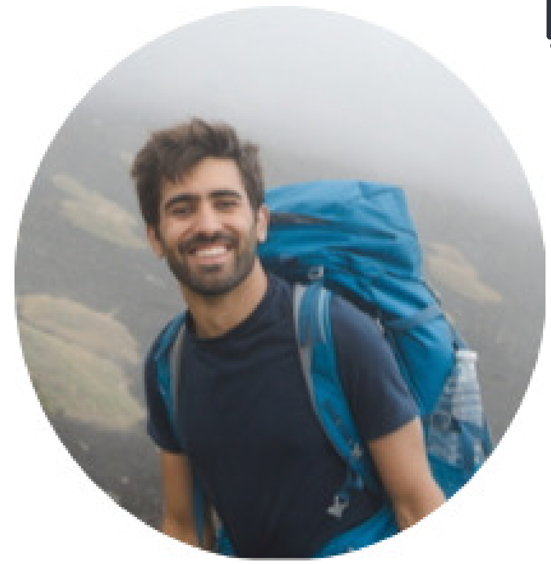
Monoidal Streams for Dataflow Programming

Elena Di Lavore, Giovanni de Felice, Mario Román
Tallinn University of Technology / U. of Oxford

Supported by the EU Estonian IT Academy.

June 20st, 2022 LiCS22

Monoidal Streams for Dataflow Programming



Elena Di Lavore

Tallinn University of Technology.

Giovanni de Felice

University of Oxford + Quantinuum

Mario Román

Tallinn University of Technology.

Part 0: Dataflow Programming

Motivation: Dataflow Programming

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

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

- Elegant recursive dataflow syntax.
- Signal flow, 'trace-like' diagrams.

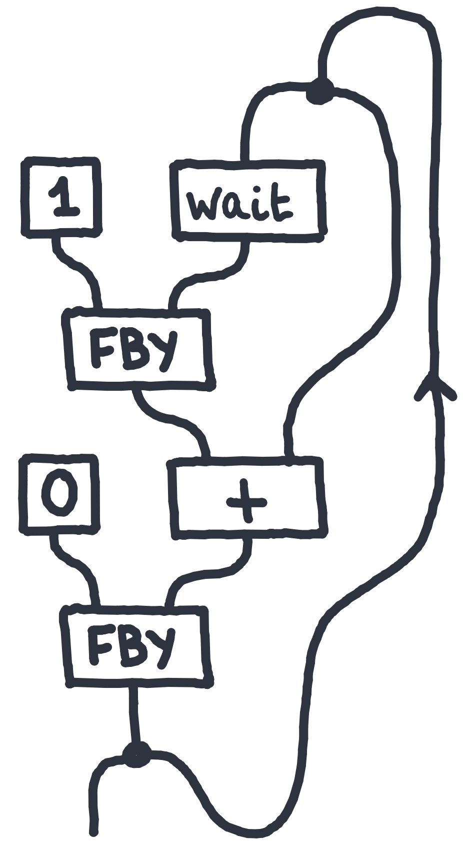


FIG. Signal flow graph.

E.g. LUSTRE, LUCID. Ashcroft, Wadge, 85

Motivation: Dataflow Programming

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

"followed by" "delayed stream"

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

- Elegant recursive dataflow syntax.
- Signal flow, 'trace-like' diagrams.

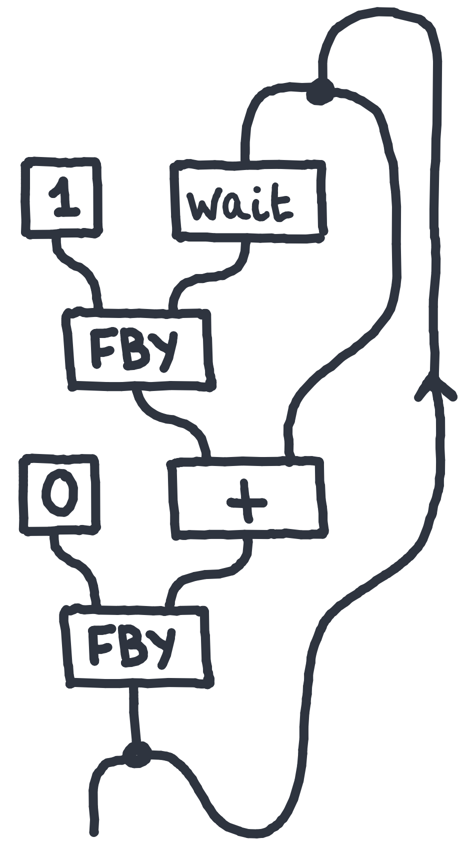


FIG. Signal flow graph.

E.g. Lustre, Lucid. Ashcroft, Wadge, 85

Semantics: Stream Functions

DEFINITION. A causal stream function $f \colon \mathbb{X} \to \mathbb{Y}$ is a family of functions, indexed by natural number time, from all past inputs to the current output.

$$
\begin{aligned}
f_0 &: X_0 \to Y_0 \\
f_1 &: X_0 \times X_1 \to Y_1 \\
f_2 &: X_0 \times X_1 \times X_2 \to Y_2 \\
&\dots
\end{aligned}
$$

PROPERTIES: Sprunger, Katsumata 19

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

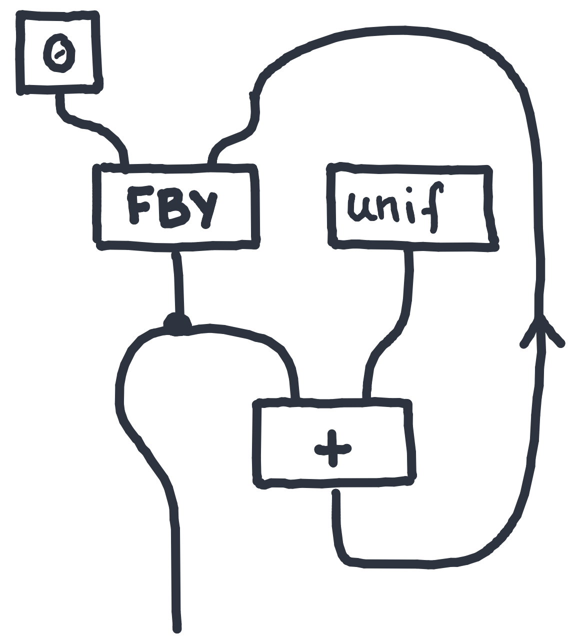
CATEGORICALLY: The cokleisli category of the non-empty list monoidal comonad,

$\mathrm{List}^+ \colon [\mathbb{N}, \mathrm{Set}] \to [\mathbb{N}, \mathrm{Set}]$, $\mathrm{List}^+(\mathbb{X})_n = \prod_{i=0}^{n} X_i$. Uustalu, Vene 08

Other Theories of Processes

What about other theories of processes?

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



E.g. stochastic functions ⇝ stochastic streams.

Given a theory of processes,
we construct a theory of streams
over them.

