphi \wedge \psi \mid \varphi \to \psi$$
$$\mid \forall x : A.\, \varphi \mid \exists x : A.\, \varphi, \mid \mu(\varphi(\vec{x})).\, \psi \mid \nu(\varphi(\vec{x})).\, \psi$$

Further Examples

- Weak bisimilarity: compare processes, while ignoring internal computation steps
- Substream relation: all elements of a sequence appear in order in another sequence

Thank you very much for your attention!