MONOIDAL STREAMS

Mario Román joint with Elena Di Lavore, Giovanni de Felice

May 31st, 2022

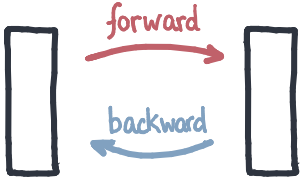
INTERCATS

Supported by the European Union through the ESF Estonian IT Academy Research Measure.

Motivation

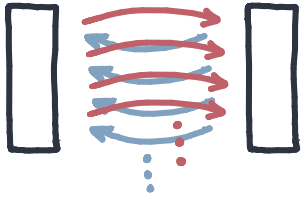
Monoidal lenses (optics) model single, bidirectional, interactions.

- Database update,
- Open games,
- Bayesian update.



How to model repeated interactions with optic-like constructions? We would want

- Dataflow programming,
- Repeated games,
- Stochastic processes.



Much literature on these ideas presented at Intercats.

Motivation: Dataflow Programming

Dataflow programming is a paradigm for repeated processes: every declaration is a sequence of values.

"followed by" "delayed stream"

$$fib = 0 \ \text{FBY} \ (fib + 1 \ \text{FBY} \ \text{WAIT} \ fib)$$
$$nat = 0 \ \text{FBY} \ (1 + nat)$$

- Elegant 'recursive' dataflow syntax.
- Signal flow, 'trace-like' diagrams, see FIGURE.
- Semantics: causal streams for the cartesian case.

E.g. Lustre, Lucid.

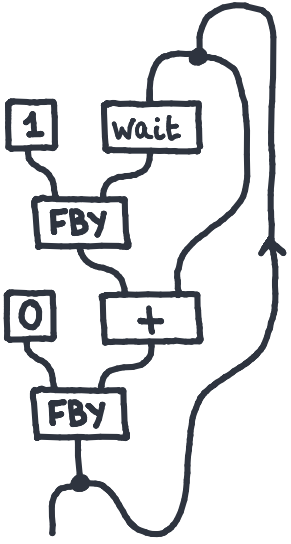


FIG. Signal flow graph.

Causal Stream Functions

DEFINITION. A *causal stream function* $f : X \to Y$ is a family of functions $f_n : X_0 \times \cdots \times X_n \to Y_n$.

$f_0 : X_0 \to Y_0$
$f_1 : X_0 \times X_1 \to Y_1$
$f_2 : X_0 \times X_1 \times X_2 \to Y_2$
$\cdots$

These form a monoidal category, the cokleisli category of the non-empty list monoidal comonad,

$$\mathrm{List}^+ : [\mathbb{N}, \mathrm{SET}] \to [\mathbb{N}, \mathrm{SET}], \quad \mathrm{List}^+(X)_n = \prod_{i=0}^{n} X_i .$$

We have

- a delay functor taking $X = (X_0, X_1, X_2, \ldots)$ into $\partial X = (1, X_0, X_1, \ldots)$;
- a trace-like operator taking $\partial S \otimes X \to S \otimes Y$ into $X \to Y$;
- coalgebraic reasoning and coinductive arguments.

T. Uustalu, V. Vene. Comonadic Notions of Computation.
D. Sprunger, S. Katsumata. Differentiable Causal Computations via Delayed Trace

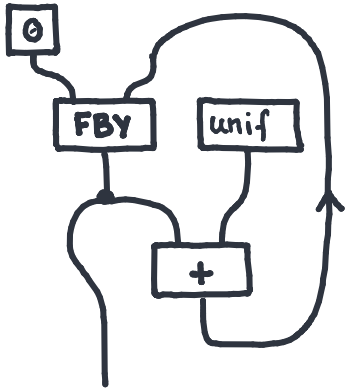
Motivation: Dataflow Programming

What about the non-cartesian case?

walk = 0 FBY UNIFORM{-1,1} + walk

We imagine what the syntax should be, but, what should replace streams in the semantics?

These will be monoidal streams.



Can we extend the comonad to monoidal categories? No. We need a different solution.

THEOREM (DLdFR). The non-empty list functor $\mathrm{List}^{+}(X)_n = \bigotimes_{i=0}^{n} X_i$ is a monoidal comonad if and only if $(\otimes)$ is a cartesian product.

Motivation

In the process interpretation of monoidal categories, morphisms $A \to B$ are processes with an input A and an output B.

However, most processes (servers, drivers, agents,...) are continuously taking inputs and producing outputs,

closed diagram vs

open,[1] repeated[2] diagram.

Román.[1] Open Diagrams via Coend Calculus. ACT'20.

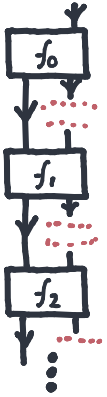
Román.[2] Comb Diagrams for Discrete-Time Feedback. Preprint.

Motivation

Monoidal lenses (optics) model single, bidirectional, interactions.

- Database update,
- Open games,
- Bayesian update.

How to model repeated interactions with optic-like constructions? We would want



- Dataflow programming,
- Repeated games,
- Stochastic processes.

Much literature on these ideas presented at Intercats.

Motivation

Today, given any symmetric monoidal category $(\mathbb{C}, \otimes, I)$, we will build a symmetric monoidal category of stream processes $\mathbf{Stream}(\mathbb{C})$ such that

- $\mathbf{Stream}(\mathbb{C})$ has an id-on objs functor from $[\mathbb{N}, \mathbb{C}]$;
- $\mathbf{Stream}(\mathbb{C})$ has a delay monoidal functor, ∂;
- $\mathbf{Stream}(\mathbb{C})$ has delayed feedback taking $\partial S \otimes X \to S \otimes Y$ into $X \to Y$;
- $\mathbf{Stream}(\mathbb{C})$ has a coalgebraic description;

- $\mathbf{Stream}(\mathbb{C})$ is cartesian when $\mathbb{C}$ is;
- $\mathbf{Stream}(\mathbf{Set})$ is the classical causal streams;
- $\mathbf{Stream}(\mathbf{Stoch})$ is causal discrete stochastic processes.

- $\mathbf{Stream}(\mathbb{C})$ is symm. premonoidal, effectful or Freyd when $\mathbb{C}$ is.

Monoidal Streams for Dataflow Programming

ArXiv: 2202.02061, to be presented at LiCS'22.

Elena Di Lavore
Tallinn University of Technology.

Giovanni de Felice
University of Oxford + Quantinuum

Mario Román
Tallinn University of Technology.

Synopsis

Three definitions from universal properties, and three explicit constructions. Each one a quotient of the previous.

1. Intensional streams, a first naïve version. Fail to form a category.
2. Extensional streams, a free category with feedback.
3. Observational streams, definitive solution to a fixpoint equation.

