An Adequate Semantics for Hybrid While

Sergey Goncharov[a] Renato Neves[b]

QAPL/ETAPS 2019, April 6, Prague

[a]Friedrich-Alexander-Universität Erlangen-Nürnberg
[b]INESC TEC (HASLab) & University of Minho

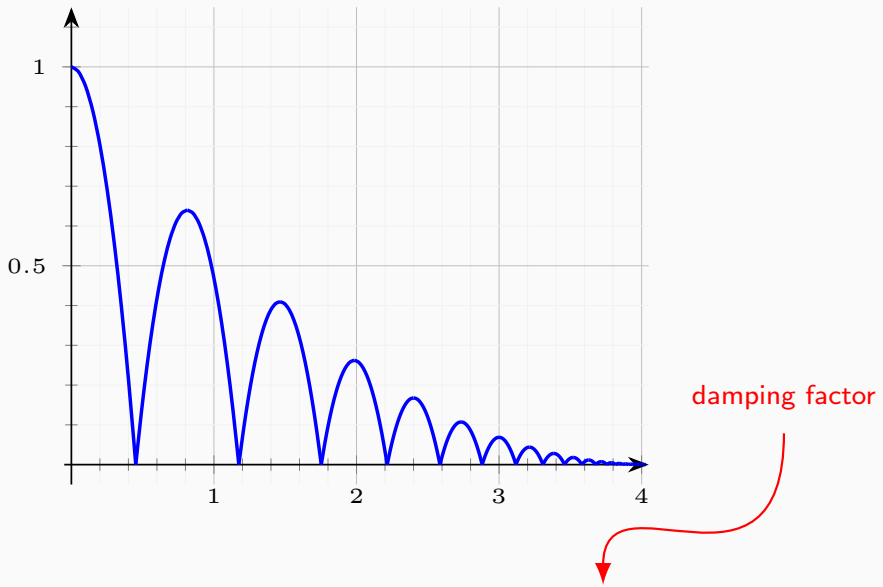
Example: Bouncing Ball

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 $v \mapsto -cv$, but it changes continuously in the meanwhile
- Zeno behaviour: the state of rest is only reachable in the limit

Semantics of Hybridness

Stepping stones:

- Hybrid automata [Alur, Courcoubetis, Henzinger, and Ho, 1993], [Henzinger, 1996]
- Platzer's differential dynamic logic [Platzer, 2008]
- Moggi's computational monads [Moggi, 1991]
- Kick, Power and Simpson's evolution comonad [Kick, Power, and Simpson, 2006]
- Fine-grain call-by-value [Levy, Power, and Thielecke, 2002]
- Hybrid monads (with iteration) [Neves, Barbosa, Hofmann, and Martins, 2016], [Goncharov, Jakob, and Neves, 2018]

Here:

- HybCore – a deterministic hybrid call-by-value while-language
- Adequate operational semantics w.r.t. hybrid and duration monads

Prelude: A Simple Adequate While-Language

A Simple While-Language

A simple call-by-value while-language:

- types are given with the grammar

$$A, B, \ldots ::= \mathbf{N} \mid 1 \mid 2 \mid A \times B$$

natural numbers

- value and computation judgments

$$\Gamma \vdash_{\mathsf{v}} v : A \qquad \text{and} \qquad \Gamma \vdash_{\mathsf{c}} p : A$$

variable context

variable context

(thus, we piggyback on fine-grain call-by-value[1])

- A signature of Σ operations $f : A \to B$ with atomic B

[1] Levy, Power, and Thielecke 2002, Modelling Environments in Call-By-Value Programming Languages

