On a Monadic Encoding of Continuous Behaviour

Renato Neves

joint work with: Luís Barbosa, Manuel Martins, Dirk Hofmann

INESC TEC (HASLab) & Universidade do Minho

October 1, 2015

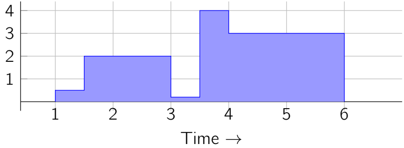
The main goal

A coalgebraic calculus of hybrid components.

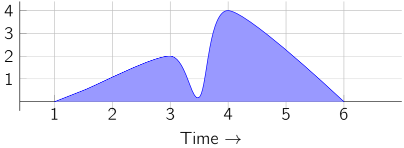
Motivation & Context

Hybrid systems possess both discrete and continuous behaviour.

A discrete evolution



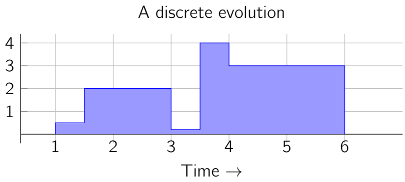
A continuous evolution

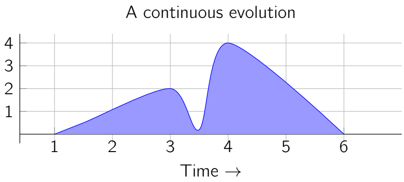


- They are often complex
- but can be seen as the composition of (simpler) components.

Motivation & Context

Hybrid systems possess both discrete and continuous behaviour.

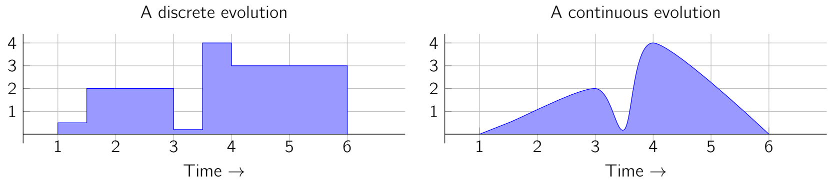




- They are often complex
- but can be seen as the composition of (simpler) components.

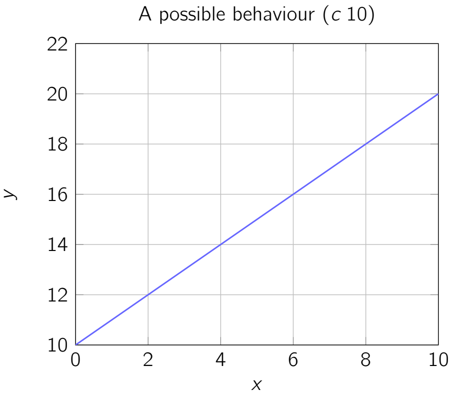
Motivation & Context

Hybrid systems possess both discrete and continuous behaviour.

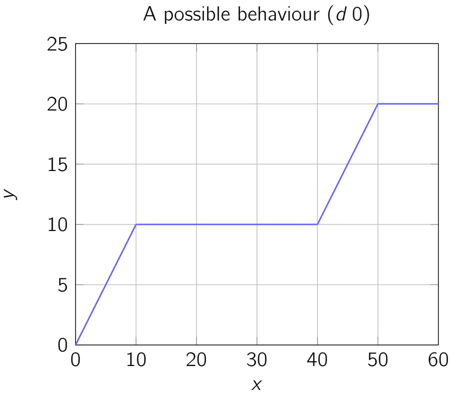


- They are often complex
- but can be seen as the composition of (simpler) components.

Thermostat



Water level regulator



Hybrid components (coalgebraically)

Arrows of type $S \times I \to S \times \mathcal{H}O$ where

- $S \times I \to S$ defines the internal (discrete) transitions
- and $S \times I \to \mathcal{H}O$ the observable (continuous) behaviour.

This favours a coalgebraic perspective !