Two known particular cases, and an avenue for more.

1. Cartesian monoidal streams $(\mathbf{Set}, \times)$ are causal functions (as in Uustalu-Vene, Sprunger-Jacobs).
2. Stochastic streams $(\mathbf{Kl}(\mathcal{D}), \times)$ are controlled stochastic processes (classical in the literature).
3. Kleisli streams of strong monads. Freyd categories in general.

Extra: implementing signal flow graphs and dataflow programs.

Part 0: Coalgebra

Coalgebra

We are going to be solving functor fixpoints, $FX \cong X$. Final coalgebras give canonical solutions to these equations. Initial algebras work too, but they will be less interesting.

Definition. A coalgebra morphism $f : (X, \alpha : X \to FX) \to (Y, \beta : Y \to FY)$ is a morphism $f : X \to Y$ making the following diagram commute.

$$\begin{array}{ccc} X & \xrightarrow{f} & Y \\ {\scriptstyle\alpha}\downarrow & = & \downarrow{\scriptstyle\beta} \\ FX & \xrightarrow[Ff]{} & FY \end{array}$$

Theorem (Lambek). If the final coalgebra exists, it is a final fixpoint.

Coalgebra

How to construct final coalgebras?

Theorem (Adamek). If the following limit exists and is preserved by F, it is the final coalgebra.

$$\lim_{n \in \mathbb{N}} F^{\text{on}}(1) = \lim_{n \in \mathbb{N}} \left(1 \xleftarrow{!} F1 \xleftarrow{F!} FF1 \xleftarrow{FF!} FFF1 \leftarrow \cdots\right).$$

That is, to compute a fixpoint, repeatedly apply F and it will converge. Hopefully.

Part 1: Intensional Streams

Streams and Stream Functions

"A stream of types $\mathbb{A} = (A_0, A_1, A_2, \ldots)$ is an element of A_0 together with a stream of types $\mathbb{A}^+ = (A_1, A_2, A_3, \ldots)$."

$$S(\mathbb{A}) \cong A_0 \times S(\mathbb{A}^+)$$

Here, $\mathbb{A} \in [\mathbb{N}, \mathrm{Set}]$. Streams should be defined by a functor $S : [\mathbb{N}, \mathrm{Set}] \to \mathrm{Set}$.
We are defining the final coalgebra of $\Phi : [[\mathbb{N}, \mathrm{Set}], \mathrm{Set}] \to [[\mathbb{N}, \mathrm{Set}], \mathrm{Set}]$, given by

$$\Phi(S)(\mathbb{A}) \cong A_0 \times S(\mathbb{A}^+).$$

By Adamek's Theorem, the candidate solution is

$$\lim_{n \in \mathbb{N}} (1 \| \leftarrow \Phi 1 \| \leftarrow \cdots \leftarrow \Phi^n 1 \| \leftarrow \cdots)(\mathbb{A}) =$$

$$\lim_{n \in \mathbb{N}} (1 \leftarrow A_0 \leftarrow A_0 \times A_1 \leftarrow A_0 \times A_1 \times A_2 \leftarrow \cdots) = \prod_{n=0}^{\infty} A_n ;$$

and it is a solution, $A_0 \times \prod_{n=1}^{\infty} A_n \cong \prod_{n=0}^{\infty} A_n$.

Streams and Stream Functions

"A stream function from $X=(X_0, X_1, X_2, \ldots)$ to $Y=(Y_0, Y_1, Y_2, \ldots)$ is a function $X_0 \to Y_0$ communicating along a memory channel M with a stream function from $X^+=(X_1, X_2, X_3, \ldots)$ to $Y^+=(Y_1, Y_2, Y_3, \ldots)$."

$$T(X;Y) = \sum_{M \in \mathrm{SET}} \hom(X_0, M \times Y_0) \times T(M \times X_1, X_2, X_3, \ldots; Y_1, Y_2, Y_3, \ldots).$$

Here, $T\colon [\mathbb{N}, \mathrm{SET}]^{\mathrm{op}} \times [\mathbb{N}, \mathrm{SET}] \to \mathrm{SET}$. We are defining the final coalgebra of an endofunctor $\Phi\colon [[\mathbb{N}, \mathrm{SET}]^{\mathrm{op}} \times [\mathbb{N}, \mathrm{SET}], \mathrm{SET}] \to [[\mathbb{N}, \mathrm{SET}]^{\mathrm{op}} \times [\mathbb{N}, \mathrm{SET}], \mathrm{SET}]$ given by

$$\Phi(T)(X,Y) = \sum_{M \in \mathrm{SET}} \hom(X_0, M \times Y_0) \times T(M \times X_1, X_2, X_3, \ldots; Y_1, Y_2, Y_3, \ldots).$$

Monoidal Streams

"A monoidal stream from $X=(X_0, X_1, X_2, \dots)$ to $Y=(Y_0, Y_1, Y_2, \dots)$ is a morphism $X_0 \to Y_0$ communicating along a memory channel M with a monoidal stream from $X^+=(X_1, X_2, X_3, \dots)$ to $Y^+=(Y_1, Y_2, Y_3, \dots)$."

$$T(X;Y) = \sum_{M \in \mathrm{SET}} \hom(X_0, M \otimes Y_0) \times T(M \otimes X_1, X_2, X_3, \dots; Y_1, Y_2, Y_3, \dots).$$

Here, $T : [[\mathbb{N}, \mathrm{SET}]^{\mathrm{op}} \times [\mathbb{N}, \mathrm{SET}]] \to \mathrm{SET}$. We are defining the final coalgebra of an endofunctor $\Phi : [[\mathbb{N}, \mathrm{SET}]^{\mathrm{op}} \times [\mathbb{N}, \mathrm{SET}], \mathrm{SET}] \to [[\mathbb{N}, \mathrm{SET}]^{\mathrm{op}} \times [\mathbb{N}, \mathrm{SET}], \mathrm{SET}]$ given by

$$\Phi(T)(X,Y) = \sum_{M \in \mathrm{SET}} \hom(X_0, M \otimes Y_0) \times T(M \otimes X_1, X_2, X_3, \dots; Y_1, Y_2, Y_3, \dots).$$

Monoidal Streams

$$\phi(T)(X,Y) = \sum_{M \in \mathrm{SET}} \hom(X_0, M \otimes Y_0) \times T(M \otimes X_1, X_2, X_3, \ldots; Y_1, Y_2, Y_3, \ldots).$$