Term Formation

Values:

$$\frac{}{\Gamma \vdash_{\mathrm{v}} \star : 1} \qquad \frac{x : A \ \text{ in } \ \Gamma}{\Gamma \vdash_{\mathrm{v}} x : A} \qquad \frac{f : A \to B \in \Sigma \qquad \Gamma \vdash_{\mathrm{v}} v : A}{\Gamma \vdash_{\mathrm{v}} f(v) : B}$$

$$\frac{}{\Gamma \vdash_{\mathrm{v}} \mathsf{true} : 2} \qquad \frac{}{\Gamma \vdash_{\mathrm{v}} \mathsf{false} : 2} \qquad \frac{\Gamma \vdash_{\mathrm{v}} v : A \qquad \Gamma \vdash_{\mathrm{v}} w : B}{\Gamma \vdash_{\mathrm{v}} \langle v, w \rangle : A \times B}$$

Computations:

$$\frac{\Gamma \vdash_{\mathrm{c}} p : A \quad \Gamma, x : A \vdash_{\mathrm{c}} q : B}{\Gamma \vdash_{\mathrm{c}} x := p;\ q : B} \qquad \frac{\Gamma \vdash_{\mathrm{v}} v : A \times B \quad \Gamma, x : A, y : B \vdash_{\mathrm{c}} p : C}{\Gamma \vdash_{\mathrm{c}} \langle x, y \rangle := v;\ p : C}$$

$$\frac{\Gamma \vdash_{\mathrm{v}} v : A}{\Gamma \vdash_{\mathrm{c}} \lceil v \rceil : A} \qquad \frac{\Gamma \vdash_{\mathrm{v}} v : 2 \qquad \Gamma \vdash_{\mathrm{c}} p : A \qquad \Gamma \vdash_{\mathrm{c}} q : A}{\Gamma \vdash_{\mathrm{c}} \mathsf{if}\ v\ \mathsf{then}\ p\ \mathsf{else}\ q : A}$$

$$\frac{\Gamma \vdash_{\mathrm{c}} p : A \qquad \Gamma, x : A \vdash_{\mathrm{v}} v : 2 \qquad \Gamma, x : A \vdash_{\mathrm{c}} q : A}{\Gamma \vdash_{\mathrm{c}} x := p\ \mathsf{while}\ v\ \{q\} : A}$$

Small-Step Operational Semantics

Closed values, Closed computations:

$$v, w ::= x \mid \star \mid \mathsf{true} \mid \mathsf{false} \mid \langle v, w \rangle \mid f(v) \qquad (f \in \Sigma)$$

$$p, q ::= \langle x, y \rangle := \langle v, w \rangle;\ p \mid \mathsf{if}\ v\ \mathsf{then}\ p\ \mathsf{else}\ q \mid \lceil v \rceil$$
$$\mid x := p;\ q \mid x := p\ \mathsf{while}\ v\ \{q\}$$

Rules:

$$\frac{}{\langle x, y \rangle := \langle v, w \rangle;\ q \to q[v/x, w/y]} \qquad \frac{p \to p'}{x := p;\ q \to x := p';\ q}$$

$$\frac{}{x := \lceil v \rceil;\ q \to q[v/x]} \qquad \frac{}{\mathsf{if}\ \mathsf{true}\ \mathsf{then}\ p\ \mathsf{else}\ q \to p} \qquad \frac{}{\mathsf{if}\ \mathsf{false}\ \mathsf{then}\ p\ \mathsf{else}\ q \to q}$$

$$\frac{p \to p'}{x := p\ \mathsf{while}\ v\ \{q\} \to x := p'\ \mathsf{while}\ v\ \{q\}} \qquad \frac{w[v/x] = \mathsf{false}}{x := \lceil v \rceil\ \mathsf{while}\ w\ \{q\} \to \lceil v \rceil}$$

$$\frac{w[v/x] = \mathsf{true}}{x := \lceil v \rceil\ \mathsf{while}\ w\ \{q\} \to x := q[v/x]\ \mathsf{while}\ w\ \{q\}}$$