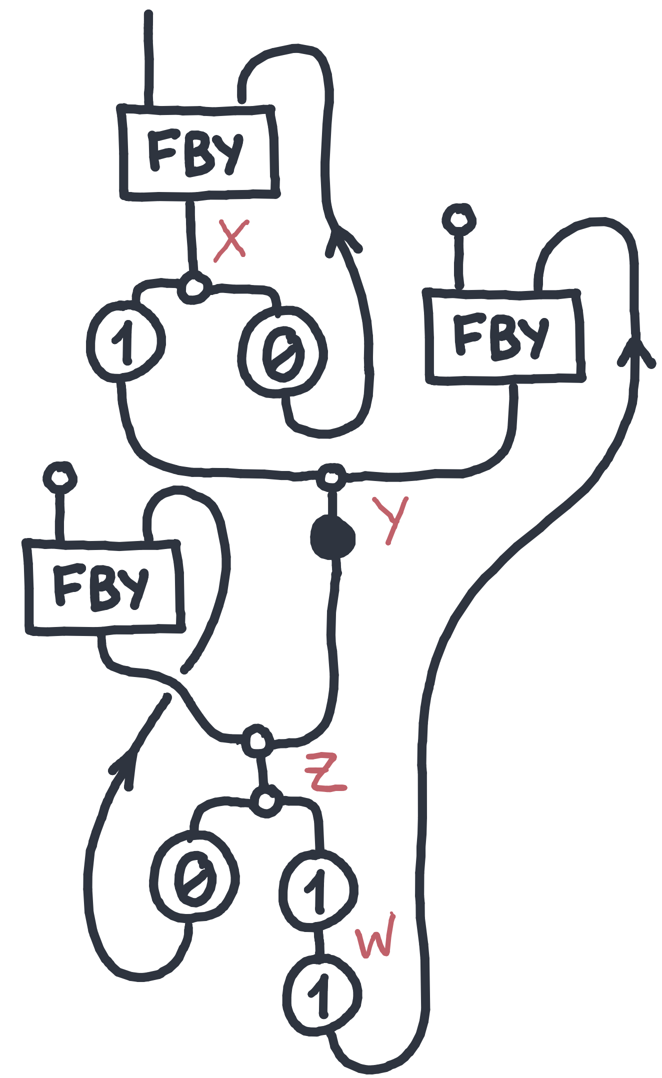
Other Theories of Processes

What about other theories of processes?

$$X = (0 \text{ FBY } X) + (1 \text{ FBY } Y)$$
$$Y = \bullet(Z)$$
$$Z = (0 \text{ FBY } Z) + (1 \text{ FBY } W)$$
$$W = (1 \text{ FBY } Y)$$

E.g. non-deterministic functions $\rightsquigarrow$ non-deterministic streams.

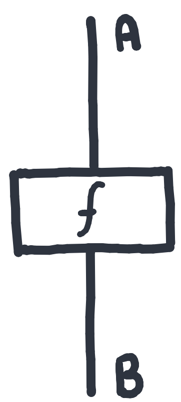
Given a theory of processes,
we construct a theory of streams
over them.



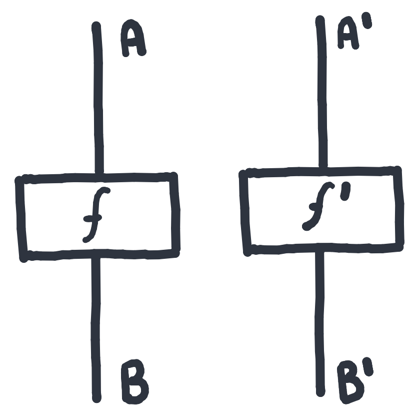
Part 1: Monoidal Categories

Monoidal Categories: Process Theories

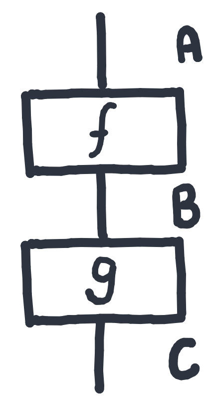
Monoidal categories are an algebra of parallel and sequential composition.
String diagrams are the internal language of monoidal categories.



Process



Parallel composition

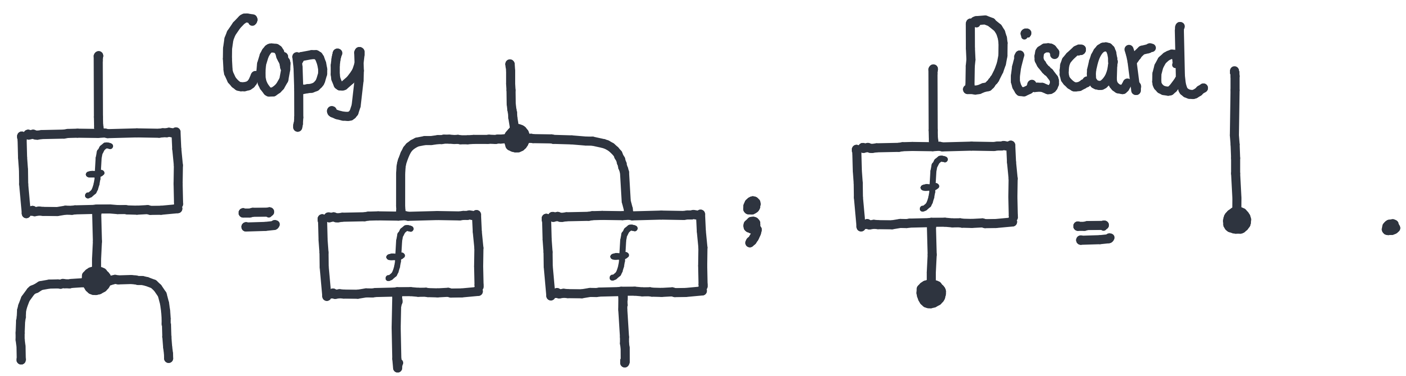


Sequential composition

Bènabou

Monoidal Categories: Cartesian

DEFINITION. Monoidal categories are cartesian when processes can be copied and discarded uniformly.



Deterministic functions	✓	✓
Stochastic functions	✗	✓
Partial functions	✓	✗
Relations	✗	✗

T. Fox

Monoidal Dataflow Programming

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

What about other theories of processes?

$\Downarrow$

What about other, non-cartesian, monoidal categories?

Synopsis

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

- Stream($\mathbb{C}$) has an id-on objs functor from $[\mathbb{N}, \mathbb{C}]$;
- Stream($\mathbb{C}$) has a delay monoidal functor, ∂;
- Stream($\mathbb{C}$) has delayed feedback taking $\partial S \otimes X \to S \otimes Y$ into $X \to Y$;
- Stream($\mathbb{C}$) has a coalgebraic description;
- Stream($\mathbb{C}$) is cartesian when $\mathbb{C}$ is;
- Stream(Set) is the classical causal streams;
- Stream(Stoch) is causal discrete stochastic processes.
- Stream($\mathbb{C}$) is symm. premonoidal, effectful or Freyd when $\mathbb{C}$ is.