By Adamek's Theorem, the candidate solution is

$$\lim_{n \in \mathbb{N}} (\mathbb{1} \leftarrow \phi\mathbb{1} \leftarrow \phi^2\mathbb{1} \leftarrow \cdots)(X,Y) =$$

$$\lim_{n \in \mathbb{N}} \left(1 \leftarrow \sum_{M_0} \hom(X_0, M_0 \otimes Y_0) \leftarrow \sum_{M_0, M_1} \hom(X_0, M_0 \otimes Y_0) \times \hom(M_0 \otimes X_1, M_1 \otimes Y_1) \leftarrow \cdots\right) =$$

$$\lim_{n \in \mathbb{N}} \sum_{M_0, \ldots, M_n} \prod_{i \in \mathbb{N}}^{n} \hom(M_{i-1} \otimes X_i, M_i \otimes Y_i) \cong \sum_{M : [\mathbb{N}, \mathbb{G}]} \prod_{n \in \mathbb{N}} \hom(M_{n-1} \otimes X_n, M_n \otimes Y_n).$$

$M_{-1} := I$

And it is indeed a solution, products distribute over coproducts.

$$\sum_{M \in \mathrm{SET}} \hom(X_0, M \otimes Y_0) \times \sum_{M : [\mathbb{N}, \mathbb{G}]} \prod_{n \in \mathbb{N}} \hom(M_{n-1} \otimes X_n, M_n \otimes Y_n) =$$

$$\sum_{M : [\mathbb{N}, \mathbb{G}]} \prod_{n \in \mathbb{N}} \hom(M_{n-1} \otimes X_n, M_n \otimes Y_n).$$

Monoidal Streams

Definition (DLdFR). An (intensional) monoidal stream $\mathbb{X} \to \mathbb{Y}$ is a family of objects $M_0, M_1, M_2, \ldots$ and a family of morphisms $f_n : M_{n-1} \otimes X_n \to M_n \otimes Y_n$.

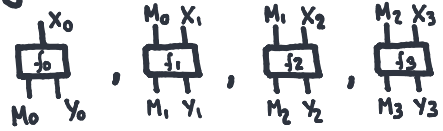
$$T(\mathbb{X}, \mathbb{Y}) = \sum_{M : [N, \mathcal{C}]} \prod_{n \in \mathbb{N}} \hom(M_{n-1} \otimes X_n, M_n \otimes Y_n).$$

Theorem (DLdFR). The set of monoidal streams, depending on inputs and outputs, is the terminal fixpoint of

$$T(\mathbb{X}, \mathbb{Y}) = \sum_{M \in \mathrm{SET}} \hom(X_0, M \otimes Y_0) \times T(M \otimes X_1, X_2, X_3, \ldots; Y_1, Y_2, Y_3, \ldots).$$

Monoidal Streams

How to interpret a monoidal stream? In diagrams,

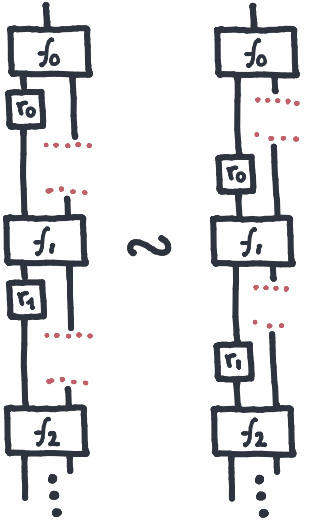
$(f_n : M_{n-1} \otimes X_n \to M_n \otimes Y_n)$ is $\{$  $\}$.

and yields the following "open diagram".

Extensional Streams

Monoidal streams are too explicit. Having memories $M_n = A_n \otimes (B_n \otimes C_n)$ is different from having memories $M_n = (A_n \otimes B_n) \otimes C_n$. This makes them fail to form a category.

Intensionally different streams are extensionally equivalent.



DEFINITION. An (extensional) monoidal stream is an equivalence class of intensional streams under the minimal equivalence relation containing $(\sim)$.

$$\sum_{M:[\mathbb{N},\mathbb{C}]} \prod_{n \in \mathbb{N}} \hom(M_{n-1} \otimes X_n, M_n \otimes Y_n) \Big/ \langle \sim \rangle$$

$$= \int^{M:[\mathbb{N},\mathbb{C}]} \prod_{n \in \mathbb{N}} \hom(M_{n-1} \otimes X_n, M_n \otimes Y_n)$$

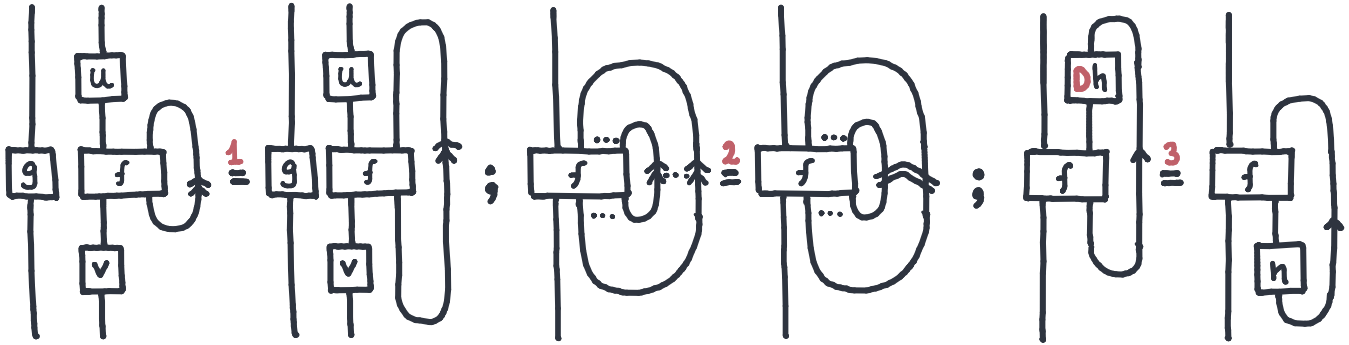
I like coends, but you may not; so let me justify them.

Part 2 : Extensional Streams

Feedback

Feedback monoidal categories (Katis-Sabadini-Walters) axiomatize signal flow graphs using a guarded feedback operator $\mathrm{fbk} : \hom(DS \otimes A, S \otimes B) \to \hom(A, B)$.

Axioms.



