Monsters and Lax Sigma-universes

Chris Purdy (Royal Holloway University of London)

Monsters and their properties

Monadic streams are a generalisation of the stream data type:

$$\mathbb{S}_{\mathbf{Id}} A := \nu X. A \times X$$
$$\mathbb{S}_M A := \nu X. M(A \times X)$$

They are parameterised by a functor M, and can be instantiated to many interesting (non-well founded) data types.

For example:

- If $M = 1 + -$, then $\mathbb{S}_M$ is the lazy list functor
- $\mathbb{S}_{R \to -} A$ is the type of potentially infinite state machines with input alphabet R and output alphabet A
- $\mathbb{S}_{\mathbf{IO}} A$ is the type of (never-terminating) processes that output elements of type A
- $\mathbb{S}_{\mathbf{List}} A$ is the type of non-well-founded branch-labelled trees over A.

Monsters and their properties

I have been interested in studying the theoretical properties of monadic streams (or "monsters" for short), initially to assist in designing a Haskell library of helper functions for monsters, and more recently just for the sheer fun/hell of it.

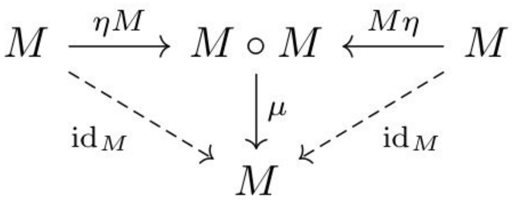
In particular, I wanted to know how different structures over M relate to those over $\mathbb{S}_M$. Inspired by typical Haskell type classes, I pursued the following questions:

- If M is a functor, is $\mathbb{S}_M$ also? Fairly obvious - a generalisation can be found in Varmo Vene's PhD thesis
- If M is a lax monoidal functor, is $\mathbb{S}_M$ also? Requires the monoid to have a four middle interchange law - this and the above I've formalised in Agda (using `agda-categories`)
- If M is representable, is $\mathbb{S}_M$ also? If M has representing object R, then M-monsters are representable by non-empty lists of R
- If M is a monad, is $\mathbb{S}_M$ also? Not in general - the state monad is a counter-example

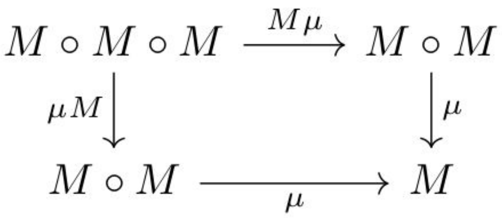
$$\mathbb{S}_M A := \nu X.M(A \times X)$$

A brief monad refresher

A monad is a triple (M, η, μ), where $M : \mathcal{C} \to \mathcal{C}$ is a functor, and $\eta : 1 \Rightarrow M$ and $\mu : M \circ M \Rightarrow M$ are natural transformations obeying the following laws:



Left and right identity laws



Associativity law

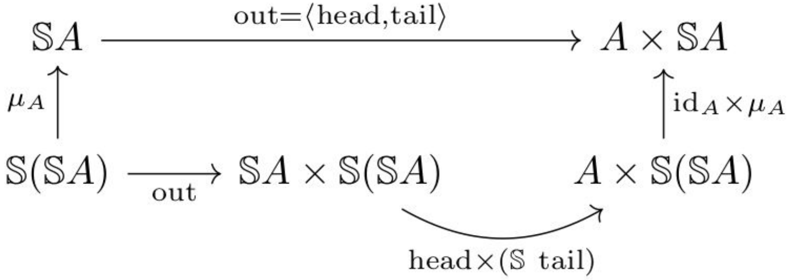
This makes M a monoidal object in the endofunctor category $[\mathcal{C}, \mathcal{C}]$.

Why could it be a monad?

