Nondeterminism as first class citizen for Hidden Logic

Daniel Gebler and Jörg Endrullis

VU University Amsterdam

CMCS 2012, Tallinn

Table of contents

Hidden Logic - Overview

Objective

- Semantics to OO software engineering
- Verification & Refinement of Design, not Code
- Behavioral abstraction
- Proof automation (Circular Coinduction)
- Tool support (CIRC)

Related Approaches

- Context induction [Hennicker, 1990]
- Observational Logic [Bidoit, Hennicker, Kurz, 2002]
- Observational proofs by rewriting [Bouhoula and Rusinowitch, 2002]
- Coherent Hidden Algebra [Diaconescu and Futatsugi, 2000]

Hidden Logic - Specifications and Semantics

Hidden specifications

A *hidden specification* is a tuple (Σ, Γ, E), where

- Σ a many-sorted signature with *hidden* and *visible* sorts,
- Γ a many-sorted subsignature of Σ,
- E is a set of equations.

Behavioral semantics

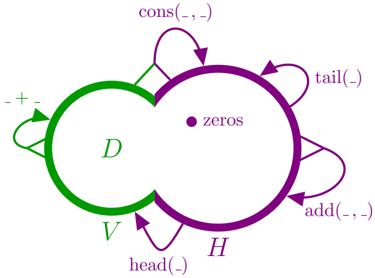
- *Experiments* are Γ-terms of visible sort with one "place-holder"
- *Behavioral equivalence* is non-distinguishability under experiments

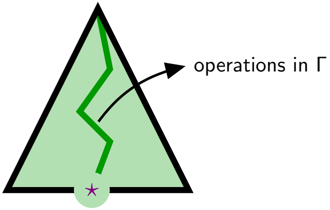
Coalgebraic nature

- $G_\Gamma : Set^H \to Set^H$
- $G_\Gamma(X)_h = \prod_{\gamma \in \Gamma_{hw,s}} X_s^{D_w}$
- $\mathrm{HAlg}(\Gamma) \simeq G_\Gamma - \mathrm{Coalg}$

Hidden Logic - Example

Sorts	Visible sort: $\mathbb{N}$, Hidden sort: Stream
Operations	$\mathrm{head}: \mathrm{Stream} \to \mathbb{N}$
	$\mathrm{tail}: \mathrm{Stream} \to \mathrm{Stream}$
	$\mathrm{add}: \mathrm{Stream} \times \mathrm{Stream} \to \mathrm{Stream}$
Equations	$\mathrm{head}(\mathrm{add}(s, s')) = \mathrm{head}(s) + \mathrm{head}(s')$
	$\mathrm{tail}(\mathrm{add}(s, s')) = \mathrm{add}(\mathrm{tail}(s), \mathrm{tail}(s'))$
Experiments	$\mathrm{head}(\bullet)$, $\mathrm{head}(\mathrm{tail}^n(\bullet))$

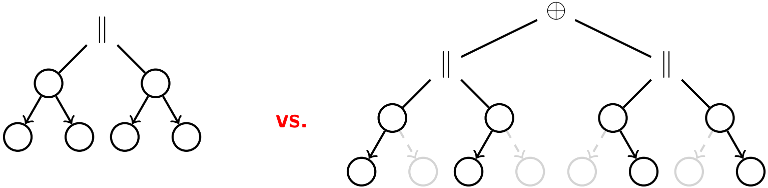




Problem Motivation (intuitive)

1) Underspecification vs. Inherent nondeterminism

2) Sharing choices between nondeterministic systems



Leading example

Specification

$$\mathrm{rand} : \to \mathit{Stream} \qquad \mathrm{dup} : \mathit{Stream} \to \mathit{Stream}$$

$$\mathrm{rand} = (0 \oplus 1) : \mathrm{rand}$$

$$\mathrm{dup}(\sigma) = \mathrm{hd}(\sigma) : \mathrm{hd}(\sigma) : \mathrm{dup}(\mathrm{tl}(\sigma))$$

Example 1: Underspecification vs. Inherent nondeterminism

$$\mathit{add}(\mathit{rand}, \mathit{rand}) \stackrel{?}{=}$$

Example 2: Sharing choices between nondeterministic systems

$$\mathit{dup}(\mathit{rand}) \stackrel{?}{=} \mathit{hd}(\mathit{rand}) : \mathit{hd}(\mathit{rand}) : \mathit{dup}(\mathit{tl}(\mathit{rand}))$$