Big-Step Operational Semantics

$$\frac{}{\lceil v \rceil \Downarrow v} \qquad \frac{p \Downarrow v \quad q[v/x] \Downarrow w}{x := p;\ q \Downarrow w} \qquad \frac{p[v/x, w/y] \Downarrow u}{\langle x, y \rangle := \langle v, w \rangle;\ p \Downarrow u}$$

$$\frac{p \Downarrow v}{\text{if true then } p \text{ else } q \Downarrow v} \qquad \frac{q \Downarrow v}{\text{if false then } p \text{ else } q \Downarrow v}$$

$$\frac{p \Downarrow w \qquad v[w/x] = \text{false}}{x := p \text{ while } v \ \{q\} \Downarrow w}$$

$$\frac{p \Downarrow w \qquad v[w/x] = \text{true} \qquad x := q[w/x] \text{ while } v \ \{q\} \Downarrow u}{x := p \text{ while } v \ \{q\} \Downarrow u}$$

Proposition: $p \Downarrow w$ iff $p \to^\star \lceil w \rceil$

Monads

Definition (Monad)

A monad $\mathbb{T}$ over category $\mathbf{C}$ is given by a Kleisli triple $(T, \eta, -^\star)$ where

- T is an endomap on $|\mathbf{C}|$
- η is a family of morphisms $\eta_X : X \to TX$, called monad unit
- $(-)^\star$ assigns to each $f : X \to TY$ a morphism $f^\star : TX \to TY$

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

This means that the hom-sets $\mathrm{Hom}(X, TY)$ form a category (Kleisli category) under Kleisli composition $f \diamond g = f^\star\, g$ and $\eta_X \in \mathrm{Hom}(X, TX)$

Definition (Elgot Monad)

$\mathbb{T}$ with an iteration operator $(-)^\dagger : \mathrm{Hom}(X, T(Y + X)) \mapsto \mathrm{Hom}(X, TY)$, satisfying axioms of iteration (omitted) is called Elgot

Denotational Semantics

Assuming $\Gamma = (x_1 : A_1, \ldots, x_n : A_n)$, we interpret

$$[\![\Gamma \vdash_{\mathsf{v}} v : A]\!] : A_1 \times \ldots \times A_n \to A$$

$$[\![\Gamma \vdash_{\mathsf{c}} p : A]\!] : A_1 \times \ldots \times A_n \to TA$$

where $\mathbb{T}$ is the maybe monad $TX = X \cup \{\bot\}$

Kleisli star

unit

Moreover, e.g.

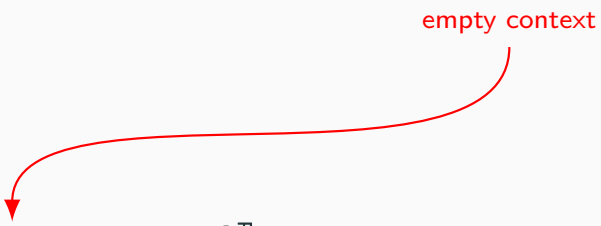
$$\frac{[\![\Gamma \vdash_{\mathsf{c}} p : A]\!] = h \qquad [\![\Gamma, x : A \vdash_{\mathsf{c}} q : B]\!] = u}{[\![\Gamma \vdash_{\mathsf{c}} x := p;\ q : B]\!] = \lambda \bar{x}.\, u^\star(\bar{x}, h(\bar{x}))} \qquad \frac{[\![\Gamma \vdash_{\mathsf{v}} v : A]\!] = h}{[\![\Gamma \vdash_{\mathsf{c}} \lceil v \rceil : A]\!] = \eta\, h}$$

$$\frac{[\![\Gamma \vdash_{\mathsf{c}} p : A]\!] = h \qquad [\![\Gamma, x : A \vdash_{\mathsf{v}} v : 2]\!] = b \qquad [\![\Gamma, x : A \vdash_{\mathsf{c}} q : A]\!] = l}{[\![\Gamma \vdash_{\mathsf{c}} x := p \text{ while } v\ \{q\} : A]\!] = \lambda \bar{x}.\big((\lambda x.\ \text{if } b(\bar{x}, x) \text{ then } \mathsf{inr}\, l(\bar{x}, x) \text{ else } \eta(\mathsf{inl}(x)))^\dagger\big)^\star(h(\bar{x}))}$$