The fourth question was inspired by the fact that pure streams $\mathbb{S}_{\mathbf{Id}}$ form a monad:

$$\begin{aligned} \text{out } (\eta_A \; a) &:= a, \eta_A \; a \\ \text{out } (\mu_A \; ss) &:= \text{head } (\text{head } ss), \mu_A \; ((\mathbb{S}_{\mathbf{Id}} \text{ tail}) \; (\text{tail } ss)) \end{aligned}$$

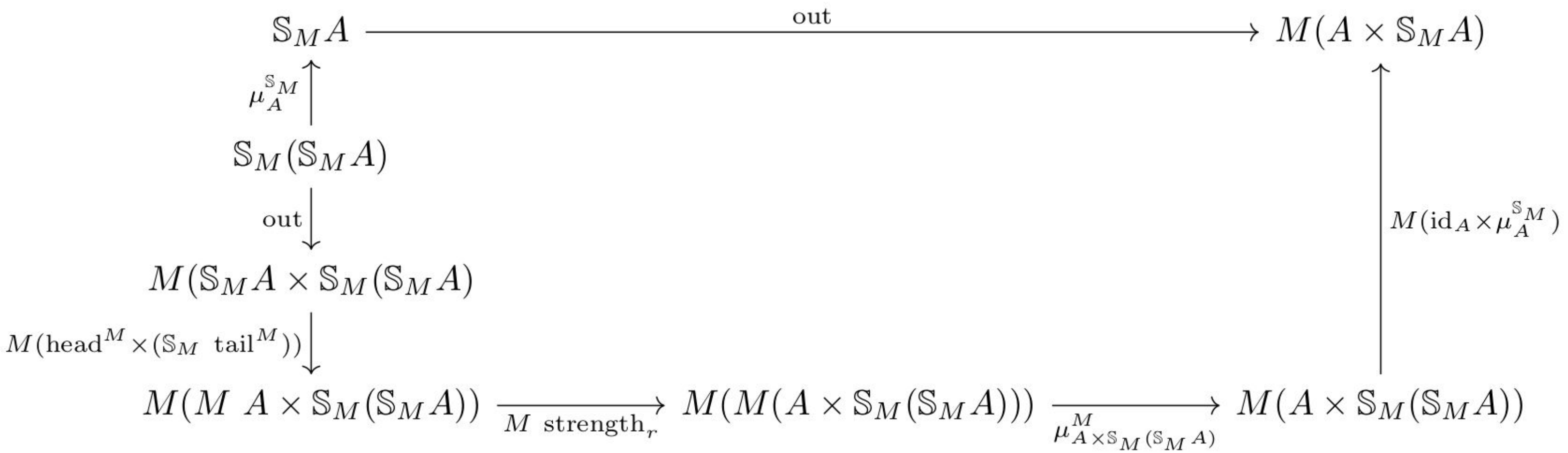
Intuitively, η_A repeats indefinitely the given element of A, and μ_A takes the diagonal of a stream of streams.



μ for pure streams

$\mathbb{S}_M A := \nu X.M(A \times X)$

Why could it be a monad?



μ for monadic streams over M. Notice that if $M = \mathbf{Id}$ this degenerates to the above square for pure streams

Where head and tail for monsters can be defined as follows:

$$\text{head}_A^M := M\pi_1 \circ \text{out}$$
$$\text{tail}_A^M := \text{in} \circ \mu^M_{A \times \mathbb{S}_M A} \circ M(\text{out} \circ \pi_2) \circ \text{out}$$

$$\mathbb{S}_M A := \nu X. M(A \times X)$$

Monad counter-example

$\eta^{\mathbb{S}_M}$ and $\mu^{\mathbb{S}_M}$ are not law-abiding in general. It is fairly easy to construct a counter-example for the left-unit and associativity laws using the state monad $M = S \to - \times S$.

Concretely, if you choose $S = \mathbb{N}$ and define the following state-monster:

$$\mathrm{out}(s\ n) = \lambda m.((m, s\ (n+1)), m+n)$$

You will find that the left-identity law doesn't hold:

$$\mu^{\mathbb{S}_M}_{\mathbb{N}}\ (\eta^{\mathbb{S}_M}_{\mathbb{S}_M \mathbb{N}}\ (s\ 1)) \neq s\ 1$$

This can be seen by "unfolding" the state-monsters on the left and right of the equation.