Behavioral Specification

Nondeterministic Hidden specification

A *nondeterminsitic hidden specification* is a tuple $(\Sigma_{fun}, \Sigma_{rel}, \Gamma, E)$

- Σ_{fun} a many-sorted signature of deterministic functions
- Σ_{rel} a many-sorted signature of nondeterministic functions
- $\Sigma = \Sigma_{fun} \cup \Sigma_{rel} \cup \{\oplus_s \mid s \in \mathcal{S}\}$
- E a set of equations
  - $\ell \doteq r$ (behavioral deterministic)
  - $\ell = r$ (behavioral nondeterministic)

Algebraic and Behavioral Semantics

Nondeterministic Hidden Algebra

A *nondeterminsitic hidden algebra* is a Σ-*multialgebra* $\langle A, [\![\cdot]\!]\rangle$ with interpretation

- $[\![f]\!] : A_{s_1} \times \ldots \times A_{s_n} \to \mathcal{P}^+(A_s)$ for $f \in \Sigma_{s_1 \ldots s_n, s}$
- $[\![f]\!](a_1, \ldots, a_n)$ singleton for $f \in \Sigma_{fun}$
- Extension to $[\![f]\!] : \mathcal{P}^+(A_{s_1}) \times \ldots \times \mathcal{P}^+(A_{s_n}) \to \mathcal{P}^+(A_s)$ via union
- Assignment: $\alpha : \mathcal{X} \to A$
- Natural lifting to terms $[\![\cdot]\!]_\alpha : Ter(\Sigma, \mathcal{X}) \to \mathcal{P}^+(A)$
- $[\![s \oplus t]\!]_\alpha = [\![s]\!]_\alpha \cup [\![t]\!]_\alpha$

Behavioral equivalence

$$a \equiv b \quad \text{iff} \quad [\![C[* : s]]\!]_{* \mapsto a} = [\![C[* : s]]\!]_{* \mapsto b}$$

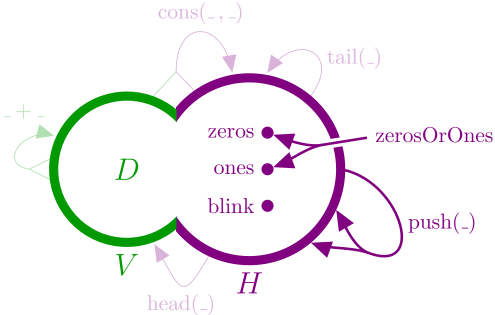
$$\text{for every } C \in Ter(\Sigma, \{*\})_v$$

Leading example (II)

Specification

$$\mathsf{zerosOrOnes} = \mathsf{zeros} \oplus \mathsf{ones} \qquad \mathsf{zeros} = 0 : \mathsf{zeros}$$

$$\mathsf{push}(\sigma) = (0 \oplus 1) : \sigma \qquad \mathsf{ones} = 1 : \mathsf{ones}$$



Representation of nondeterministic operations:

$$[\![f]\!] : \mathcal{P}^+(A_{s_1}) \times \ldots \times \mathcal{P}^+(A_{s_n}) \to \mathcal{P}^+(A_s)$$

with requirement: $[\![f]\!](A_1, \ldots, A_n) = \bigcup_{a_1 \in A_1, \ldots, a_n \in A_n} f(\{a_1\}, \ldots, \{a_n\})$

Sharing of terms

$$\mathsf{rand} = (0 \oplus 1) : \mathsf{rand} \qquad \mathsf{zeros} = 0 : \mathsf{zeros}$$

$$\mathsf{add}(x : \sigma, y : \tau) = (x + y) : \mathsf{add}(\sigma, \tau) \qquad \mathsf{fun}(\sigma) = \mathsf{add}(\sigma, \sigma)$$

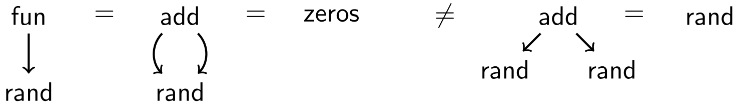
Adding two independent random streams gives a random stream:

$$\mathsf{add}(\mathsf{rand}, \mathsf{rand}) = \mathsf{rand}$$

But we have

$$\mathsf{fun}(\mathsf{rand}) = \mathsf{zeros} \neq \mathsf{add}(\mathsf{rand}, \mathsf{rand})$$

Idea: *sharing* to express that both rand's refer to the same random choice:

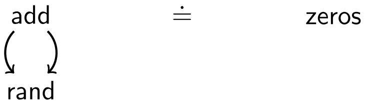


We introduce sharing during equational reasoning if variable is duplicated.