Hybrid components (coalgebraically)

Arrows of type $S \times I \to S \times \mathcal{H}O$ where

- $S \times I \to S$ defines the internal (discrete) transitions
- and $S \times I \to \mathcal{H}O$ the observable (continuous) behaviour.

This favours a coalgebraic perspective !

Coalgebras & Hybrid systems (related work)

- Object–oriented hybrid systems of coalgebras plus monoid actions [Jacobs, 2000]. A coalgebra for the (discrete) assignments, a monoid for the (continuous) evolutions.

- Notions of bisimulation for hybrid systems (resort to open maps) [Haghverdi et al., 2005].

Coalgebras & Hybrid systems (related work)

- Object–oriented hybrid systems of coalgebras plus monoid actions [Jacobs, 2000]. A coalgebra for the (discrete) assignments, a monoid for the (continuous) evolutions.

- Notions of bisimulation for hybrid systems (resort to open maps) [Haghverdi et al., 2005].

Components as coalgebras

We view a component as

$$\langle\, s \in S, c : S \times I \to \mathcal{B}(S \times O)\,\rangle$$

where $\mathcal{B}$ is a (strong) monad that captures a specific type of behaviour [Barbosa, 2001].

[Barbosa, 2001] shows how to generate a rich component algebra from a strong monad.

Motivation & Context

Different monads capture different types of behaviour ...

... and thus different kinds of component

- Maybe monad ($\mathcal{M}$) $\rightsquigarrow$ faulty components
- Powerset monad ($\mathcal{P}$) $\rightsquigarrow$ non–deterministic components
- Distribution monad ($\mathcal{D}$) $\rightsquigarrow$ probabilistic components
- Hybrid monad ($\mathcal{H}$) $\rightsquigarrow$ hybrid components

Motivation & Context

Different monads capture different types of behaviour ...

... and thus different kinds of component

- Maybe monad ($\mathcal{M}$) $\rightsquigarrow$ faulty components
- Powerset monad ($\mathcal{P}$) $\rightsquigarrow$ non–deterministic components
- Distribution monad ($\mathcal{D}$) $\rightsquigarrow$ probabilistic components
- Hybrid monad ($\mathcal{H}$) $\rightsquigarrow$ hybrid components

Motivation & Context

Different monads capture different types of behaviour . . .

. . . and thus different kinds of component

- Maybe monad ($\mathcal{M}$) $\rightsquigarrow$ faulty components
- Powerset monad ($\mathcal{P}$) $\rightsquigarrow$ non–deterministic components
- Distribution monad ($\mathcal{D}$) $\rightsquigarrow$ probabilistic components
- Hybrid monad ($\mathcal{H}$) $\rightsquigarrow$ hybrid components

Motivation & Context

Different monads capture different types of behaviour . . .

. . . and thus different kinds of component

- Maybe monad ($\mathcal{M}$) $\rightsquigarrow$ faulty components
- Powerset monad ($\mathcal{P}$) $\rightsquigarrow$ non–deterministic components
- Distribution monad ($\mathcal{D}$) $\rightsquigarrow$ probabilistic components
- Hybrid monad ($\mathcal{H}$) $\rightsquigarrow$ hybrid components

Motivation & Context

Different monads capture different types of behaviour ...

... and thus different kinds of component

- Maybe monad ($\mathcal{M}$) $\rightsquigarrow$ faulty components
- Powerset monad ($\mathcal{P}$) $\rightsquigarrow$ non–deterministic components
- Distribution monad ($\mathcal{D}$) $\rightsquigarrow$ probabilistic components
- Hybrid monad ($\mathcal{H}$) $\rightsquigarrow$ hybrid components

Monad $\mathcal{H}$

It is defined in such a way that

$$\frac{c : S \times I \to S \times \mathcal{H}O}{c : S \times I \to S \times (O^{\mathsf{T}} \times D)} \text{ unfold } \mathcal{H}$$

where $\mathsf{T} = \mathbb{R}_{\geq 0}$ and $D = [0, \infty]$.