$\mathbb{S}_{S \to (- \times S)}\ A := \nu X.S \to ((A \times X) \times S)$

Revising the theorem

The question then became, what is the weakest restriction on M such that $\mathbb{S}_M$ is a monad? Since all representable functors are monads, I knew that M being representable was one possible restriction, so I decided to check whether that was the weakest one - i.e. show that M is representable if and only if $\mathbb{S}_M$ is a monad.

I first tackled this problem via brute force and commuting diagrams, but I found I couldn't get my hands on a possible representing object. So I went for a different approach, based on the fact that we are restricting our view to container functors, and after coming across the slides for Thorsten's 2018 talk "Monadic containers and universes":

A brief container refresher

Container functors (polynomial functors on **Set**) are a class of strictly-positive type, for which fixed-points are guaranteed to exist. A container $S \lhd P$ is given by:

- A set S of "shapes"
- A family $P : S \to \mathbf{Set}$ of "positions"

You can interpret a container $S \lhd P$ as a functor:

$$[\![S \lhd P]\!]\ A := \sum_{s:S} P\ s \to A$$

$$[\![S \lhd P]\!]\ f\ (s, g) := (s, f \circ g)$$

A container's set of shapes is a singleton iff its functor interpretation is representable.

Container representation of monsters

Given a container $S \triangleleft P$, its monster-container is $\mathrm{M}_{S,P} \triangleleft \mathrm{Path}_{S,P}$ where $\mathrm{M}_{S,P}$ is the greatest fixed-point of $[\![S \triangleleft P]\!]$, and $\mathrm{Path}_{S,P}$ is defined inductively by:

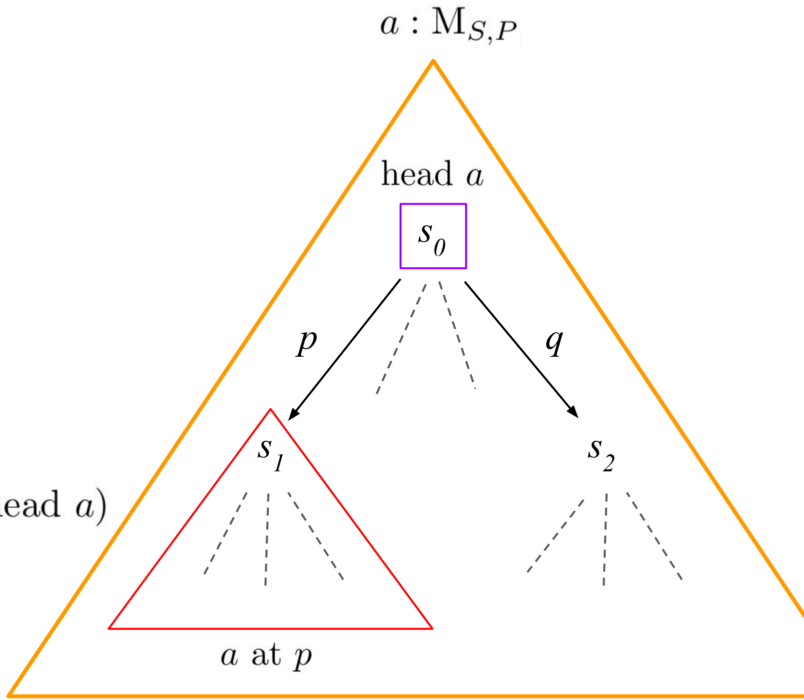
$$\frac{p : P(\text{head } a)}{\text{end } p : \mathrm{Path}_{S,P}\ a}$$

$$\frac{p : P(\text{head } a) \qquad \gamma : \mathrm{Path}_{S,P}\ (a \text{ at } p)}{\text{step } (p, \gamma) : \mathrm{Path}_{S,P}\ a}$$

where $a : \mathrm{M}_{S,P}$, and:

- $(\text{head } a) : S$ is the "root" of a
- $(a \text{ at } p) : \mathrm{M}_{S,P}$ is the "sub-tree" of a at position $p : P(\text{head } a)$

(These are just the destructor of $\mathrm{M}_{S,P}$ split into two maps)