Behavioral Reasoning

$$\mathsf{rand} = (0 \oplus 1) : \mathsf{rand}$$
$$\mathsf{add}(x : \sigma, y : \tau) \doteq (x + y) : \mathsf{add}(\sigma, \tau)$$
$$\mathsf{zeros} \doteq 0 : \mathsf{zeros}$$

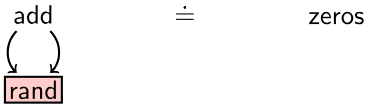
with $\{:, \mathsf{add}, \mathsf{zeros}, +\} \subseteq \Sigma_{fun}$ and $\{\mathsf{rand}\} \subseteq \Sigma_{rel}$.



Behavioral Reasoning

$$\mathsf{rand} = (0 \oplus 1) : \mathsf{rand}$$
$$\mathsf{add}(x : \sigma, y : \tau) \doteq (x + y) : \mathsf{add}(\sigma, \tau)$$
$$\mathsf{zeros} \doteq 0 : \mathsf{zeros}$$

with $\{:, \mathsf{add}, \mathsf{zeros}, +\} \subseteq \Sigma_{fun}$ and $\{\mathsf{rand}\} \subseteq \Sigma_{rel}$.



equational reasoning

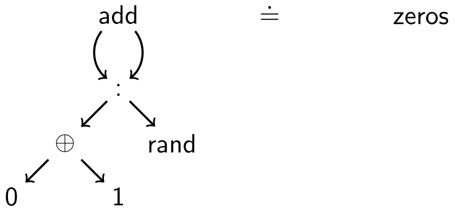
Behavioral Reasoning

$$\mathsf{rand} = (0 \oplus 1) : \mathsf{rand}$$

$$\mathsf{add}(x : \sigma, y : \tau) \doteq (x + y) : \mathsf{add}(\sigma, \tau)$$

$$\mathsf{zeros} \doteq 0 : \mathsf{zeros}$$

with $\{:, \mathsf{add}, \mathsf{zeros}, +\} \subseteq \Sigma_{fun}$ and $\{\mathsf{rand}\} \subseteq \Sigma_{rel}$.



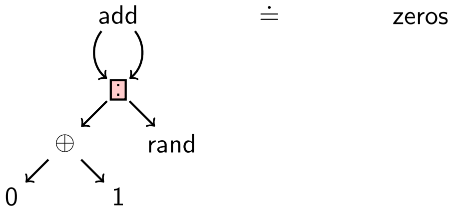
Behavioral Reasoning

$$\mathsf{rand} = (0 \oplus 1) : \mathsf{rand}$$

$$\mathsf{add}(x : \sigma, y : \tau) \doteq (x + y) : \mathsf{add}(\sigma, \tau)$$

$$\mathsf{zeros} \doteq 0 : \mathsf{zeros}$$

with $\{ : , \mathsf{add}, \mathsf{zeros}, + \} \subseteq \Sigma_{fun}$ and $\{\mathsf{rand}\} \subseteq \Sigma_{rel}$.

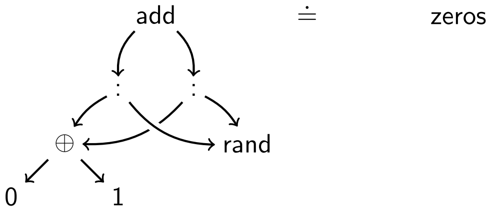


unsharing of Σ_{fun} symbol
(deterministic symbols can always be unshared)
(hence usual reasoning if $\Sigma_{rel} = \emptyset$)

Behavioral Reasoning

$$\text{rand} = (0 \oplus 1) : \text{rand}$$
$$\text{add}(x : \sigma, y : \tau) \doteq (x + y) : \text{add}(\sigma, \tau)$$
$$\text{zeros} \doteq 0 : \text{zeros}$$

with $\{:, \text{add}, \text{zeros}, +\} \subseteq \Sigma_{fun}$ and $\{\text{rand}\} \subseteq \Sigma_{rel}$.