Three definitions in terms of universal properties, and three constructions.

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 $(\mathrm{Set}, \times)$ are causal functions (as in Uustalu-Vene, Sprunger-Jacobs).
2. Stochastic streams $(\mathrm{K\ell}(D), \times)$ are controlled stochastic processes (classical in the literature).
3. Kleisli streams of strong monads. Premonoidal categories in general.

(Intensional)

Part 2: Monoidal 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 $\mathbb{A}^+ = (A_1, A_2, A_3, \ldots)$."

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

Streams and Stream Functions

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

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

By Adamek's Theorem, the candidate solution is

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

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

Adámek, 74

Streams and Stream Functions

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

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

By Adamek's Theorem, the candidate solution is

$$\lim_{n \in \mathbb{N}} (1 \leftarrow \hom(X_0, Y_0) \leftarrow \hom(X_0, Y_0) \times \hom(X_0 \times X_1, Y_1) \leftarrow \cdots) = \prod_{n=0}^{\infty} \hom(X_0 \times \cdots \times X_n, Y_n);$$

Sprunger, Katsumata 19

Monoidal Streams

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

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

By Adamek's Theorem, the candidate solution is

$$\lim_{n \in \mathbb{N}} \Big(1 \xleftarrow[n]{} \sum_{M_0} \hom(X_0, M_0 \otimes Y_0) \leftarrow \sum_{M_0 M_1} \hom(X_0, M_0 \otimes Y_0) \times \hom(M_0 \otimes X_1, M_1 \otimes Y_1) \leftarrow \dots \Big) =$$

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

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

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

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

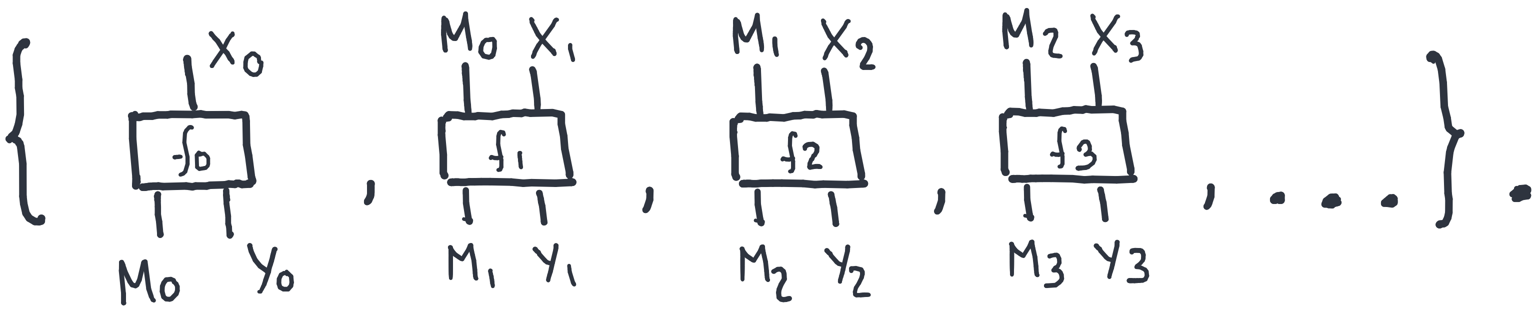
$$\mathrm{T}(\mathbb{X}, \mathbb{Y}) = \sum_{M \in \mathrm{SET}} \hom(X_0, M \otimes Y_0) \times \mathrm{T}(M \otimes X_1, X_2, X_3, \ldots; Y_1, Y_2, Y_3, \ldots).$$

Monoidal Streams

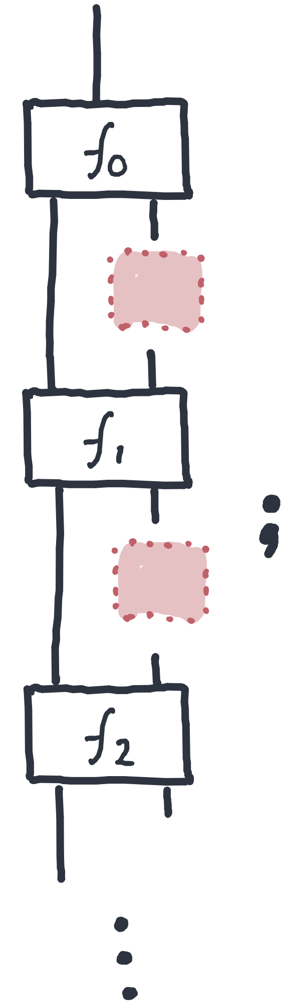
How to interpret a monoidal stream? In diagrams,

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

is

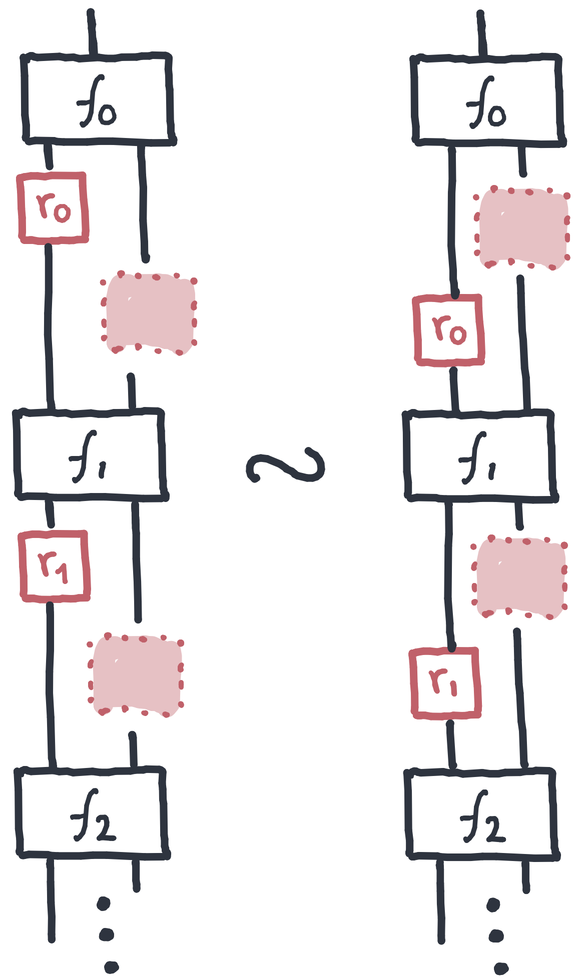


and yields the following "open diagram".



Dinaturality

Monoidal streams are too explicit.



Dinaturality generalizes categorical naturality: it is used to compose profunctors, so it appears implicitly when communicating processes in dataflow programming.

Hildebrandt, Panangaden, Winskel. 97.