Kleisli composition allows the transfer of evolution control between components.

Technically, this amounts to concatenation of evolutions.

Monad $\mathcal{H}$

It is defined in such a way that

$$\frac{c : S \times I \to S \times \mathcal{H}O}{c : S \times I \to S \times (O^{\mathsf{T}} \times D)} \ \text{unfold } \mathcal{H}$$

where $\mathsf{T} = \mathbb{R}_{\geq 0}$ and $D = [0, \infty]$.

Kleisli composition allows the transfer of evolution control between components.

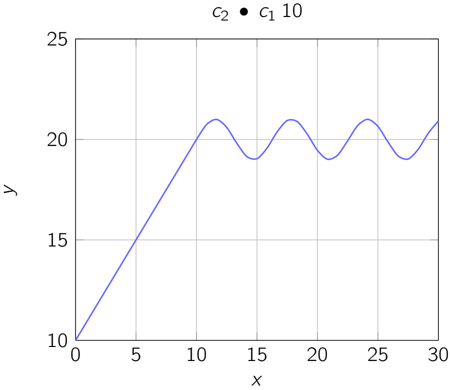
Technically, this amounts to concatenation of evolutions.

Kleisli composition (thermostat revisited)

$$c_1\ i = (\ \lambda t.(i + t),\ 10\ ),\quad c_2\ i = (\ \lambda t.(i + sin\ t),\ \infty\ )$$

Kleisli composition (thermostat revisited)

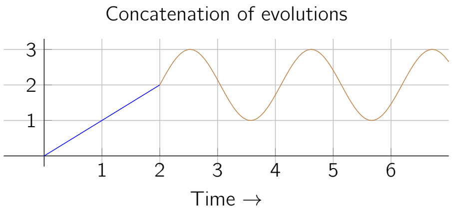
$$c_1\ i = (\ \lambda t.(i + t),\ 10\ ),\quad c_2\ i = (\ \lambda t.(i + sin\ t),\ \infty\ )$$



Monad $\mathcal{H}$ and Höfner's Algebra

Kleisli composition of Monad $\mathcal{H}$ corresponds to concatenation of evolutions in

An algebra of hybrid systems [Höfner, 2009]



Assembly of monad $\mathcal{H}$ (underlying functor)

Based upon the category of topological spaces **Top**.

Definition

Given a space $X \in |\mathbf{Top}|$,

$$\mathcal{H}X \mathrel{\widehat{=}} \{\, (f, d) \in X^{\mathsf{T}} \times D \mid f \cdot \lambda_d = f \,\}$$

where $\lambda_d = id \lhd \leq_d \rhd \underline{d}$.

Definition

Given a continuous function $g : X \to Y$,

$$\mathcal{H}g : \mathcal{H}X \to \mathcal{H}Y, \qquad \mathcal{H}g \mathrel{\widehat{=}} g^{\mathsf{T}} \times id$$

Intuitively, $\mathcal{H}g$ alters evolutions pointwise (but keeps durations).

Assembly of monad $\mathcal{H}$ (underlying functor)

Based upon the category of topological spaces **Top**.

Definition

Given a space $X \in |\mathbf{Top}|$,

$$\mathcal{H}X \;\widehat{=}\; \{\, (f, d) \in X^{\mathsf{T}} \times D \mid f \cdot \lambda_d = f \,\}$$

where $\lambda_d = id \lhd \leq_d \rhd \underline{d}$.

Definition

Given a continuous function $g : X \to Y$,

$$\mathcal{H}g : \mathcal{H}X \to \mathcal{H}Y, \qquad \mathcal{H}g \;\widehat{=}\; g^{\mathsf{T}} \times id$$

Intuitively, $\mathcal{H}g$ alters evolutions pointwise (but keeps durations).

Assembly of monad $\mathcal{H}$ (underlying functor)

Based upon the category of topological spaces **Top**.