using the fact that $\mathbb{T}$ is Elgot

Elgot iteration: $(f : X \to T(Y + X)) \mapsto (f^\dagger : X \to TY)$

Soundness and (Computational) Adequacy



Proposition (Soundness): if $p \Downarrow v$ then $[\![- \vdash_c p : A]\!] = v$

Proof Idea: induction over the derivation of $p \Downarrow v$

Proposition (Adequacy): if $[\![- \vdash_c p : A]\!] = v$ then $p \Downarrow v$

Proof Idea: induction over the denotational semantic rules

HybCore: Adding Hybrid Loops

A Hybrid While-Language

We simply add the type of real numbers $\mathbf{R}$, and the construct

$$\frac{\Gamma, t : \mathbf{R} \vdash_{\mathsf{v}} v : A \qquad \Gamma, x : A \vdash_{\mathsf{v}} w : 2}{\Gamma \vdash_{\mathsf{c}} x := t.\, v \;\&\; w : A}$$

Platzer's &

For example, for the bouncing ball behaviour:

velocity

height

$$\langle u, v\rangle := (\langle u, v\rangle := t.\, \mathsf{ball}(1, 0, t) \;\&\; u \geqslant 0)$$

$$\text{while } \mathsf{true} \; \{$$

$$\quad \langle u, v\rangle := \langle u, -0.8v\rangle$$

$$\quad \langle u, v\rangle := t.\, \mathsf{ball}(u, v, t) \;\&\; u \geqslant 0$$

$$\}$$

initial values

solution of $\{\dot{u} = v, \dot{v} = \mathrm{g}\}$

Types of Hybrid Loops

	Progressive	Non-progressive
Convergent	$x := 5$ while $x > 0$ $\{wait(1); x := x - 1\}$	$x := 5$ while $x > 0$ $\{x := x - 1\}$
Divergent	$x := 0$ while true $\{wait(1); x := x + 1\}$	$x := 0$ while true $\{x := x + 1\}$
Zeno	$x := 1$ while true $\{wait(x); x := x/2\}$	—

Big-Step Duration Semantics

- $p \Downarrow d, v$ means: p delivers a value v in time $d \in \mathbf{R}_+$
- $p \Downarrow v$ means: p diverges in time $d \in \bar{\mathbf{R}}_+ = \mathbf{R}_+ \cup \{\infty\}$

Some selected rules:

$$\frac{p \Downarrow d}{x := p;\ q \Downarrow d} \qquad \frac{p \Downarrow d, v \quad q[v/x] \Downarrow e}{x := p;\ q \Downarrow d + e} \qquad \frac{p \Downarrow d, v \quad q[v/x] \Downarrow e, w}{x := p;\ q \Downarrow d + e, w}$$

$$\frac{}{\lceil v \rceil \Downarrow 0, v} \qquad \frac{p \Downarrow d}{x := p \text{ while } v\ \{q\} \Downarrow d} \qquad \frac{p \Downarrow d, w \qquad v[w/x] = \text{false}}{x := p \text{ while } v\ \{q\} \Downarrow d, w}$$

Zeno behavior

$$\frac{(q_i \Downarrow d_i, w_i)_{i \in \omega} \qquad \forall i \in \omega.\ v[w_i/x] = \text{true} \wedge\ q_{i+1} = q[w_i/x]}{x := q_0 \text{ while } v\ \{q\} \Downarrow \sum_i d_i}$$

Duration Monad

Big-step semantics suggests to define the duration monad $\mathbb{Q}$:

$QX = \mathbf{R}_+ \times X \cup \bar{\mathbf{R}}_+$, $\eta(x) = \langle 0, x \rangle$, and

$$(f : X \to QY)^\star(d, x) = \langle d + e, y \rangle \quad \text{if} \quad f(x) = \langle e, y \rangle,$$
$$(f : X \to QY)^\star(d, x) = d + e \quad \text{if} \quad f(x) = e,$$
$$(f : X \to QY)^\star(d) = d.$$