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

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

(Coinductive)

Part 3 : Monoidal Streams

Coinductive Monoidal Streams

THEOREM (DLdFR). Monoidal streams (with observational eq.) are the final coalgebra of

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

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

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

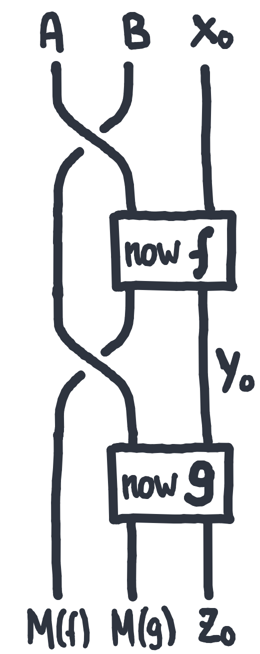
Quotiented by $f \approx g$, meaning

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

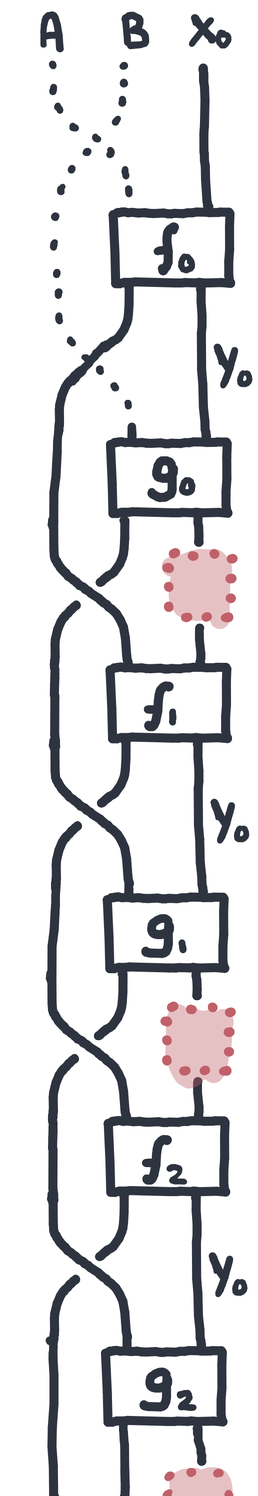
Coinductive Monoidal Streams

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 $(f^A ; g^B) \in \mathrm{Stream}(A \otimes B \cdot X, Z)$, and defined by

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



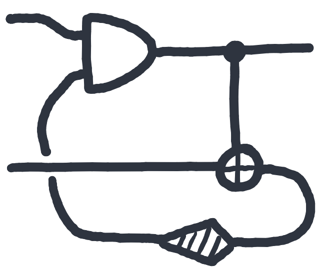
.



Practical coinduction. Rutten Kozen, Silva

FURTHER WORK

Coinduction via string diagrams.



Can we compare with the graphical language here?

Carette, de Visme, Perdrix.

Premonoidal categories.

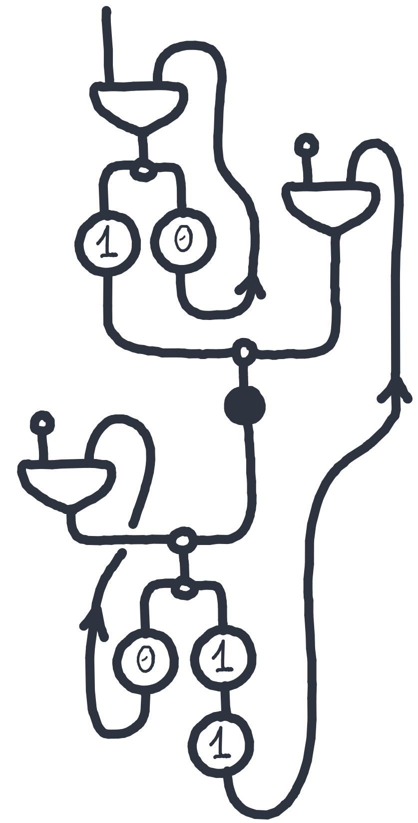
Restricting to central dinaturality.

Power, Robinson

Román. Open Diagrams.

Expressivity of monoidal streams.

Systems of equations for ω-regular languages are monoidal streams over a suitable category of total relations.



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

Park.

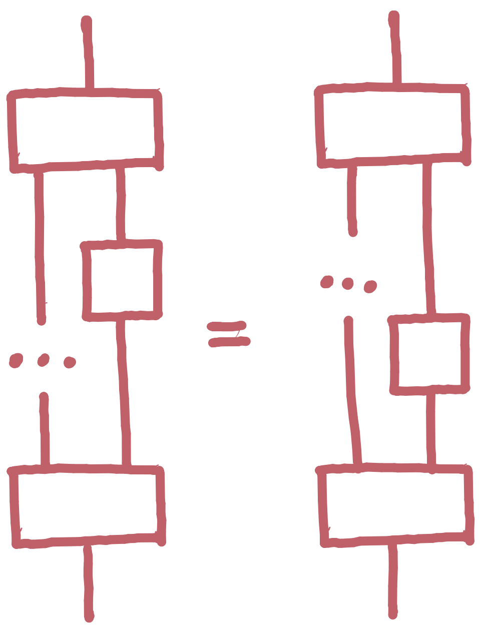
CONCLUSIONS

TAKE HOME

- Given a theory of processes, we construct a theory of streams over them.

  $(\mathbb{C}, \otimes, I) \longrightarrow (\mathbf{Stream}_{\mathbb{C}}, \otimes^{\mathbb{N}}, I)$

- The Key idea is to quotient by dinaturality before using coalgebra.



- We recover classical streams and stochastic processes.
- We can reason coinductively, as we do with streams.
- Implementation and a toy language are direct from the universal properties.

Monoidal Streams for Dataflow Programming

Elena Di Lavore
Tallinn University of Technology.

Giovanni de Felice
University of Oxford + Quantinuum

Mario Román
Tallinn University of Technology.

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 dataflow. Functional reactive programming.
- **Román.** Open diagrams via Coend Calculus.