Definition

Given a space $X \in |\mathbf{Top}|$,

$$\mathcal{H}X \;\widehat{=}\; \{\, (f, d) \in X^{\mathsf{T}} \times D \mid f \cdot \lambda_d = f \,\}$$

where $\lambda_d = id \lhd \;\leq_d\; \rhd \underline{d}$.

Definition

Given a continuous function $g : X \to Y$,

$$\mathcal{H}g : \mathcal{H}X \to \mathcal{H}Y, \qquad \mathcal{H}g \;\widehat{=}\; g^{\mathsf{T}} \times id$$

Intuitively, $\mathcal{H}g$ alters evolutions pointwise (but keeps durations).

Assembly of monad $\mathcal{H}$ (underlying functor)

An interesting algebra

$$\begin{tikzcd} \mathcal{H}X \arrow[d, "\theta"] \\ X \end{tikzcd} \qquad \theta\,(f, d) \;\widehat{=}\; f\;0$$

Assembly of Monad $\mathcal{H}$ (monad operations)

$$\begin{tikzcd} Id \arrow[dr, "\eta"'] & & \mathcal{H}\mathcal{H} \arrow[dl, "\mu"] \\ & \mathcal{H} & \end{tikzcd}$$

Definition

Given a space $X \in |\mathbf{Top}|$,

$$\eta_X \, x \mathrel{\widehat{=}} (\underline{x}, 0)$$

Defines the simplest continuous system of type $X \to \mathcal{H}X$.

$$\begin{tikzcd}
Id \arrow[dr, "\eta"'] & & \mathcal{H}\mathcal{H} \arrow[dl, "\mu"] \\
& \mathcal{H} &
\end{tikzcd}$$

Definition

Given a space $X \in |\mathbf{Top}|$,

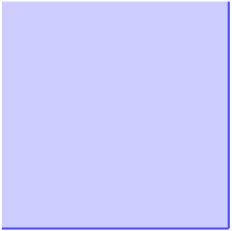
$$\mu_X\,(f, d) \mathrel{\widehat{=}} (\theta \cdot f, d) \mathbin{+\!\!+} (f\ d)$$

...

Assembly of Monad $\mathcal{H}$ (monad operations)

Let us reason

$$
\begin{aligned}
\mathcal{H}\mathcal{H}X &\subseteq \\
(\mathcal{H}X)^{\mathsf{T}} \times D &\to \\
(\mathcal{H}X)^{T} &\subseteq \\
(X^{\mathsf{T}} \times D)^{\mathsf{T}} &\cong \\
(X^{\mathsf{T}})^{\mathsf{T}} \times D^{\mathsf{T}} &\to \\
(X^{\mathsf{T}})^{\mathsf{T}} &\cong \\
X^{\mathsf{T}\times\mathsf{T}}
\end{aligned}
$$



Kleisli category $\mathbf{Top}_{\mathcal{H}}$

(An environment to study the effects of continuity over composition)

- $|\mathbf{Top}_{\mathcal{H}}| = |\mathbf{Top}|$,
- for any objects $I, O \in |\mathbf{Top}_{\mathcal{H}}|$,

$$\mathbf{Top}_{\mathcal{H}}(I, O) = \mathbf{Top}(I, \mathcal{H}O)$$

- the identity of I is η_I, and given two arrows $c_1 : I \to \mathcal{H}K$, $c_2 : K \to \mathcal{H}O$ their (sequential) composition,

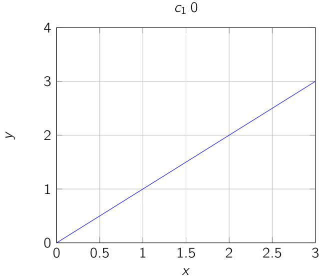
$$c_2 \bullet c_1 : I \to \mathcal{H}O$$

is equal to

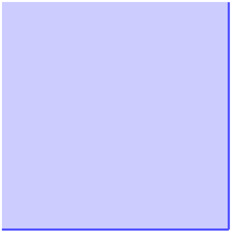
$$\mu \cdot \mathcal{H}c_2 \cdot c_1$$

Kleisli composition (of $\mathbf{Top}_{\mathcal{H}}$)

$$I \xrightarrow{c_1} \mathcal{H}K \xrightarrow{\mathcal{H}c_2} \mathcal{H}\mathcal{H}O \xrightarrow{\mu} \mathcal{H}O$$