Theorem: $\mathbb{Q}$ is an Elgot monad

Remarkably,

- $\mathbb{Q}$ is an instance of the generalized writer monad $TX = M \times X + E$ with a monoid M and an M-monoid module E
- Elgot iteration operator of $\mathbb{Q}$ is neither a least nor unique fixpoint

Soundness and Adequacy of Duration Semantics

Since our previous denotational semantics is generic w.r.t. an Elgot monad, we only add a clause for $[\![\Gamma \vdash_{\mathsf{c}} x := t.\, v \;\&\; w : A]\!]$ (omitted here)

Theorem (Soundness and Adequacy):

- $p \Downarrow d$ iff $[\![- \vdash_{\mathsf{c}} p : A]\!] = d \in QA$
- $p \Downarrow d, v$ iff $[\![- \vdash_{\mathsf{c}} p : A]\!] = \langle d, [\![- \vdash_{\mathsf{v}} v : A]\!] \rangle \in QA$

Big-Step Evolution Semantics

$p, t \Downarrow v$ means: p evaluates to v at time instant $t \in \mathbf{R}_+$

Some selected rules:

$$\frac{(p, t \Downarrow v_s)_{s \leqslant t} \quad (q[v_s/x], 0 \Downarrow w_s)_{s \leqslant t}}{x := p;\ q, t \Downarrow w_t} \qquad \frac{p \Downarrow d, v' \quad p, d \Downarrow v \quad q[v/x], t \Downarrow w}{x := p;\ q, d + t \Downarrow w}$$

$$\frac{p \Downarrow d \quad p, t \Downarrow w \quad t < d}{x := p \text{ while } v\ \{q\}, t \Downarrow w} \qquad \frac{p \Downarrow d, w' \quad p, d \Downarrow w \quad v[w/x] = \text{false}}{x := p \text{ while } v\ \{q\}, d \Downarrow w}$$

$$\frac{p \Downarrow d, w \quad v[w/x] = \text{true} \quad x := q[w/x] \text{ while } v\ \{q\}, t \Downarrow u}{x := p \text{ while } v\ \{q\}, d + t \Downarrow u}$$

Remarkably, while, is not an iterated sequential composition anymore (!)

The Hybrid Monad and Adequacy

The hybrid monad $\mathbb{H}$ is build on the following functor

$$HX = \{\langle \mathsf{cc}, d, e : [0,d] \to X\rangle \mid d \in \mathbf{R}_+\} \cup \{\langle \mathsf{cd}, d, e : [0,d] \to X\rangle \mid d \in \mathbf{R}_+\} \cup \{\langle \mathsf{od}, d, e : [0,d) \to X\rangle \mid d \in \bar{\mathbf{R}}_+\}$$

Kleisli lifting for the simple case $e : [0,d] \to X$ and $e'(x)(0) = x$ for all x:

$$(\alpha, d', e')^\star(\mathsf{cc}, d, e) = (\alpha, d + d', \lambda t.\, \text{if } \ t < d \ \text{ then } \ e(t) \ \text{ else } \ e'(t-d))$$

Theorem (Soundness and Adequacy): Given $- \vdash_{\mathsf{c}} p : A$,

1. $p, t \Downarrow v$ iff $[\![- \vdash_{\mathsf{c}} p : A]\!] = \langle \alpha, d, e\rangle$, $t < d$ and $e^t = v$
2. $p, d \Downarrow v$ and $p \Downarrow d, w$ iff $v = w$, $[\![- \vdash_{\mathsf{c}} p : A]\!] = \langle \mathsf{cc}, d, e\rangle$ and $e^d = v$

Further Work

- Transfer the duration and the hybrid monads from **Set** to **Top**
- Program logics for hybrid programs
- Principled combinations of hybridness with other effects (e.g. via monad transformers, colimits, tensors)
- Higher order hybrid semantics and hybrid recursion (long term goal)

Questions?