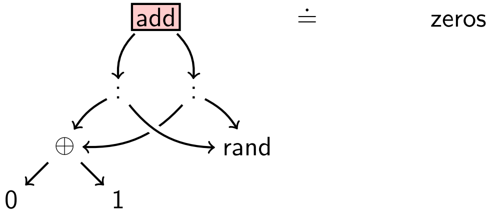
Behavioral Reasoning

$$\mathsf{rand} = (0 \oplus 1) : \mathsf{rand}$$

$$\mathsf{add}(x : \sigma, y : \tau) \doteq (x + y) : \mathsf{add}(\sigma, \tau)$$

$$\mathsf{zeros} \doteq 0 : \mathsf{zeros}$$

with $\{:, \mathsf{add}, \mathsf{zeros}, +\} \subseteq \Sigma_{fun}$ and $\{\mathsf{rand}\} \subseteq \Sigma_{rel}$.



equational reasoning

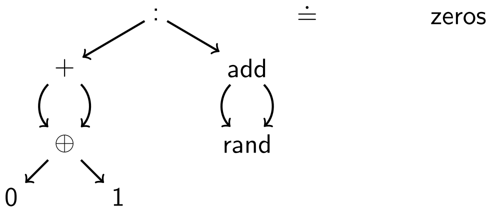
(unsharing was needed)

(no equational reasoning across symbols with multiple incoming edges)

Behavioral Reasoning

$$\mathsf{rand} = (0 \oplus 1) : \mathsf{rand}$$
$$\mathsf{add}(x : \sigma, y : \tau) \doteq (x + y) : \mathsf{add}(\sigma, \tau)$$
$$\mathsf{zeros} \doteq 0 : \mathsf{zeros}$$

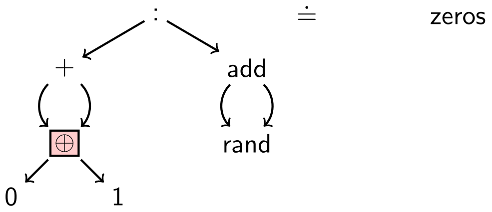
with $\{:, \mathsf{add}, \mathsf{zeros}, +\} \subseteq \Sigma_{fun}$ and $\{\mathsf{rand}\} \subseteq \Sigma_{rel}$.



Behavioral Reasoning

$$\mathrm{rand} = (0 \oplus 1) : \mathrm{rand}$$
$$\mathrm{add}(x : \sigma, y : \tau) \doteq (x + y) : \mathrm{add}(\sigma, \tau)$$
$$\mathrm{zeros} \doteq 0 : \mathrm{zeros}$$

with $\{:, \mathrm{add}, \mathrm{zeros}, +\} \subseteq \Sigma_{fun}$ and $\{\mathrm{rand}\} \subseteq \Sigma_{rel}$.

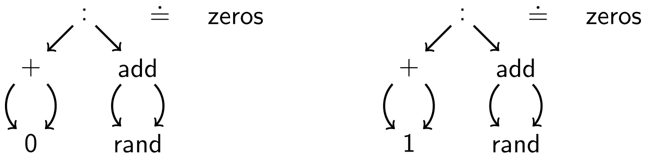


case distinction for $\oplus$

Behavioral Reasoning

$$\mathrm{rand} = (0 \oplus 1) : \mathrm{rand}$$
$$\mathrm{add}(x : \sigma, y : \tau) \doteq (x + y) : \mathrm{add}(\sigma, \tau)$$
$$\mathrm{zeros} \doteq 0 : \mathrm{zeros}$$

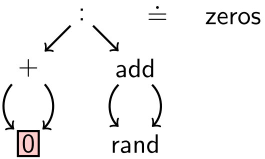
with $\{ : , \mathrm{add}, \mathrm{zeros}, + \} \subseteq \Sigma_{fun}$ and $\{\mathrm{rand}\} \subseteq \Sigma_{rel}$.