$\overset{\mathcal{H}c_2}{\longmapsto}$



Other forms of composition (in $\mathbf{Top}_{\mathcal{H}}$)

Choice (coproduct)

$$\frac{c_1 : I_1 \to \mathcal{H}O,\, c_2 : I_2 \to \mathcal{H}O}{[c_1, c_2] : I_1 + I_2 \to \mathcal{H}O}\ (+)$$

Parallelism (pullback)

$$\frac{c_1 : I \to \mathcal{H}O_1,\, c_2 : I \to \mathcal{H}O_2}{\langle\langle c_1, c_2\rangle\rangle : I \to \mathcal{H}(O_1 \times O_2)}\ (\times)$$

These operators are (co)limits, hence a number of useful laws come for free !

Other forms of composition (in $\mathbf{Top}_{\mathcal{H}}$)

Choice (coproduct)

$$\frac{c_1 : I_1 \to \mathcal{H}O,\, c_2 : I_2 \to \mathcal{H}O}{[c_1, c_2] : I_1 + I_2 \to \mathcal{H}O}\ (+)$$

Parallelism (pullback)

$$\frac{c_1 : I \to \mathcal{H}O_1,\, c_2 : I \to \mathcal{H}O_2}{\langle\langle c_1, c_2\rangle\rangle : I \to \mathcal{H}(O_1 \times O_2)}\ (\times)$$

These operators are (co)limits, hence a number of useful laws come for free !

Other forms of composition (in $\mathbf{Top}_{\mathcal{H}}$)

Choice (coproduct)

$$\frac{c_1 : I_1 \to \mathcal{H}O,\, c_2 : I_2 \to \mathcal{H}O}{[c_1, c_2] : I_1 + I_2 \to \mathcal{H}O}\ (+)$$

Parallelism (pullback)

$$\frac{c_1 : I \to \mathcal{H}O_1,\, c_2 : I \to \mathcal{H}O_2}{\langle\langle c_1, c_2\rangle\rangle : I \to \mathcal{H}(O_1 \times O_2)}\ (\times)$$

These operators are (co)limits, hence a number of useful laws come for free !

Other forms of composition (in $\mathbf{Top}_{\mathcal{H}}$)

Synchronised parallelism

$$\frac{c_1 : I \to \mathcal{H}O_1,\, c_2 : I \to \mathcal{H}O_2}{(\!| c_1, c_2 |\!) : I \to \mathcal{H}(O_1 \times O_2)}\ (s)$$

Feedback

$$\frac{c : I \to \mathcal{H}I}{\nu c : I \to \mathcal{H}I}\ (\nu)$$

Other forms of composition (in $\mathbf{Top}_{\mathcal{H}}$)

Synchronised parallelism

$$\frac{c_1 : I \to \mathcal{H}O_1,\, c_2 : I \to \mathcal{H}O_2}{(\!| c_1, c_2 |\!) : I \to \mathcal{H}(O_1 \times O_2)}\ (s)$$

Feedback

$$\frac{c : I \to \mathcal{H}I}{\nu c : I \to \mathcal{H}I}\ (\nu)$$

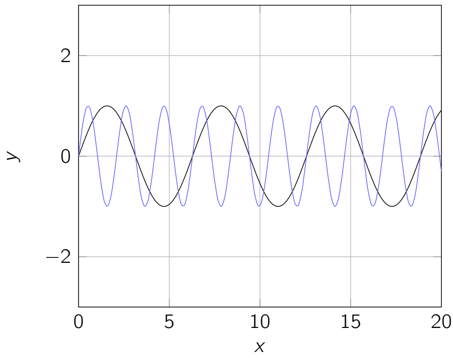
$\mathbf{Top}_{\mathcal{H}}$ (Parallelism)

$$c_1\, x = (\lambda t.\, x + (sin\, t), 20\ )$$
$$c_2\, x = (\lambda t.\, x + (sin(3 *\ t)), 20\ )$$