Feedback

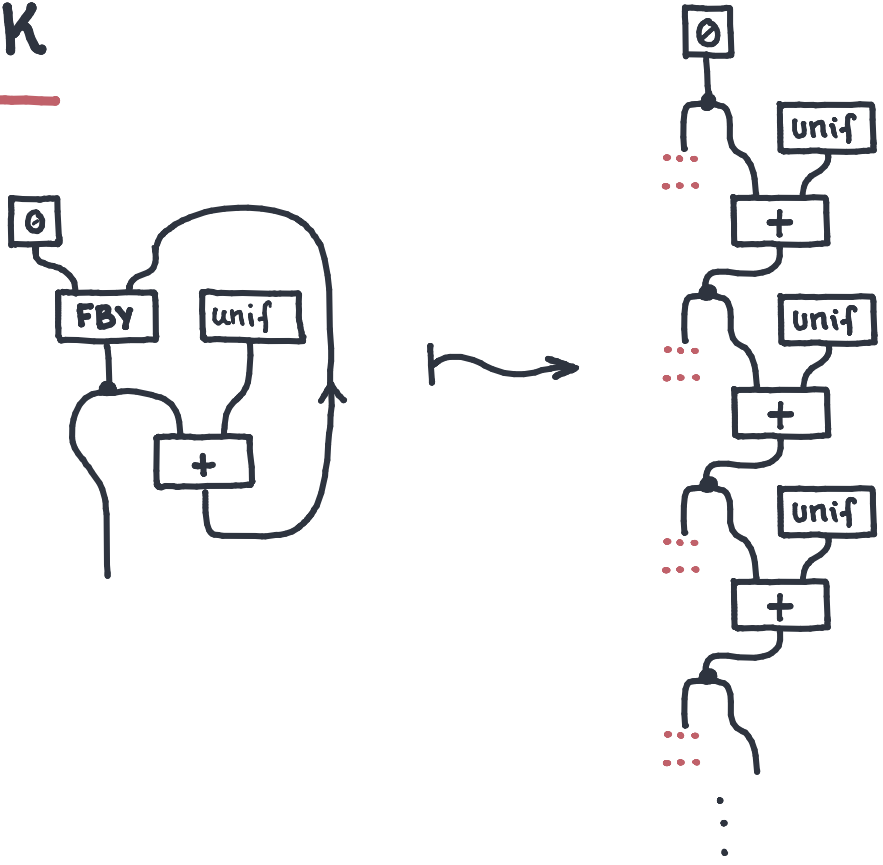
The category $[\mathbb{N}, \mathbb{C}]$ already has a "delay" functor $\partial : [\mathbb{N}, \mathbb{C}] \to [\mathbb{N}, \mathbb{C}]$, defined by $\partial(A_0, A_1, A_2, \ldots) = (I, A_0, A_1, \ldots)$.

If you want to construct a category of streams, you surely want the feedback axioms to hold. But then, ...

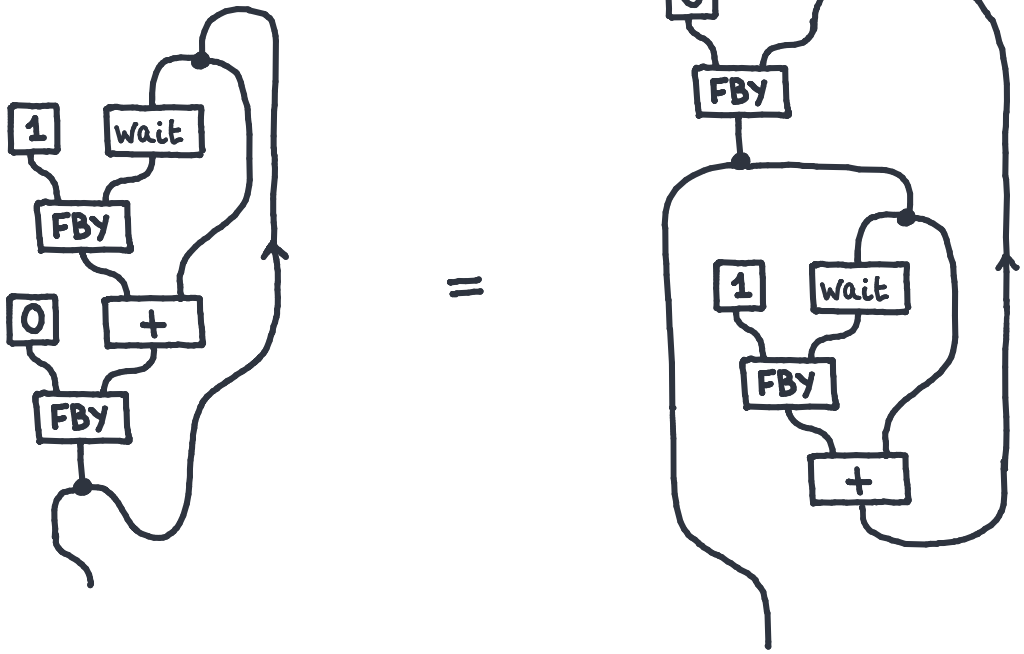
> **THEOREM** (DLdFR). Extensional streams are the morphisms of the free category with feedback over $\partial : [\mathbb{N}, \mathbb{C}] \to [\mathbb{N}, \mathbb{C}]$.

... so, you have already simulated extensional streams.

Feedback



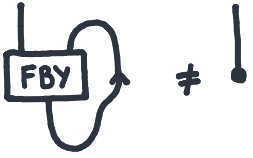
Extensional Streams

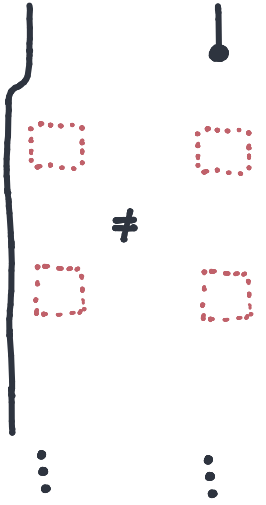


Observational Equivalence

So, we want, at least, extensional equivalence. Can we refine it?

Saving to memory without outputting is, observationally, the same as discarding. However, no amount of sliding will help us equating these two.





Part 3 : Observational Streams

Observational Equivalence

So, we want, at least, extensional equivalence. Can we refine it?

DEFINITION. The nth-truncation of an extensional stream $\langle f_n \rangle$ is the first n components, up to any continuation.

Two streams are observationally equal if their nth truncations are equal. For instance,

This is not only a reasonable-sounding rule. This makes observational streams a canonical fixpoint.

Observational Streams

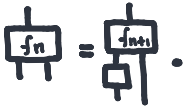
Intensional streams were the canonical fixpoint of the equation

$$T(X,Y) \cong \sum_{M \in \mathbb{C}} \hom(X_0, M \otimes Y_0) \times T(M \otimes X_1, X_2, X_3, \ldots; Y_1, Y_2, Y_3, \ldots).$$

THEOREM (DLdFR). Observational streams, in reasonably well-behaved categories (productive) are the canonical fixpoint of the equation

$$Q(X,Y) \cong \int^{M : \mathbb{C}} \hom(X_0, M \otimes Y_0) \times Q(M \otimes X_1, X_2, X_3, \ldots; Y_1, Y_2, Y_3, \ldots).$$

Why not in all categories? You could craft a category where there is no way to go from X_0 to Y_0 without knowing the future!