(Extensional)

Part 3 : Monoidal Streams

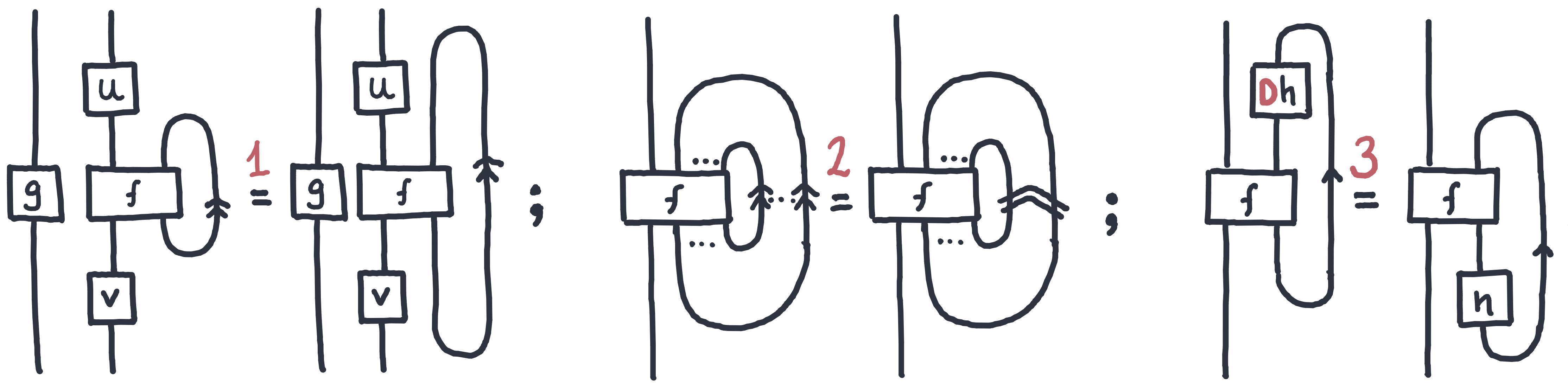
I like dinaturality, but you may not; so let me justify it.

Feedback

Katis, Sabadini, Walters 97
Hoshino, Muroya, Hasuo 14
Bonchi, Sobocinski, Zanasi 14

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

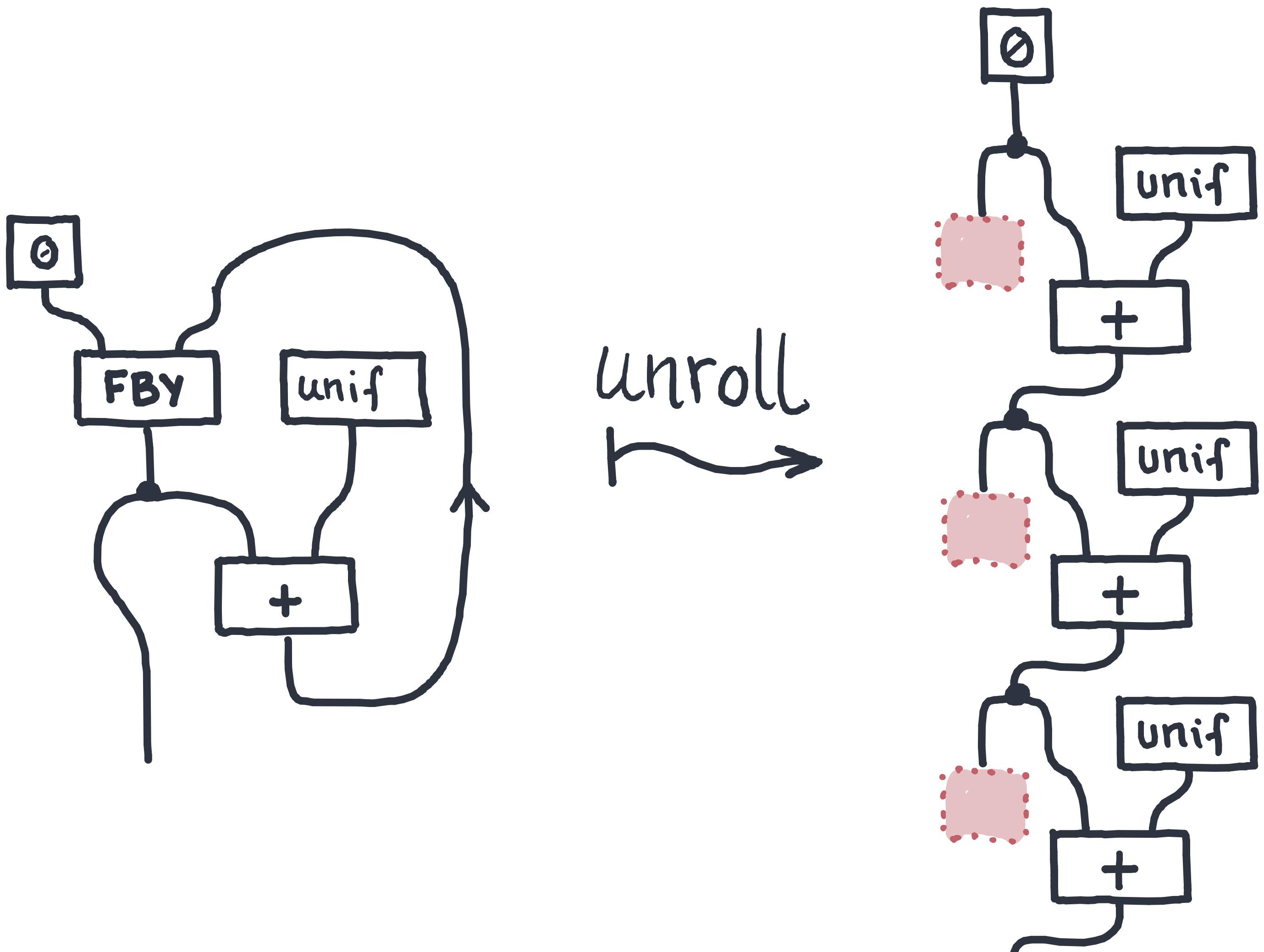
Axioms.



THEOREM (DLdFR). Extensional streams are the morphisms of the free category with feedback over $\partial : [\mathbb{N}, \mathcal{C}] \to [\mathbb{N}, \mathcal{C}]$, $\partial(A_0, A_1, A_2, \ldots) = (I, A_0, A_1, \ldots)$.

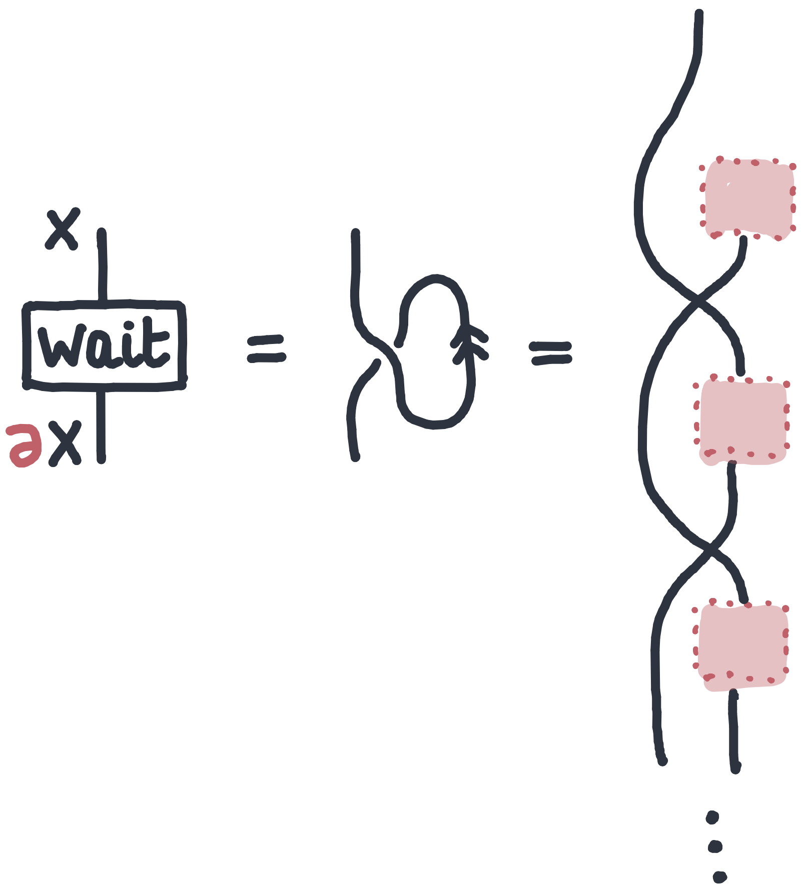
Feedback

Syntax (free category with feedback) to semantics (monoidal streams).