$\textbf{Top}_{\mathcal{H}}$ (Parallelism)

$$c_1\, x = (\lambda t.\, x + (sin\ t), 20\ )$$
$$c_2\, x = (\lambda t.\, x + (sin(3 *\ t)), 20\ )$$

$\langle\langle c_1, c_2\rangle\rangle\ 0$



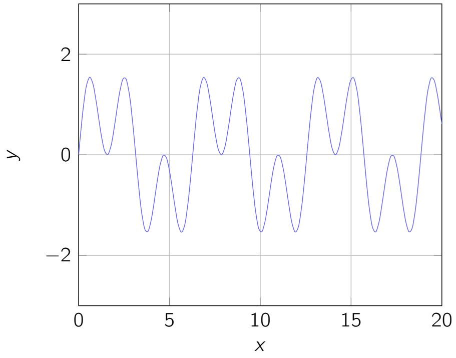
$\mathbf{Top}_{\mathcal{H}}$ (Parallelism)

$$c_3\,(x, y) = (\lambda t.\, x + y, 0)$$

$\mathbf{Top}_{\mathcal{H}}$ (Parallelism)

$$c_3\,(x, y) = (\lambda t.\, x + y, 0)$$

$c_3 \bullet \langle\langle c_1, c_2 \rangle\rangle\, 0$



Conclusions

- Our goal is a coalgebraic calculus of hybrid components
- and monad $\mathcal{H}$ seems to be a promising approach for this.

But mind

- Simulink, widely used in industry, and
- Hybrid automata, the standard formalism for the specification of hybrid systems.

- The former is highly expressive, but lacks a clear semantics.
- The latter is very intuitive, but does not have composition mechanisms as rich as Simulink.

Conclusions

- Our goal is a coalgebraic calculus of hybrid components
- and monad $\mathcal{H}$ seems to be a promising approach for this.

But mind

- Simulink, widely used in industry, and
- Hybrid automata, the standard formalism for the specification of hybrid systems.

- The former is highly expressive, but lacks a clear semantics.
- The latter is very intuitive, but does not have composition mechanisms as rich as Simulink.

Conclusions

- Our goal is a coalgebraic calculus of hybrid components
- and monad $\mathcal{H}$ seems to be a promising approach for this.

But mind

- Simulink, widely used in industry, and
- Hybrid automata, the standard formalism for the specification of hybrid systems.

- The former is highly expressive, but lacks a clear semantics.
- The latter is very intuitive, but does not have composition mechanisms as rich as Simulink.

Future work

- Introduction of non–determinism.
- Development of a calculus bisimulation–based.
- Try to answer the question:

  "Which kind of logics does monad $\mathcal{H}$ gives rise to?"

Future work

- Introduction of non–determinism.
- Development of a calculus bisimulation–based.
- Try to answer the question:

  "Which kind of logics does monad $\mathcal{H}$ gives rise to?"

Future work

- Introduction of non–determinism.
- Development of a calculus bisimulation–based.
- Try to answer the question:

  "*Which kind of logics does monad $\mathcal{H}$ gives rise to?*"

References I

Barbosa, L. S. (2001).
Components as coalgebras.
PhD thesis, DI, Minho University.

Haghverdi, E., Tabuada, P., and Pappas, G. J. (2005).
Bisimulation relations for dynamical, control, and hybrid systems.
Theoretical Computer Science, 342(2–3):229 – 261.

Höfner, P. (2009).
Algebraic calculi for hybrid systems.
PhD thesis, University of Augsburg.

References II

Jacobs, B. (2000).
Object-oriented hybrid systems of coalgebras plus monoid actions.
Theoretical Computer Science, 239(1):41 – 95.