I.e. there are infinitely descending chains for a Loebner-like order 

.

Observational Streams

Why not in all categories? You could craft a category where there is no way to go from X_0 to Y_0 without knowing the future! Adamek could fail.

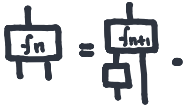
$$\int^{M:\mathbb{C}} \hom(X_0, M \otimes Y_0) \times \lim^{n \in \mathbb{N}} \int^{M_1,\ldots,M_n} \prod_{i=1}^{n} \hom(M_{i-1} \otimes X_i, M_i \otimes Y_i)$$

$\cong$ ✓ fine

$$\int^{M:\mathbb{C}} \lim^{n \in \mathbb{N}} \int^{M_1,\ldots,M_n} \hom(X_0, M \otimes Y_0) \times \prod_{i=1}^{n} \hom(M_{i-1} \otimes X_i, M_i \otimes Y_i)$$

$\cong$ ✗ only discrete coproducts commute with connected limits

$$\lim^{n \in \mathbb{N}} \int^{M:\mathbb{C}} \int^{M_1,\ldots,M_n} \hom(X_0, M \otimes Y_0) \times \prod_{i=1}^{n} \hom(M_{i-1} \otimes X_i, M_i \otimes Y_i)$$

I.e. there are infinitely descending chains for a Loebner-like order 

.

Nothing to worry in semi cartesian, compact closed and freely generated monoidals.

Part 4 : Examples

Causal Stream Functions

Theorem (DLdFR). Monoidal streams in a cartesian category are causal stream functions.

$$\begin{aligned} Q(X,Y) &\cong \int^{M:\mathbb{C}} \hom(X_0, M \times Y_0) \times Q(M \times X_1, X_2, X_3, \ldots; Y_1, Y_2, Y_3, \ldots) \\ &\cong \int^{M:\mathbb{C}} \hom(X_0, M) \times \hom(X_0, Y_0) \times Q(M \times X_1, X_2, X_3, \ldots; Y_1, Y_2, Y_3, \ldots) \\ &\cong \hom(X_0, Y_0) \times Q(X_0 \times X_1, X_2, X_3, \ldots; Y_1, Y_2, Y_3, \ldots) \end{aligned}$$

By Adamek's Theorem, we can compute the final coalgebra to be

$$Q(X,Y) \cong \prod_{n=0}^{\infty} \hom(X_0 \times \cdots \times X_n, Y_n).$$

Stochastic Processes

Definition. A controlled stochastic process $f : \mathbb{X} \to \mathbb{Y}$ is a family of stochastic functions $f_n : X_0 \times \cdots \times X_n \to D(Y_0 \times \cdots \times Y_n)$ satisfying the following property.

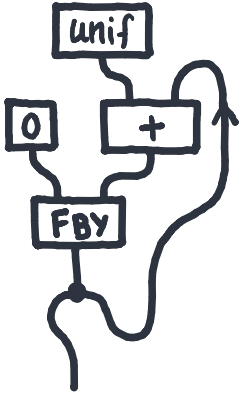
$$\begin{array}{ccc} X_0 \times \cdots \times X_{n+1} & \xrightarrow{f_{n+1}} & D(Y_0 \times \cdots \times Y_{n+1}) \\ {\scriptstyle n}\downarrow & & \downarrow{\scriptstyle D_n} \\ X_0 \times \cdots \times X_n & \xrightarrow{f_n} & D(Y_0 \times \cdots \times Y_n). \end{array}$$

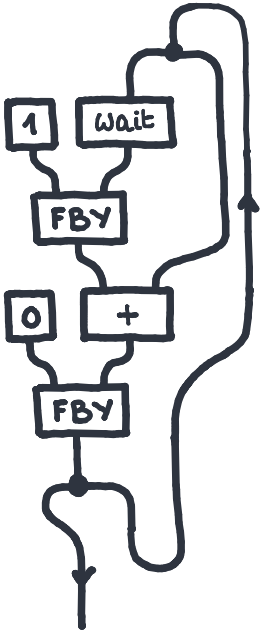
Causality: the future X_{n+1} should not influence the past $Y_0, \ldots, Y_n$.

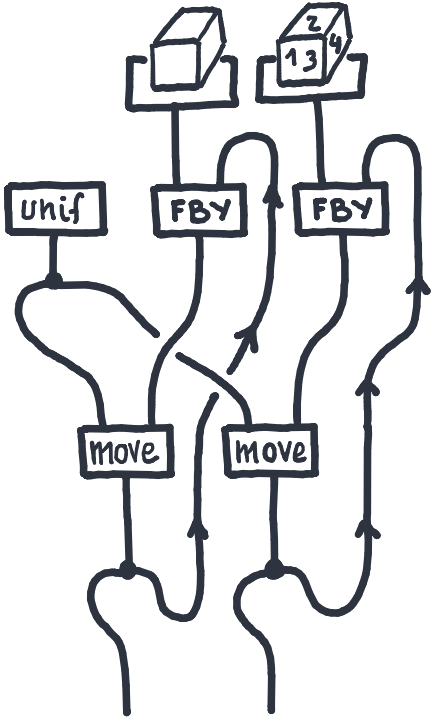
Theorem (DLdFR). Monoidal streams over $\mathrm{Kl}(D)$ coincide with controlled stochastic processes.

Proof. Non trivial. Somehow, the causality condition means that the family can be written *uniquely* as a monoidal stream.

Examples







Part 5 : Coinduction

Coinductive Monoidal Streams

Reasoning with monoidal streams is easy: they are a final coalgebra.

$$Q(X,Y) \cong \int^{M:\mathbb{C}} \hom(X_0, M \otimes Y_0) \times Q(M \otimes X_1, X_2, X_3, \ldots; Y_1, Y_2, Y_3, \ldots).$$

DEFINITION (DLdFR). A monoidal stream $f \in \mathrm{Stream}(X_0, X_1 \ldots; Y_0, Y_1, \ldots)$ is

- a memory $M(f) \in \mathbb{C}$
- a $\mathrm{now}(f) : X_0 \to M(f) \otimes Y_0$,
- and a $\mathrm{later}(f) \in \mathrm{Stream}(M \otimes X_1, X_2 \ldots; Y_1, Y_2, \ldots)$.

Quotiented by $f \approx g$, meaning

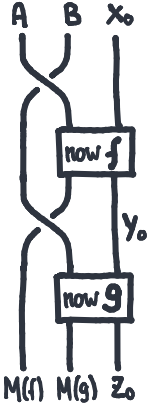
- the existence of $r : M(f) \to M(g)$,
- such that $\mathrm{now}(f); (r \otimes \mathrm{id}) = \mathrm{now}(g)$,
- and such that $\mathrm{later}(f) \approx r \cdot \mathrm{later}(g)$.