Lax Sigma-universes

Lax Σ-universes are another perspective on monadic containers, where you can reason algebraically about the shapes and positions. A lax Σ-universe is given by:

$$\begin{aligned} U &: \mathbf{Set} \\ \mathrm{El} &: U \to \mathbf{Set} \\ \iota &: U \\ \sigma &: \Pi_{a:U}.(\mathrm{El}\ a \to U) \to U \\ \mathrm{pr} &: \mathrm{El}\ (\sigma\ a\ b) \to \Sigma_{x:\mathrm{El}\ a}\mathrm{El}(b\ x) \end{aligned}$$

Such that:

$$\begin{aligned} a \otimes \iota &= a \\ \iota \otimes b &= b \\ \sigma\ a\ (\lambda x.\sigma\ (b\ x)\ (c\ x)) &= \sigma\ (\sigma\ a\ b)\ (\lambda x.c\ (\mathrm{pr}_1 x)\ (\mathrm{pr}_2 x)) \end{aligned}$$

$a \otimes b$ is shorthand for $\sigma\ a\ (\lambda\_.b)$

pr_i is shorthand for $\pi_i \circ \mathrm{pr}$

There is a bijection between lax Σ-universes and monadic containers $[\![U \triangleleft \mathrm{El}]\!]$.

A monster universe

Given a lax Σ-universe for $S \triangleleft P$, we can now define the lax Σ-universe for $\mathrm{M}_{S,P} \triangleleft \mathrm{Path}_{S,P}$ that corresponds to the monad definition for $\mathbb{S}_M$ given earlier.

$$\begin{aligned} \text{head } \iota_\infty &:= \iota \\ \iota_\infty \text{ at } \_ &:= \iota_\infty \end{aligned}$$

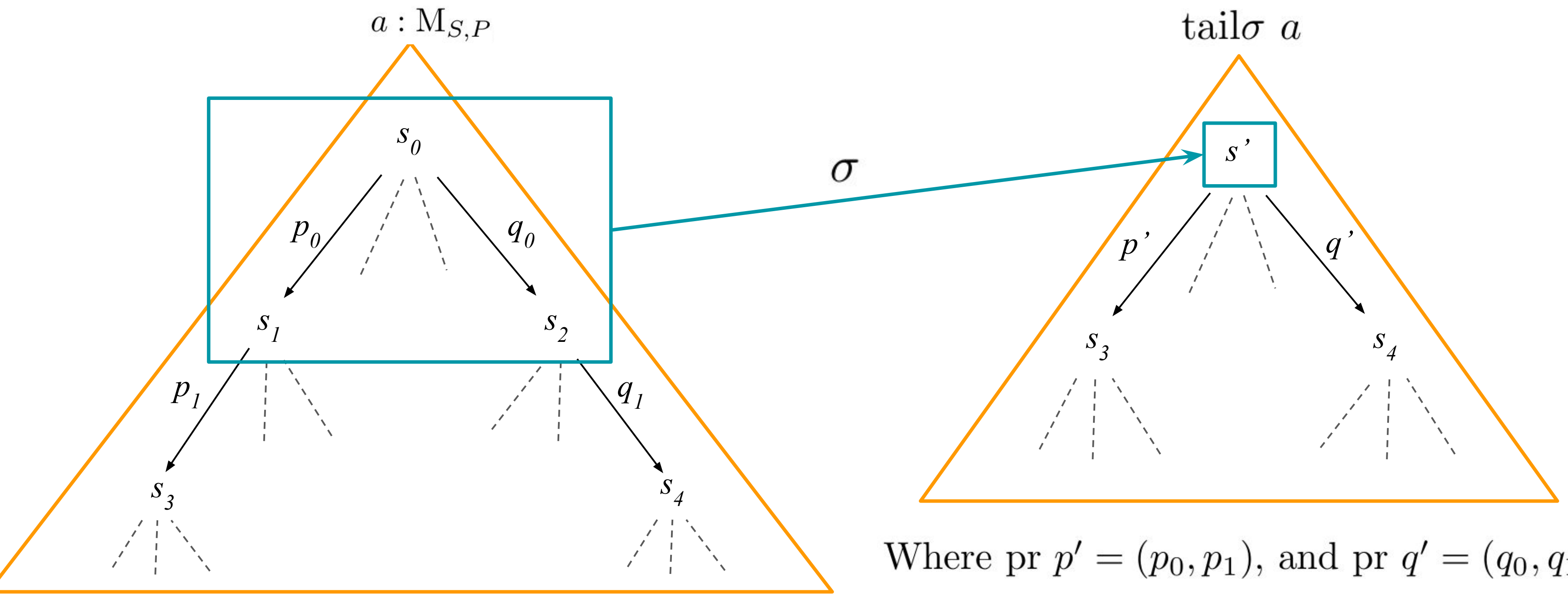
$$\begin{aligned} \text{head } (\text{tail}\sigma \ a) &:= \sigma \ (\text{head } a) \ (\lambda p.\text{head } (a \text{ at } p)) \\ (\text{tail}\sigma \ a) \text{ at } p &:= (a \text{ at } (\mathrm{pr}_1 \ p)) \text{ at } (\mathrm{pr}_2 \ p) \end{aligned}$$