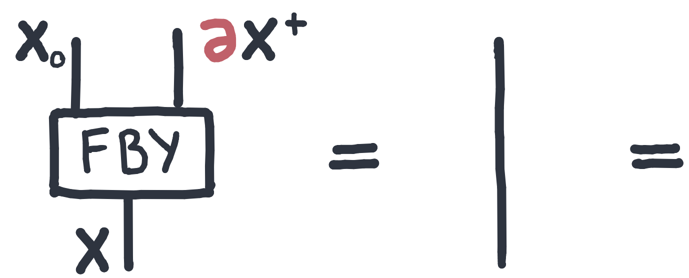


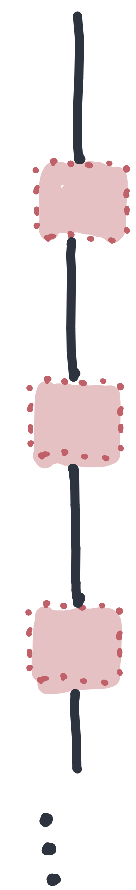
Feedback

Dataflow syntax (FBY, WAIT) has meaning in terms of feedback and [N,C].



;

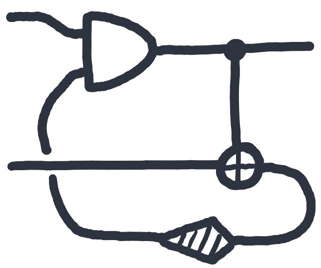




because $X_0 \otimes \partial X^+ = X$.

FURTHER WORK

Carette, de Visme, Perdrix.



Can we compare with the graphical language here?

Delayed Trace.

Kaye, Sprunger, Ghica

Better treatment of the finite memory case.

Cruttwell, Gallagher, Lemay, Pronk

Blute, Cockett, Lemay, Seely

Can we preserve the differential structure from cartesian to monoidal?

Spivak Myers Shapiro, Spivak Smithe Hornischer.

How much can we recover in terms of (continuous) dynamical systems? How much of the coalgebra in terms of polynomial functors?

Power, Robinson

Premonoidal streams are arguably more interesting.

Román. Open Diagrams.

Part 5: Examples

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

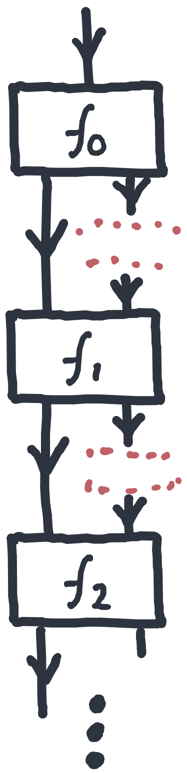
$$\begin{tikzcd}
X_0 \times \cdots \times X_{n+1} \arrow[r, "f_{n+1}"] \arrow[d, "\pi"'] & D(Y_0 \times \cdots \times Y_{n+1}) \arrow[d, "D\pi"] \\
X_0 \times \cdots \times X_n \arrow[r, "f_n"] & D(Y_0 \times \cdots \times Y_n).
\end{tikzcd}$$

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

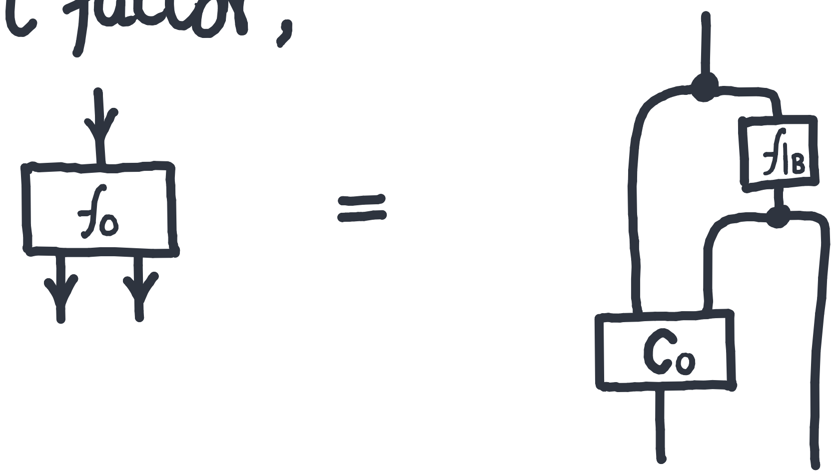
THEOREM (DLdFR). Monoidal streams over $Kl(D)$ coincide with controlled stochastic processes.

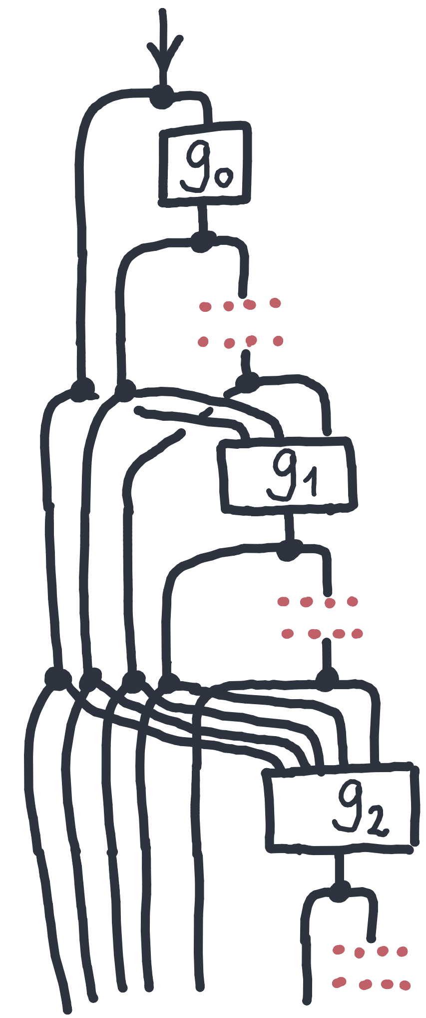
PROOF. Non trivial. Somehow, the causality condition means that the family can be written *uniquely* as a monoidal stream.