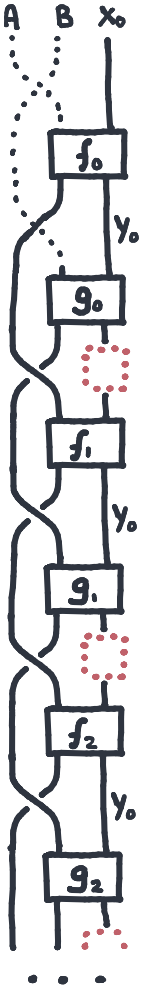
Most of the time, we reason with streams of the form
$(A \otimes X_0, X_1, \ldots) \to (Y_0, Y_1, \ldots)$; "streams with memory".
This strenghtens the coinduction hypothesis. Later, we set $A = I$.

Coinductive Monoidal Streams

Sequential Composition with memory of $f \in \mathrm{Stream}(A \cdot X, Y)$ and $g \in \mathrm{Stream}(B \cdot Y, Z)$ is written as $(f \overset{A}{\triangleright} g^{B}) \in \mathrm{Stream}(A \otimes B \cdot X, Z)$, and defined by

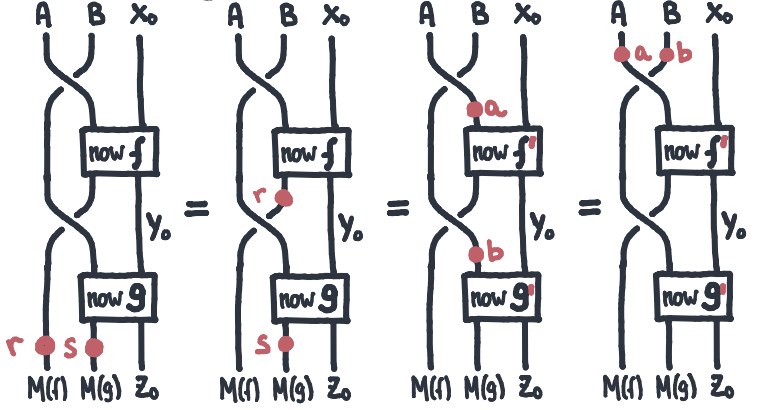
- $M(f \overset{A}{\triangleright} g^{B}) = M(f) \otimes M(g)$;
- $\mathrm{later}(f \overset{A}{\triangleright} g^{B}) = \mathrm{later}(f)^{M(f)} \overset{B}{\triangleright} \mathrm{later}(g)^{M(g)}$, by coinduction;
- $\mathrm{now}(f \overset{A}{\triangleright} g^{B}) =$ •





Coinductive Monoidal Streams

Sequential Composition is well-defined. Given generators $f \overset{r}{\approx} a \cdot f'$ and $g \overset{s}{\approx} b \cdot g'$, we can show that $(f^A \circ g^B) \overset{r \otimes s}{\approx} (a \otimes b) \cdot (f'^{A'} \circ g'^{B'})$.



By coinduction,

$$\mathrm{later}(f)^{M(f)} \circ \mathrm{later}(g)^{M(g)} \approx (r \otimes s) \cdot (\mathrm{later}(f')^{M(f')} \circ \mathrm{later}(g')^{M(g')}).$$

using $\mathrm{later}(f) \approx r \cdot \mathrm{later}(f')$, $\mathrm{later}(g) \approx s \cdot \mathrm{later}(g')$.

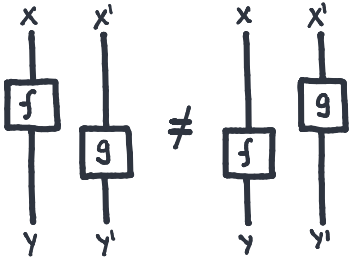
In particular, when $a = \mathrm{id}_A$, $b = \mathrm{id}_B$, we have

$$(f \approx f') \wedge (g \approx g') \Rightarrow f^A \circ g^B \approx f'^A \circ g'^B.$$

Part 6 : Extra

Premonoidal Categories

A sym. premonoidal category $(\mathbb{C}, \boxtimes, I)$ is a sym. monoidal category without the interchange law.



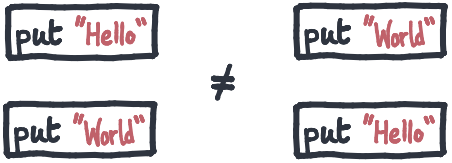
They usually have a family of "pure" morphisms that do satisfy interchange, forming a monoidal $\mathbb{V}$.

$$\mathbb{V} \longrightarrow \mathbb{C}$$

Pure — id-on-objects functor → Effectful

This is called a Freyd category.

Main Example. The Kleisli category of a strong monad, $\mathrm{Set} \to \mathrm{Kl}(T)$.



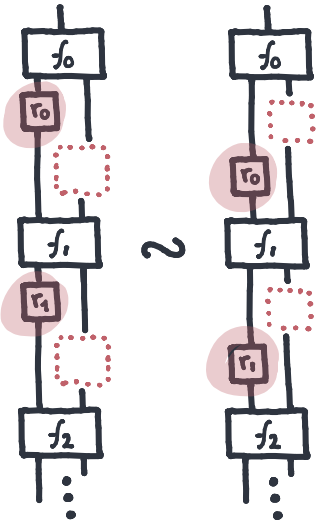
Premonoidal Categories (Work in progress)

Definition. The set of premonoidal streams in a Freyd category $\mathbb{V} \to \mathbb{C}$ is the final fixpoint of

$$Q(X,Y) \cong \int^{M:\mathbb{V}} \hom_{\mathbb{C}}(X_0, M \otimes Y_0) \times Q(M \otimes X_1, X_2, X_3, \ldots; Y_1, Y_2, Y_3, \ldots).$$

Only pure morphisms should slide.

Theorem. If $\mathbb{V}$ is cartesian, then this final coalgebra is constructed by observational streams.



Implementation

Obvious candidate: Haskell Arrows give two notations for Set-based Freyd categories.

1. Arrow notation ($\ggg$, $***$) mimics string diagrams. We use it on the paper for this reason.
    - ArrowLoop is for traced categories, in theory; it works for feedback.
2. Do-notation ($\bullet \leftarrow f \prec \bullet$) gives automatic coherence. It is the internal language of Freyd categories but it is inexplicably restricted to Set-based ones. Can we have acausal notation for feedback?

I like coinduction in Agda. However, coend quotients are painful. Coherence is difficult to prove.

Expressivity

Orthogonal to the rest of the paper. What is the expressivity of the feedback syntax over some generators? The $St(\cdot)$ construction answers this.

- κ-Linear functions: hidden multivariable linear recurrence equations.