References

Rajeev Alur, Costas Courcoubetis, Thomas A. Henzinger, and Pei-Hsin Ho. Hybrid automata: An algorithmic approach to the specification and verification of hybrid systems. In Robert L. Grossman, Anil Nerode, Anders P. Ravn, and Hans Rischel, editors, *Workshop on the Theory of Hybrid Systems, Denmark. 1992*, volume 736 of *Lecture Notes in Computer Science*, pages 209–229. Springer, 1993.

Sergey Goncharov, Julian Jakob, and Renato Neves. A semantics for hybrid iteration. In Sven Schewe and Lijun Zhang, editors, *29th International Conference on Concurrency Theory (CONCUR 2018)*, 2018.

References II

Thomas A. Henzinger. The theory of hybrid automata. In *LICS96': Logic in Computer Science, 11th Annual Symposium, New Jersey, USA, July 27-30, 1996*, pages 278–292. IEEE, 1996.

Marco Kick, John Power, and Alex Simpson. Coalgebraic semantics for timed processes. *Inf. Comput.*, 204(4):588–609, 2006.

Paul Blain Levy, John Power, and Hayo Thielecke. Modelling environments in call-by-value programming languages. *Inf. & Comp*, 185:2003, 2002.

Stefan Milius. Completely iterative algebras and completely iterative monads. *Inf. Comput.*, 196(1):1–41, 2005.

Eugenio Moggi. Notions of computation and monads. *Inf. Comput.*, 93: 55–92, 1991.

References III

Renato Neves, Luís S. Barbosa, Dirk Hofmann, and Manuel A. Martins. Continuity as a computational effect. *Journal of Logical and Algebraic Methods in Programming*, 85(5):1057–1085, 2016.

André Platzer. Differential dynamic logic for hybrid systems. *J. Automated Reasoning*, 41(2):143–189, Aug 2008.

Tarmo Uustalu. Generalizing substitution. *ITA*, 37:315–336, 2003.

Equivalence of Semantics and Determinacy

Proposition: Big-step and small-step semantics are equivalent as follows:

$$p \Downarrow w \qquad \text{iff} \qquad p \to^\star \lceil w \rceil$$

Proof Idea: for the ($\Rightarrow$) direction, induction over the derivation of $p \Downarrow w$; for the ($\Leftarrow$) direction, prove

Lemma: $p \to q$ with $q \Downarrow w$ imply $p \Downarrow w$

.. and then induction over the length of $p \to^\star \lceil w \rceil$

Proposition (Determinacy): $p \Downarrow v$ for at most one v

Small-Step Duration Semantics

Small-step reduction $\xrightarrow{d}$ is now indexed by durations d. We reinterpret $\rightarrow$ as $\xrightarrow{0}$ when appropriate, otherwise override the rules:

$$\frac{p \xrightarrow{d} p'}{x := p;\ q \xrightarrow{d} x := p';\ q} \qquad \frac{p \xrightarrow{d}}{x := p;\ q \xrightarrow{d}}$$

$$\frac{\forall s \leqslant d.\, w[v[s/t]/x] = \mathsf{true} \qquad \forall e > 0.\, \exists s \in (d, d+e).\, w[v[s/t]/x] = \mathsf{false}}{x := t.\, v \ \&\ w \xrightarrow{d} \lceil v[d/t] \rceil}$$

$$\frac{\forall s < d.\, w[v[s/t]/x] = \mathsf{true} \qquad w[v[d/t]/x] = \mathsf{false}}{x := t.\, v \ \&\ w \xrightarrow{d}}$$

$$\frac{p \xrightarrow{d}}{x := p \ \mathsf{while}\ v\ \{q\} \xrightarrow{d}} \qquad \frac{p \xrightarrow{d} p'}{x := p \ \mathsf{while}\ v\ \{q\} \xrightarrow{d} x := p' \ \mathsf{while}\ v\ \{q\}}$$

$$\frac{w[v/x] = \mathsf{true}}{x := \lceil v \rceil \ \mathsf{while}\ w\ \{q\} \xrightarrow{0} x := q[v/x] \ \mathsf{while}\ w\ \{q\}}$$