$$\begin{aligned} \text{head } (\sigma_\infty \ a \ b) &:= \sigma \ (\text{head } a) \ (\lambda p.\text{head } (b \ (\text{end } p))) \\ (\sigma_\infty \ a \ b) \text{ at } p &:= \sigma_\infty \ (a \text{ at } (\mathrm{pr}_1 \ p)) \ (\lambda \gamma.\text{tail}\sigma \ (b \ (\text{step } (\mathrm{pr}_1 \ p, \gamma)))) \end{aligned}$$

A monster universe - making sense of $\mathrm{tail}\sigma$

$$\mathrm{head}\ (\mathrm{tail}\sigma\ a) := \sigma\ (\mathrm{head}\ a)\ (\lambda p.\mathrm{head}\ (a\ \mathrm{at}\ p))$$

$$(\mathrm{tail}\sigma\ a)\ \mathrm{at}\ p := (a\ \mathrm{at}\ (\mathrm{pr}_1\ p))\ \mathrm{at}\ (\mathrm{pr}_2\ p)$$



Where $\mathrm{pr}\ p' = (p_0, p_1)$, and $\mathrm{pr}\ q' = (q_0, q_1)$

A monster universe - making sense of σ_∞

$$\text{head}\ (\sigma_\infty\ a\ b) := \sigma\ (\text{head}\ a)\ (\lambda p.\text{head}\ (b\ (\text{end}\ p)))$$

$$(\sigma_\infty\ a\ b)\ \text{at}\ p := \sigma_\infty\ (a\ \text{at}\ (\text{pr}_1\ p))\ (\lambda\gamma.\text{tail}\sigma\ (b\ (\text{step}\ (p, \gamma))))$$

A monster universe - left-unit law

Recall that at this point, we are trying to prove that $[\![S \triangleleft P]\!]$ is representable, if its monster-container is a monad. For the latter to be the case, we require that these definitions obey (among other things) the left-identity law:

$$\sigma_\infty \; \iota_\infty \; (\lambda\_.a) = a$$

It turns out that this law holds iff the following equation does:

$$a \text{ at } p = \text{tail}\sigma \; a$$

which is quite a strong requirement!

Concluding the proof

This is where things get non-constructive. We can proceed by considering two cases:

- For all $s : S$, $P\ s$ has at least one element
- There is at least one $s : S$, where $P\ s$ is empty

In the first case, we can prove that for all $s : S$, $s = \iota$. This means that $[\![S \triangleleft P]\!]$ is representable by $P\ \iota$!

In the second case, we can construct a counter-example to associativity for σ_∞, leading to a contradiction.

Further work

There are still some holes in the proof that I would like to fill.

For starters, I've taken it as implied that the only possible definition of μ for $\mathbb{S}_M$ is taking the diagonal (in a specific way). I have a partial proof of there being a unique monad definition for pure streams, which might imply that there only one definition of μ that works, but formalising this seems difficult.

The translation of functions on $\mathbb{S}_M A$ to functions within the lax Σ-universe for monster-containers also needs to be formalised.