Stochastic Monoidal Streams



In Markov categories with conditionals, we can factor,



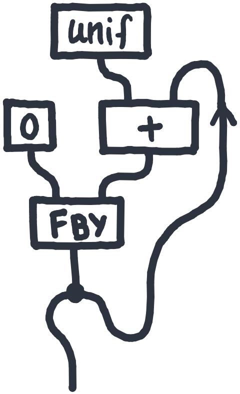


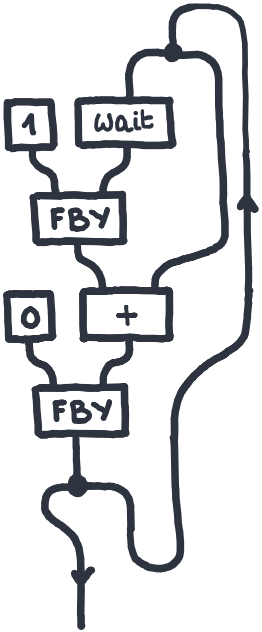
We use this to show that monoidal streams of stochastic functions are the same as controlled stochastic processes.

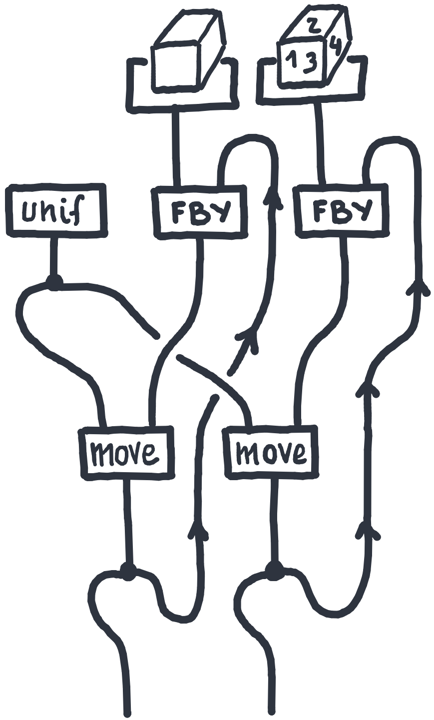
Fritz 19

EXTRA : IMPLEMENTATION

EXAMPLES

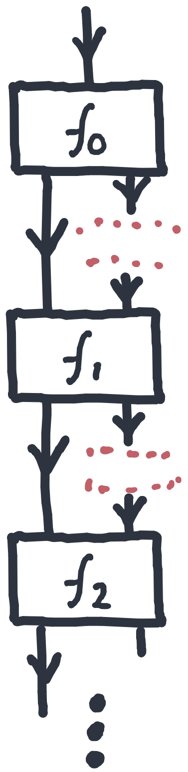




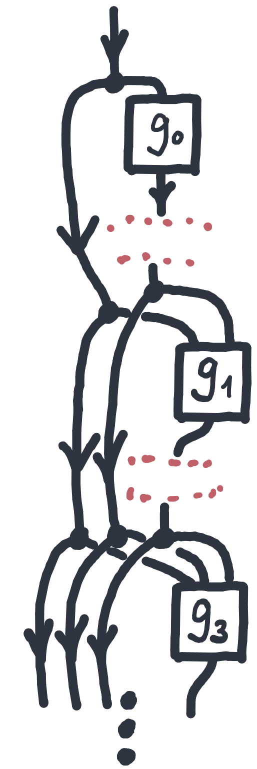


Cartesian Monoidal Streams

THEOREM. In cartesian monoidal categories, monoidal streams are causal stream functions.



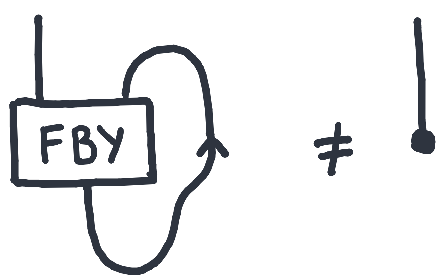
$$\begin{aligned}
&\mathbb{S}(\mathbb{X},\mathbb{Y}) \\
&\cong \int^{M:\mathbb{C}} \hom(X_0, M\times Y_0)\times \mathbb{S}(M\times X_1, X_2, X_3,\dots; Y_1, Y_2, Y_3,\dots) \\
&\cong \int^{M:\mathbb{C}} \hom(X_0, M\times Y_0)\times \mathbb{S}(M\times X_1, X_2, X_3,\dots; Y_1, Y_2, Y_3,\dots) \\
&\cong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

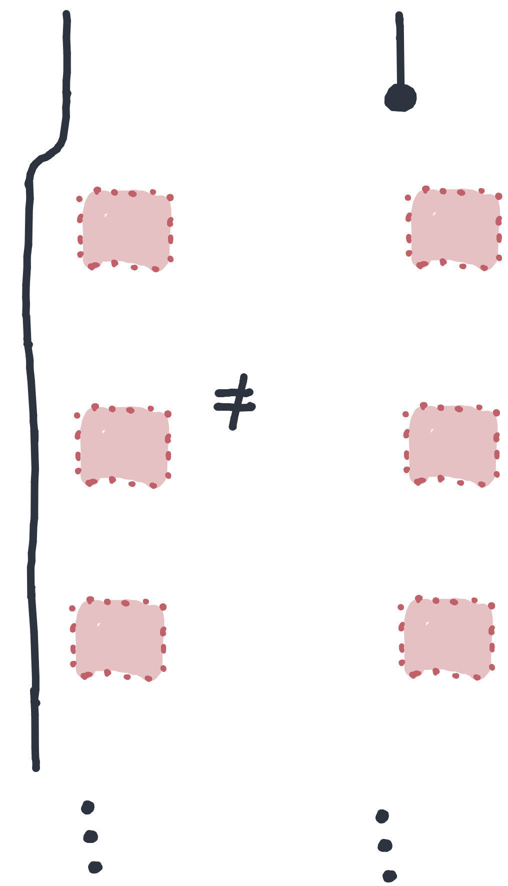


Observational Equivalence

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.

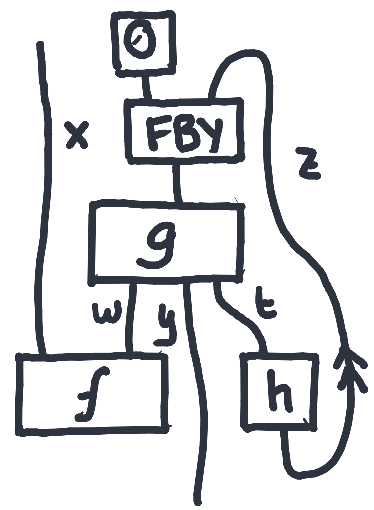




Two streams are observationally equal if their nth truncations can be made equal.

IMPLEMENTATION

Type theory for sym. monoidal categories with feedback.