Equivalence of Big-Step and Small-Step

We define $\xrightarrow{d}{}^{*}$ as a "transitive closure" of $\xrightarrow{d}$

Theorem: $p \xrightarrow{d}{}^{*} \lceil v \rceil$ iff $p \Downarrow d, v$ and $p \xrightarrow{d}{}^{*}$ iff $p \Downarrow v$

Proof Idea: same same (but different)

Towards Elgotness

$\mathbb{Q}$ yields neither inductive nor coinductive semantics:

- we cannot order-enrich $\mathbb{Q}$, for there is no canonical choice of divergence among $\bar{\mathbf{R}}_+ \subseteq QX$
- durations play roles of observables, but the semantics does not distinguish $p \xrightarrow{d} q \xrightarrow{e} r$ from $p \xrightarrow{d+e} r$

We thus define more fine grained layered duration monad

$\widehat{Q}X = \nu\gamma.\,(X + \mathbf{R}_+ \times \gamma)$, which induces a monad $\mathbb{Q}$ by a general argument[2]

[2] Uustalu 2003, Generalizing Substitution

Layered Duration Monad

$\widehat{Q}$ is an analogue of the delay monad. It also simplifies in ZFC:

$\widehat{Q}X = \mathbf{R}_+^\star \times X \cup \mathbf{R}_+^\omega$, $\eta(x) = \langle \epsilon, x \rangle \in QX$, and

$$(f : X \to \mathbf{R}_+^\star \times Y \cup \mathbf{R}_+^\omega)^\star(w, x) = \langle uw, y \rangle \quad \text{if} \quad f(x) = \langle u, y \rangle \in \mathbf{R}_+^\star \times Y$$
$$(f : X \to \mathbf{R}_+^\star \times Y \cup \mathbf{R}_+^\omega)^\star(w, x) = uw \quad \text{if} \quad f(x) = u \in \mathbf{R}_+^\omega$$
$$(f : X \to \mathbf{R}_+^\star \times Y \cup \mathbf{R}_+^\omega)^\star(w) = w$$

$\widehat{Q}$ is completely iterative[3], i.e. for every guarded $f : X \to \widehat{Q}(Y + X)$, there is unique solution $f^\dagger : X \to \widehat{Q}Y$ of equation $f^\dagger = [\eta, f^\dagger]^\star f$,

where $f : X \to \widehat{Q}(Y + Z)$ is guarded if it factors as

$$X \to (\mathbf{R}_+^\star \times Y) \cup (\mathbf{R}_+^+ \times Z) \cup \mathbf{R}_+^\omega \hookrightarrow (\mathbf{R}_+^\star \times Y) \cup (\mathbf{R}_+^\star \times Z) \cup \mathbf{R}_+^\omega \cong Q(Y + Z)$$

[3] Milius 2005, Completely iterative algebras and completely iterative monads

Duration Monad as a Quotient

Let $\widehat{Q}_{\approx}X$ be the quotient of $\widehat{Q}X$ under the "weak bisimulation" relation $\approx$ generated by the clauses

$$\langle x, r_1 \dots r_n \rangle \approx \langle x, s_1 \dots s_m \rangle, \quad r_1 \dots \approx s_1 \dots, \quad \left(\text{where } \sum\nolimits_i r_i = \sum\nolimits_j s_j\right)$$

Let $\rho_X : \widehat{Q}X \to \widehat{Q}_{\approx}X$ be the emerging quotienting map:

$$\rho_X(x, r_1 \dots r_n) = \left\langle x, \sum\nolimits_i r_i \right\rangle, \qquad \rho_X(r_1 r_2 \dots) = \sum\nolimits_i r_i,$$

Theorem:

- ρ has a right inverse $\upsilon : \widehat{Q}_{\approx} \to \widehat{Q}$
- (ρ, υ) is an iteration-congruent retraction, hence $\widehat{Q}_{\approx}$ is Elgot
- $\widehat{\mathbb{Q}}_{\approx} \cong \mathbb{Q}$