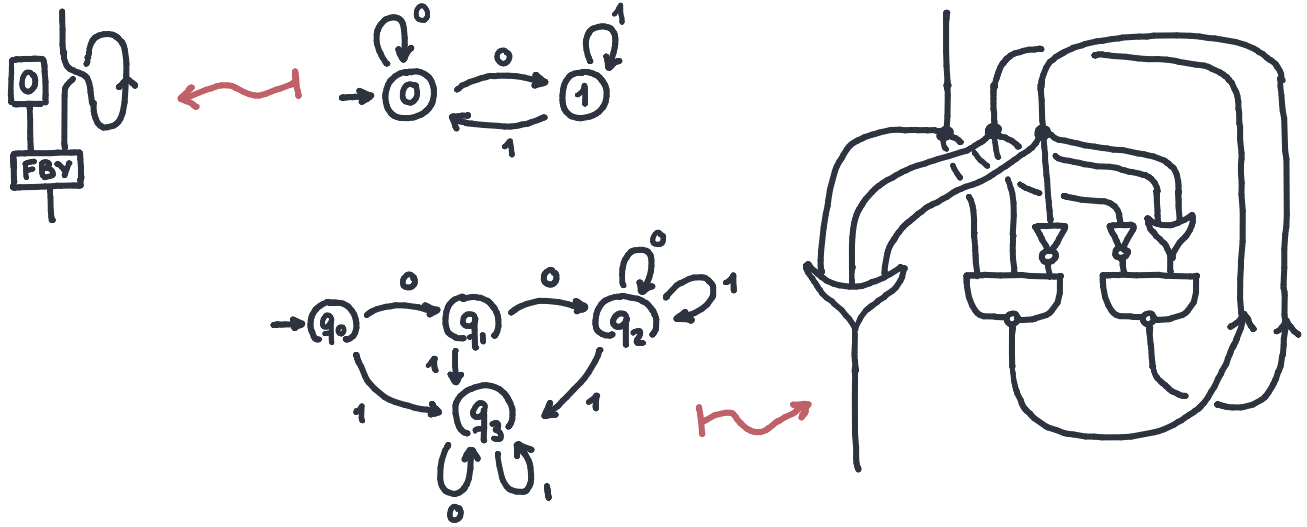
$$\begin{pmatrix} s_1 \\ \vdots \\ s_n \end{pmatrix}_{t+1} = \begin{pmatrix} \lambda_{11} & \cdots & \lambda_{1n} & \mu_{11} & \cdots & \mu_{1\kappa} \\ \vdots & & \vdots & \vdots & & \vdots \\ \lambda_{n1} & \cdots & \lambda_{nn} & \mu_{1n} & \cdots & \mu_{n\kappa} \end{pmatrix} \begin{pmatrix} s_1 \\ \vdots \\ s_n \\ a_1 \\ \vdots \\ a_\kappa \end{pmatrix}_t$$

$$\begin{pmatrix} b_1 \\ \vdots \\ b_n \end{pmatrix}_t = \begin{pmatrix} \varphi_{11} & \cdots & \varphi_{1n} & \psi_{11} & \cdots & \psi_{1\kappa} \\ \vdots & & \vdots & \vdots & & \vdots \\ \varphi_{n1} & \cdots & \varphi_{nn} & \psi_{1n} & \cdots & \psi_{n\kappa} \end{pmatrix} \begin{pmatrix} s_1 \\ \vdots \\ s_n \\ a_1 \\ \vdots \\ a_\kappa \end{pmatrix}_t$$

Expressivity

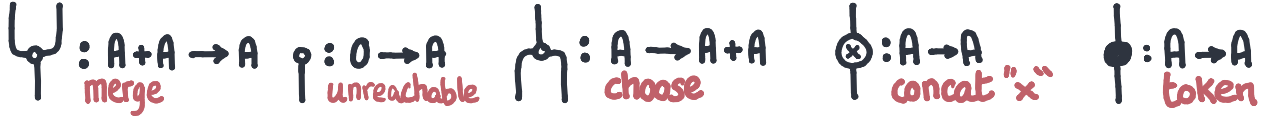
Orthogonal to the rest of the paper. What is the expressivity of the feedback syntax over some generators? The St(•) construction answers this.

- Boolean circuits: controlled deterministic automata (w/ boolean IO).

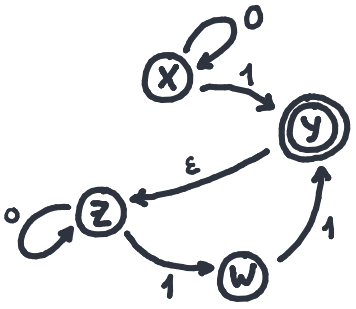


Expressivity

Monoidal streams over total relations can be interpreted as Büchi automata.

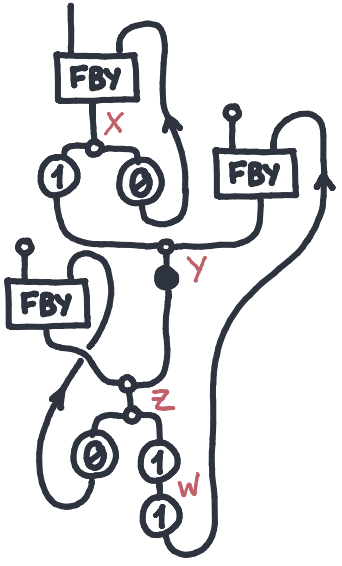


Example: $0^*1(0^*11)^\omega$



Park's equations.

$$X \Rightarrow 0X + 1Y$$
$$Y \Rightarrow Z$$
$$Z \Rightarrow 0Z + 1W$$
$$W \Rightarrow 1Y$$



END

Related Work


Katis-Sabadini-Walters. Categories with feedback, missing delay.


Sprunger-Katsumata, Ghica-Kaye. Finite memory or cartesian streams.


Uustalu-Vene. Cartesian streams, distributive laws for effects?


Hughes-Paterson. ArrowLoop are similar, but for traces.


Carette-De Visme-Perdrix. Syntax for similar streams.


Many others. Categorical data-flow. Functional reactive programming.

