Towards Constructive Hybrid Semantics

Tim Lukas Diezel  Sergey Goncharov

FAU Erlangen-Nürnberg

FSCD, Paris (I wish), July 2, 2020
Bouncing ball is a simple Newtonian system specified by differential equation \( \ddot{h} = -g \) \((g \approx 9.8)\) whose solution is

\[
h(t) = h_0 + v_0 t - \frac{gt^2}{2}
\]

with initial values:

- \( v_0 = 0, h_0 \neq 0 \) (peak height)
- \( h_0 = 0, v_0 \neq 0 \) (zero height)

**Features:**

- deterministic
- hybrid: the velocity changes discretely at the bottom \( v \leftrightarrow -cv \), but it changes continuously in the meanwhile
- progressive: every iteration consumes non-zero time
- Zeno behaviour: the state of rest is only reachable in the limit

**damping factor**
Bouncing ball can be formalized in idealized language HybCore: 

\[
\begin{align*}
  x &:= \lceil (1, 0) \rceil \text{ while true} \\
  \{ & \quad (h, v) := (x := t. \text{ball}(x, t) \& \text{fst } x \geq 0); \\
  & \quad \lceil (h, -c \, v) \rceil \\
  \} 
\end{align*}
\]

Here, \( \text{ball}(a, b, t) \) is the solution of the initial value problem 
\[
\begin{align*}
  \dot{h} = v, \quad \dot{v} = -g, \quad h(0) = a, \quad v(0) = b 
\end{align*}
\]

The critical element of the semantics is Elgot iteration: 
\[
(f : X \rightarrow T(Y \uplus X)) \mapsto (f^\dagger : X \rightarrow TY) \text{ for a suitable monad } T
\]

---

\(^1\)Sergey Goncharov and Renato Neves, An adequate while-language for hybrid computation (PPDP 2019)
We investigate hybrid semantics and its generalizations from the categorical and type-theoretic perspectives.

Relevant preceding work:

- Modelling partiality in intentional type theory under countable choice: James Chapman, Tarmo Uustalu, and Niccolò Veltri. *Quotienting the delay monad by weak bisimilarity* (ICTAC 2015)


Here we generalize by rendering hybridness as partiality extended over time.
Our main construction is implemented as a quotient inductive-inductive type in cubical Agda\(^2\) and is available under:

https://github.com/sergey-goncharov/hybrid-agda

Many insights were borrowed from the preceding implementation of the partialiy monad by Danielsson (http://www.cse.chalmers.se/~nad/listings/partiality-monad)

\(^2\)Andrea Vezzosi, Anders Mörtberg, and Andreas Abel. Cubical Agda: A dependently typed programming language with univalence and higher inductive types (ICFP 2019)
Outline

Hybrid semantics, non-constructively
Categorical abstraction
Characterizing classical semantics
Some details of Agda formalization
Conclusions
HYBRID SYSTEMS, NON-CONSTRUCTIVELY
**Definition (Monad)**

A monad $T$ (on $\text{Set}$) is given by a Kleisli triple $(T, \eta, -^*)$ where:

- $T$ sends sets to sets
- $\eta$ is a family of morphisms $\eta_X : X \rightarrow TX$, forming monad unit
- $(-)^*$ assigns to each $f : X \rightarrow TY$ a morphism $f^* : TX \rightarrow TY$

satisfying the laws: $\eta^* = \text{id}$, $f^* \eta = f$, $(f^* g)^* = f^* g^*$

This entails that $f : X \rightarrow TY$ and $g : Y \rightarrow TZ$ (regarded as programs) can be Kleisli composed to $g^* f : X \rightarrow TZ$

**Definition (Elgot Monad)**

$T$ with an iteration operator $(-)^\dagger : (X \rightarrow T(Y \sqcup X)) \rightarrow (X \rightarrow TY)$, satisfying axioms of iteration (omitted) is called Elgot
We distinguish

- **Duration semantics**: \( TX = \mathbb{R}_+ \times X \cup \bar{\mathbb{R}}_+ \) – a computation either converges in finite time with a value in \( X \), or diverges in possibly infinite time (\( \bar{\mathbb{R}}_+ = \mathbb{R}_+ \cup \{\infty\} \))