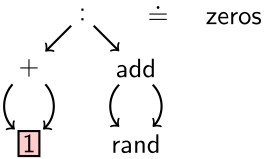


Behavioral Reasoning

$$\mathsf{rand} = (0 \oplus 1) : \mathsf{rand}$$
$$\mathsf{add}(x : \sigma, y : \tau) \doteq (x + y) : \mathsf{add}(\sigma, \tau)$$
$$\mathsf{zeros} \doteq 0 : \mathsf{zeros}$$

with $\{:, \mathsf{add}, \mathsf{zeros}, +\} \subseteq \Sigma_{fun}$ and $\{\mathsf{rand}\} \subseteq \Sigma_{rel}$.



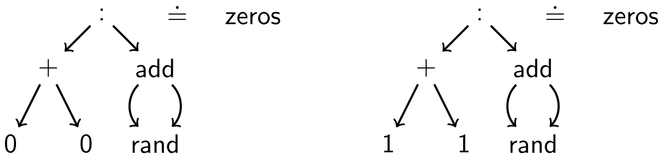


unsharing of Σ_{fun} symbol

Behavioral Reasoning

$$\mathrm{rand} = (0 \oplus 1) : \mathrm{rand}$$
$$\mathrm{add}(x : \sigma, y : \tau) \doteq (x + y) : \mathrm{add}(\sigma, \tau)$$
$$\mathrm{zeros} \doteq 0 : \mathrm{zeros}$$

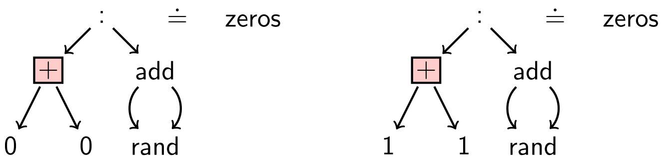
with $\{:, \mathrm{add}, \mathrm{zeros}, +\} \subseteq \Sigma_{fun}$ and $\{\mathrm{rand}\} \subseteq \Sigma_{rel}$.



Behavioral Reasoning

$$\mathsf{rand} = (0 \oplus 1) : \mathsf{rand}$$
$$\mathsf{add}(x : \sigma, y : \tau) \doteq (x + y) : \mathsf{add}(\sigma, \tau)$$
$$\mathsf{zeros} \doteq 0 : \mathsf{zeros}$$

with $\{\, : , \mathsf{add}, \mathsf{zeros}, + \} \subseteq \Sigma_{fun}$ and $\{\mathsf{rand}\} \subseteq \Sigma_{rel}$.

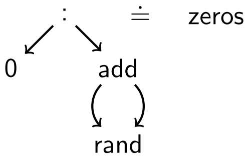


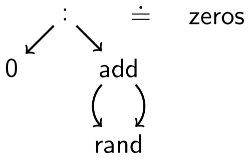
equational reasoning

Behavioral Reasoning

$$\mathsf{rand} = (0 \oplus 1) : \mathsf{rand}$$
$$\mathsf{add}(x : \sigma, y : \tau) \doteq (x + y) : \mathsf{add}(\sigma, \tau)$$
$$\mathsf{zeros} \doteq 0 : \mathsf{zeros}$$

with $\{\, : , \mathsf{add}, \mathsf{zeros}, +\} \subseteq \Sigma_{fun}$ and $\{\mathsf{rand}\} \subseteq \Sigma_{rel}$.





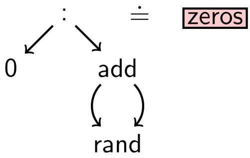
Behavioral Reasoning

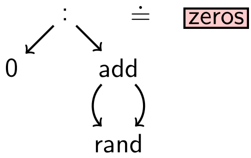
$$\mathsf{rand} = (0 \oplus 1) : \mathsf{rand}$$

$$\mathsf{add}(x : \sigma, y : \tau) \doteq (x + y) : \mathsf{add}(\sigma, \tau)$$

$$\mathsf{zeros} \doteq 0 : \mathsf{zeros}$$

with $\{ : , \mathsf{add}, \mathsf{zeros}, +\} \subseteq \Sigma_{fun}$ and $\{\mathsf{rand}\} \subseteq \Sigma_{rel}$.