Motivation

Lenses can be used to describe a single exchange,

$$\mathsf{Lenses}(A_0,A_1;B_0,B_1) = \int^{M} \hom(A_0, B_0\times M)\times\hom(M\times A_1, B_1) \cong \hom(A_0,B_0)\times\hom(A_0\times A_1, B_1)\,;$$

but "fixpointing" the exchange, we get causal stream functions.

$$\mathsf{Stream}(A;B) = \int^{M} \hom(A_0, B_0\times M)\times \mathsf{Stream}(M\cdot A^+;B^+) = \hom(A_0,B_0)\times\mathsf{Stream}(A_0\cdot A^+;B^+).$$

$$\mathsf{Stream}(A;B) \cong \mathsf{Optic}(\mathsf{Set}, \mathsf{Stream})((A_0,A^+);(B_0,B^+)).$$

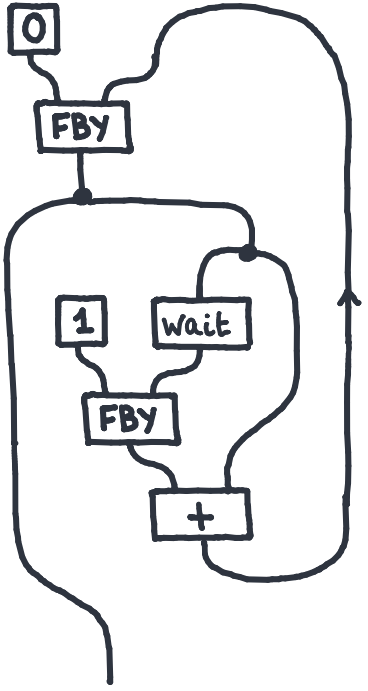
Given the category Stoch of stochastic functions, the category of stochastic processes is the terminal fixpoint of

$$\mathsf{StochProc}(A;B) \cong \mathsf{Optic}(\mathsf{Stoch}, \mathsf{StochProc})((A_0,A^+);(B_0,B^+))$$

Motivation: Dataflow Programming

fib = 0 FBY (fib + 1 FBY WAIT fib)

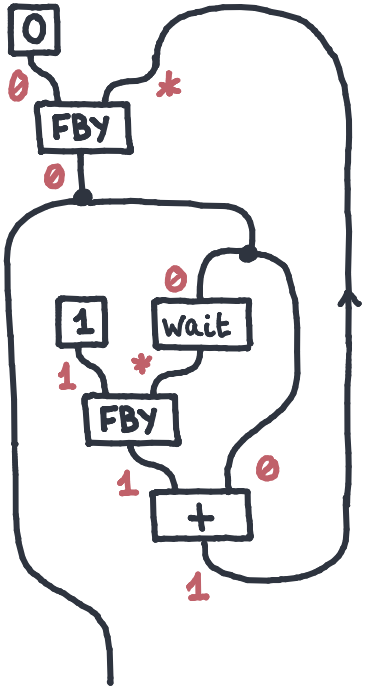
Time.	fib	WAIT(fib)	1 FBY WAIT fib
0			
1			
2			
3			
4			
⋮			



Motivation: Dataflow Programming

$fib = 0 \text{ FBY } (fib + 1 \text{ FBY WAIT } fib)$

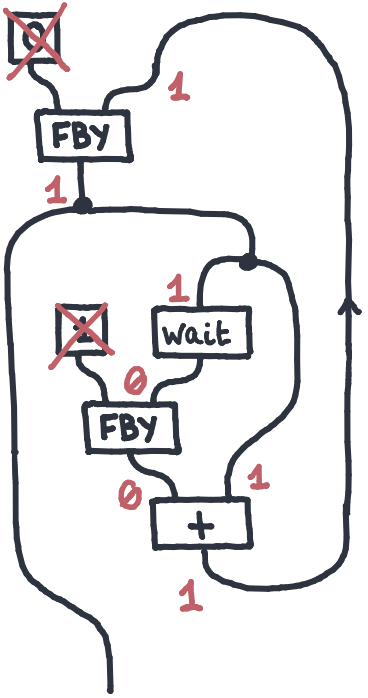
Time.	fib	WAIT(fib)	1 FBY WAIT fib
0	0	*	1
1			
2			
3			
4			
⋮			



Motivation: Dataflow Programming

$fib = 0 \text{ FBY } (fib + 1 \text{ FBY WAIT } fib)$

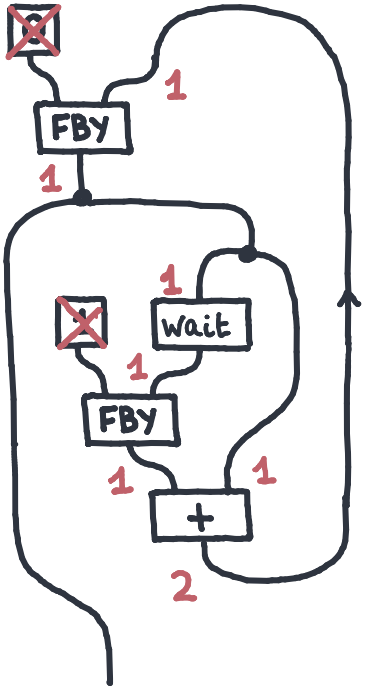
Time.	fib	WAIT(fib)	1 FBY WAIT fib
0	0	*	1
1	1	0	0
2			
3			
4			
⋮			



Motivation: Dataflow Programming

$fib = 0 \text{ FBY } (fib + 1 \text{ FBY WAIT } fib)$

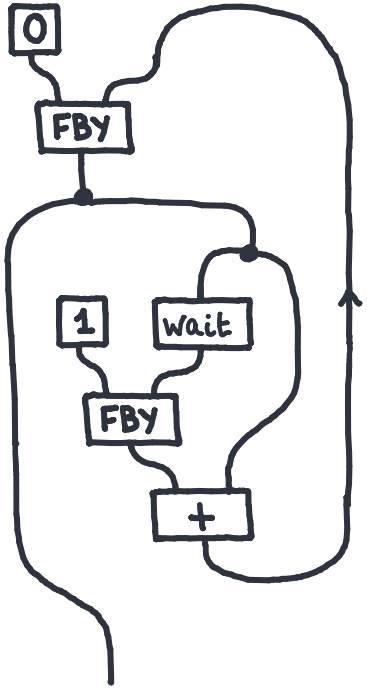
Time.	fib	WAIT(fib)	1 FBY WAIT fib
0	0	*	1
1	1	0	0
2	1	1	1
3			
4			
⋮			



Motivation: Dataflow Programming

$fib = 0 \text{ FBY } (fib + 1 \text{ FBY WAIT } fib)$

Time.	fib	WAIT(fib)	1 FBY WAIT fib
0	0	*	1
1	1	0	0
2	1	1	1
3	2	1	1
4	3	2	2
⋮	⋮	⋮	⋮



Motivation: Dataflow Programming

$$fib = 0 \text{ FBY } (fib + 1 \text{ FBY WAIT } fib)$$

Time.	fib	WAIT(fib)	1 FBY WAIT fib
0	0	*	1
1	1	0	0
2	1	1	1
3	2	1	1
4	3	2	2
⋮	⋮	⋮	⋮

How to ensure the output is well-defined? Delayed types.