- **Evolution semantics**: \( TX = S^{[0,\mathbb{R}_+)} \times X \cup S^{[0,\bar{\mathbb{R}}_+)} \) where

  - \( S^{[0,\mathbb{R}_+)} = \sum_d: \mathbb{R}_+ \to S \) is the space of finite trajectories
  - \( S^{[0,\bar{\mathbb{R}}_+)} = \sum_d: \bar{\mathbb{R}}_+ \to S \) is the space of possibly infinite trajectories over \( S \)

**Idea for Abstraction**: Vault to general monoids instead of concrete \( \mathbb{R}_+, S^{[0,\mathbb{R}_+)} \)
Fix a (not necessarily commutative) monoid \((\mathbb{M}, +, 0)\)

A **monoid \(\mathbb{M}\)-module** is a set \(\mathbb{E}\) equipped with a map \(\triangleright: \mathbb{M} \times \mathbb{E} \to \mathbb{E}\), subject to the laws

\[
0 \triangleright e = e \quad (m + n) \triangleright e = m \triangleright (n \triangleright e)
\]

Every monoid-module pair \((\mathbb{M}, \mathbb{E})\) induces the **generalized writer monad**: \(T = \mathbb{M} \times (0) \cup \mathbb{E}\)

For example, with \(\mathbb{M} = \mathbb{E} = 1\) we obtain the **maybe monad** \((0) \cup \{\bot\}\), which is incidentally an Elgot monad

**Problem:** Elgotness relies on the law of excluded middle!
CATEGORICAL CONSTRUCTION
We extend previous construction of partiality monad \((M = 1)\)

Recall the concept of free object: Given a functor \(U : C \rightarrow \text{Set}\), an object \(FX\) is free on \(X\) if there is \((\eta_X : X \rightarrow UFX)_X\) and for every \(f : X \rightarrow UY\) there is unique \(f^* : FX \rightarrow Y\) such that

\[
\begin{array}{ccc}
UFX & \xrightarrow{Uf^*} & UY \\
\eta_X & \downarrow & \downarrow \\
X & \xrightarrow{f} & Y
\end{array}
\]

Standard facts:

- All free objects \(FX\) exist iff \(F\) is a left adjoint to \(U\)
- If all free objects exist then \((UF, \eta, (-)^*)\) constitutes a monad

---

\(^3\)Thorsten Altenkirch, Nils Danielsson, and Nicolai Kraus. Partiality, revisited – the partiality monad as a quotient inductive-inductive type
We assume $\mathbb{M}$ to be ordered with $0$ being the bottom and right monotone $+$

**Examples:** $1$, $\mathbb{N}$, $\mathbb{Q}_+$, $\mathbb{R}_+$, $S^{[0,\mathbb{R}_+)}$, $S^*$ (for last two $+$ is neither commutative, nor left monotone)

**Definition (Complete $\mathbb{M}$-Modules)**

An ordered $\mathbb{M}$-module is additionally equipped with a partial order $\sqsubseteq$ and $\bot$, such that

$$
\begin{align*}
\bot \sqsubseteq x & \quad x \sqsubseteq y \quad a \triangleright x \sqsubseteq a \triangleright y \quad a \leq b \\
& \quad a \triangleright \bot \sqsubseteq b \triangleright \bot
\end{align*}
$$

An ordered $\mathbb{M}$-module is **complete** if for any directed $(s_i)_i$ on $\mathbb{E}$ there is a least upper bound $\bigcup_i s_i$ and

$$
\bigcup_i a \triangleright s_i \sqsubseteq a \triangleright \bigcup_i s_i
$$
Theorem

Let $\mathbf{Alg}_L$ be the category of complete $\mathbb{M}$-modules and $U : \mathbf{Alg}_L \to \mathbf{Set}$ be the obvious forgetful functor

1. All free objects w.r.t. $U$ exist, yielding a monad $\tilde{L}$
2. $\tilde{L}$ is enriched over directed complete partial orders, and moreover, Kleisli composition is strict on both sides
3. $\tilde{L}$ is an Elgot monad with the iteration operator $(f : X \to \tilde{L}(Y \oplus X))^{\dagger}$ calculated as a least fixed point of the map $[(\eta, -)]*f : (X \to \tilde{L}Y) \to (X \to \tilde{L}Y)$