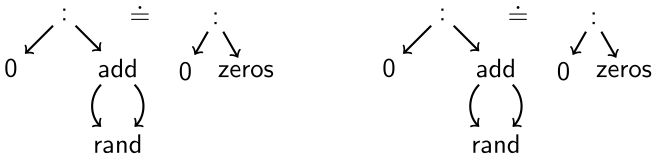


equational reasoning

Behavioral Reasoning

$$\mathsf{rand} = (0 \oplus 1) : \mathsf{rand}$$
$$\mathsf{add}(x : \sigma, y : \tau) \doteq (x + y) : \mathsf{add}(\sigma, \tau)$$
$$\mathsf{zeros} \doteq 0 : \mathsf{zeros}$$

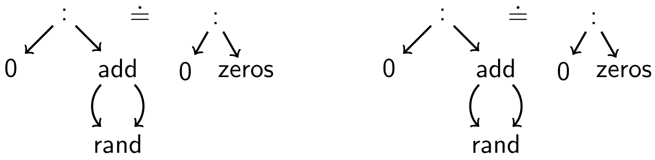
with $\{:, \mathsf{add}, \mathsf{zeros}, +\} \subseteq \Sigma_{fun}$ and $\{\mathsf{rand}\} \subseteq \Sigma_{rel}$.



Behavioral Reasoning

$$\mathsf{rand} = (0 \oplus 1) : \mathsf{rand}$$
$$\mathsf{add}(x : \sigma, y : \tau) \doteq (x + y) : \mathsf{add}(\sigma, \tau)$$
$$\mathsf{zeros} \doteq 0 : \mathsf{zeros}$$

with $\{:, \mathsf{add}, \mathsf{zeros}, +\} \subseteq \Sigma_{fun}$ and $\{\mathsf{rand}\} \subseteq \Sigma_{rel}$.

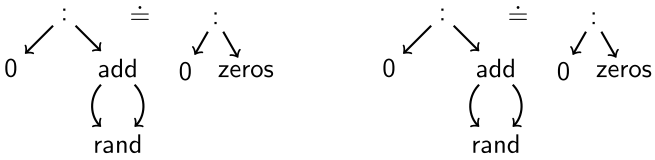


circular coinduction: heads are equal
tails are exactly the equation we started from

Behavioral Reasoning

$$\mathsf{rand} = (0 \oplus 1) : \mathsf{rand}$$
$$\mathsf{add}(x : \sigma, y : \tau) \doteq (x + y) : \mathsf{add}(\sigma, \tau)$$
$$\mathsf{zeros} \doteq 0 : \mathsf{zeros}$$

with $\{:, \mathsf{add}, \mathsf{zeros}, +\} \subseteq \Sigma_{fun}$ and $\{\mathsf{rand}\} \subseteq \Sigma_{rel}$.



circular coinduction: heads are equal
tails are exactly the equation we started from

qed

Equational Reasoning and Sharing

No equational reasoning across symbols with multiple incoming edges

$$\mathsf{push}(\sigma) = (0 \oplus 1) : \sigma \qquad \mathsf{zeros} = 0 : \mathsf{zeros}$$

$$\mathsf{add}(x : \sigma, y : \tau) = (x + y) : \mathsf{add}(\sigma, \tau)$$

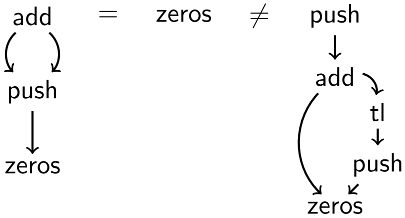
$$\mathsf{add}(\mathsf{push}(\sigma), \tau) = \mathsf{push}(\mathsf{add}(\sigma, \mathsf{tl}(\tau))) \qquad \Sigma_{rel} = \{\mathsf{push}\}$$

Last equation:

if the first bit of the first argument is random,

then first bit of outcome is random

However, this holds only since the arguments are not shared!



Conclusion

Summary

- Nondeterminism as first class citizen
- Pointwise lifting of deterministic behavior
- Sharing allows to replicate choices in nondeterministic systems
- Nondeterministic and sharing extensions are conservative ($\Sigma_{rel} = \emptyset$)

Future work

- Coalgebraic interpretation
- Formalize Circular Coinduction proof rules for sharing
- Interplay circular induction and circular coinduction with sharing
- Implementation CIRC
- Samples (QoS/Security of P2P networks)