This is true both classically and constructively, thanks to

quotient inductive-inductive types\textsuperscript{4}

\textsuperscript{4}Thorsten Altenkirch, Paolo Capriotti, Gabe Dijkstra, Nicolai Kraus, and Fredrik Nordvall Forsberg. Quotient inductive-inductive types.
L VIA DIRECTED COMPLETION
Let $\mathbb{M}_X = \mathbb{M} \times (X \cup \{\bot\})$, and define $\triangleright_X$, $\sqsubseteq_X$ with rules:

\[
\begin{align*}
    a \triangleright_X (b, p) &= (a + b, p) \\
    (a, \text{inl} \ p) &\sqsubseteq_X (a, \text{inl} \ p) \\
    (a, \text{inr} \ \bot) &\sqsubseteq_X (b, p)
\end{align*}
\]

**Lemma**

*For any set $X$, $(\mathbb{M}_X, \triangleright_X, (0, \text{inr} \ \bot), \sqsubseteq_X)$ is an ordered $\mathbb{M}$-module.*

Intuitively, we are interested in limits of directed sequences over $\mathbb{M}_X$, e.g.

- **convergent**: $(1, \text{inr} \ \bot) \sqsubseteq_X \ldots \sqsubseteq_X (n, \text{inr} \ \bot) \sqsubseteq_X (n + 1, \text{inl} \ 0)$
- **divergent**: $(1, \text{inr} \ \bot) \sqsubseteq_X \ldots \sqsubseteq_X (1, \text{inr} \ \bot) \sqsubseteq_X \ldots$
- **Zeno**: $(1/2, \text{inr} \ \bot) \sqsubseteq_X \ldots \sqsubseteq_X (n/(n + 1), \text{inr} \ \bot) \sqsubseteq_X \ldots$
**Directed Completion**

- \((s_i)_i \preceq_X (t_i)_i\) if \(\forall i: \mathbb{N}. \exists j: \mathbb{N}. s_i \subseteq_X t_j\)
- \(s \sim_X t\) if \(s \preceq_X t\) and \(t \preceq_X s\)
- \(\tilde{M}_X\) is the quotient under \(\sim_X\) of the set of directed sequences over \(M_X\) and write \([s_i]_i\) instead of \([ (s_i)_i ]_\sim\)

**Theorem (Only Classically!)**

\((\tilde{M}_X, \triangleright_X, \perp_X, \preceq_X, \bigvee_X)\) is a free complete \(M\)-module on \(X\) with

- \([s]_\sim \preceq_X [t]_\sim\) if \(s \preceq_X t\)
- \(a \triangleright_X [s_i]_i = [a \triangleright_X s_i]_i\)
- \(\perp_X = [(0, \text{inr} \perp)]_i\)
- \(\bigvee_X [s_{i,j}]_j = [s_{\pi^{-1}_1(i), \pi^{-1}_2(i)}]_i\)

where \(\pi(x, y) = \frac{1}{2}(x + y)(x + y + 1) + x\)

Cantor pairing function
Let us write $\tilde{M}_0$ as $\tilde{M}$

**Theorem (Only Classically!)**

- $\tilde{L}X$ and $\tilde{M}_X$ are isomorphic as complete $M$-modules
- $\tilde{L}X \cong M \times X \cup \tilde{M}$ – *generalized writer monad over* $(M, \tilde{M})$

**Examples:**

- For $M = 1$, $\tilde{L}X = X \cup \{\bot\}$
- For $M = \mathbb{N}$, $\tilde{L}X = \mathbb{N} \times X \cup \tilde{N}$
- **But** for $M = \mathbb{R}_+$, $\tilde{L}X = \mathbb{R}_+ \times X \cup \tilde{R}_+$ where $\tilde{R}_+ \cong \mathbb{R}_+ \cup (\mathbb{R}_+ \setminus \{0\}) \cup \{\infty\} \cong \tilde{R}_+ \cup (\mathbb{R}_+ \setminus \{0\})$, because of Zeno behaviour!
A conservatively complete $\mathbb{M}$-module additionally satisfies

$$\bigcup_i a_i \triangleright \bot = \left( \bigvee_i a_i \right) \triangleright \bot$$

whenever $\bigvee_i a_i$ exists

- Again, we obtain an Elgot monad $\tilde{L}$ by the same argument
- There is $\tilde{\mathbb{M}}_X$ – an analogue of $\tilde{\mathbb{M}}_X$
- Again, $\tilde{L}X \cong \mathbb{M} \times X \cup \tilde{\mathbb{M}}$
- $\overline{\mathbb{R}}_+ = \mathbb{R}_+ \cup \{\infty\}$, $S^{(0,\mathbb{R}_+)} \cong S^{(0,\overline{\mathbb{R}}_+)}$, etc

Analogously, we introduce free conservatively complete $\mathbb{M}$-modules $\tilde{L}X$ on $X$, and obtain an Elgot monad $\tilde{L}$
MLTT and HoTT
**Formulas as Types**

In Martin-Löf Type Theory:

<table>
<thead>
<tr>
<th>Formula</th>
<th>Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>□</td>
<td>1</td>
</tr>
<tr>
<td>⊥</td>
<td>0</td>
</tr>
<tr>
<td>$A \land B$</td>
<td>$A \times B$</td>
</tr>
<tr>
<td>$A \lor B$</td>
<td>$A \uplus B$</td>
</tr>
<tr>
<td>$\forall x \to A(x)$</td>
<td>$\prod_{x : 1} A(x)$</td>
</tr>
<tr>
<td>$\exists \left[ x \right] A(x)$</td>
<td>$\sum_{x : 1} A(x)$</td>
</tr>
</tbody>
</table>

Some examples:

- A is a **(mere) proposition**: IsProp $A = \forall (x \, y : A) \to x \equiv y$
- A is a **set**: IsSet $A = \forall (x \, y : A) \to \text{IsProp} \ (x \equiv y)$
- A is **decidable**: IsDec $A = A \lor \neg A$
In HoTT and in Cubical Agda (which implements HoTT), we can express properly more, e.g. \textit{propositional truncation} as a quotient inductive type:

\begin{verbatim}
data _∥_∥ (A : Set ℓ) : Set ℓ where  
|_| : A → ∥ A ∥  
∥∥|-prop : IsProp ∥ A ∥
\end{verbatim}

\textbf{Axiom of Countable Choice:}

\[ ACω \{ℓ\} = \forall (P : \mathbb{N} \rightarrow \text{Set} \ ℓ) \rightarrow (\forall n \rightarrow ∥ P n ∥) \rightarrow ∥ (\forall n \rightarrow P n) ∥ \]

Implementing $\tilde{\mathcal{L}}$ and $\bar{\mathcal{L}}$ requires more sophisticated quotient inductive-inductive types
Chains, intensionally and extensionally directed sequences:

- **Inc** \( \sigma = \forall (n : \mathbb{N}) \rightarrow \sigma n \leq \sigma (\text{suc } n) \)
- **Dir** \( \sigma = \forall (n m : \mathbb{N}) \rightarrow \exists [k] (\sigma n \leq \sigma k \land \sigma m \leq \sigma k) \)
- **\|Dir\|** \( \sigma = \forall (n m : \mathbb{N}) \rightarrow \| \exists [k] (\sigma n \leq \sigma k \land \sigma m \leq \sigma k) \| \)

**Theorem**

Let (a), (b) and (c) stand for completeness of a fixed set \( A \) w.r.t. **\|Dir\|**, **Dir** and **Inc** correspondingly. Then

- (a) \( \Rightarrow \) (b) \( \Rightarrow \) (c)
- (b) \( \Rightarrow \) (a) under countable choice
- (c) \( \Rightarrow \) (a) under the decidability of \( \leq \) on \( A \) (i.e. under \( \forall (x y : A) \rightarrow \text{IsDec } (x \leq y) \))

Our \( \tilde{L} \) and \( \bar{L} \) are based on intensional completeness
FURTHER WORK
Further Work

- Implement the concept of free objects in cubical Agda
- Is it possible to rebase $\tilde{L}$ and $\bar{L}$ on extensional completeness (currently based on intensional completeness)? Which approach would be the right one if both are possible?
- Implement classical characterizations of $\tilde{L}$ and $\bar{L}$ in cubical Agda (possible under countable choice?)
- Further flavours of hybrid semantics, e.g. non-deterministic