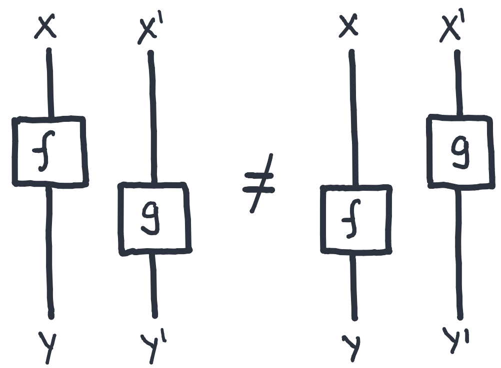

```
FBK z.
  SPLIT g(0 FBY h(z)) → [w,y,t] IN
  SPLIT f x w → [] IN
  RETURN y.
```

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

- ArrowLoop is for traced categories, in theory; it works for feedback.
- Github: mroman42/arrow-streams.
- Agda is a good candidate for coinduction.

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.

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

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

Only pure morphisms should slide.

THEOREM. If $\mathbb{V}$ is cartesian, and T is weakly cartesian then this final coalgebra is constructed by observational streams.

Extra : Expressivity

EXPRESSIVITY

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

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

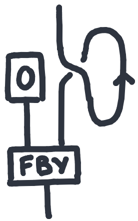
$$\begin{pmatrix} s_1 \\ \vdots \\ s_n \end{pmatrix}_{t+1} = \begin{pmatrix} \lambda_{11} & \cdots & \lambda_{1n} & \mu_{11} & \cdots & \mu_{1k} \\ \vdots & & \vdots & \vdots & & \vdots \\ \lambda_{n1} & \cdots & \lambda_{nn} & \mu_{1n} & \cdots & \mu_{nk} \end{pmatrix} \begin{pmatrix} s_1 \\ \vdots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

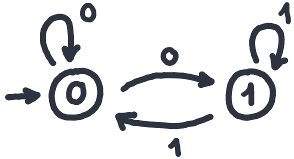
$$\begin{pmatrix} b_1 \\ \vdots \\ b_n \end{pmatrix}_{t} = \begin{pmatrix} \varphi_{11} & \cdots & \varphi_{1n} & \psi_{11} & \cdots & \psi_{1k} \\ \vdots & & \vdots & \vdots & & \vdots \\ \varphi_{n1} & \cdots & \varphi_{nn} & \psi_{1n} & \cdots & \psi_{nk} \end{pmatrix} \begin{pmatrix} s_1 \\ \vdots \\ s_n \\ a_1 \\ \vdots \\ a_k \end{pmatrix}_t$$

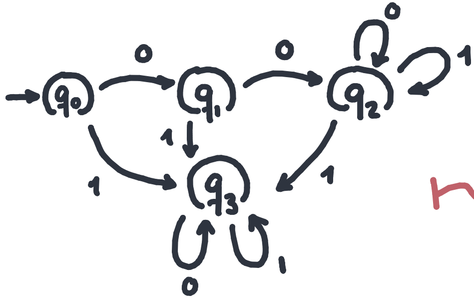
EXPRESSIVITY

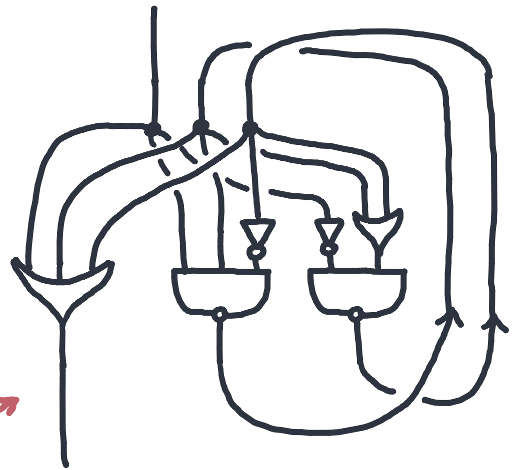
Orthogonal to the rest of the paper. What is the expressivity of the feedback syntax over some generators? The St(·) construction answers this.

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







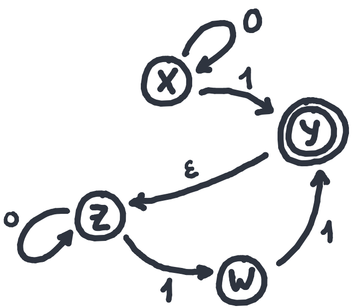


Expressivity

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

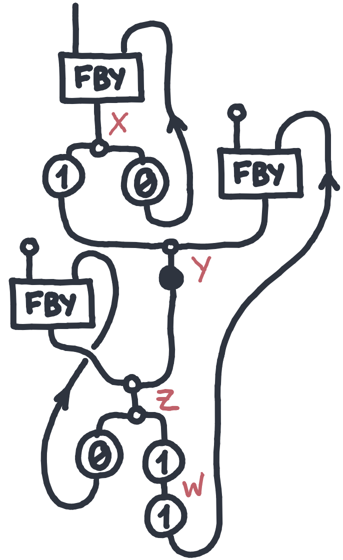
: $A+A \to A$ merge : $0 \to A$ unreachable : $A \to A+A$ choose : $A \to A$ concat "x" : $A \to A$ token

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



Park's equations.

$$X \Rightarrow 0X + 1Y$$
$$Y \stackrel{\bullet}{\Rightarrow} Z$$
$$Z \Rightarrow 0Z + 1W$$
$$W \Rightarrow 1Y$$



Monoidal category: TotalRel, +.

Extra : Coinduction

Coinductive Monoidal Streams

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

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

Quotiented by $f \approx g$, meaning

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

Coinductive Monoidal Streams

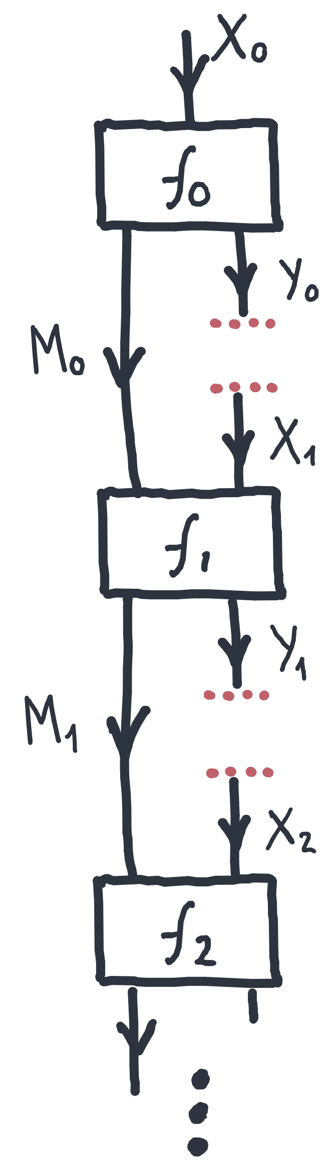
Reasoning with monoidal streams is easy: they are a final coalgebra.

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

Let me write $\mathbb{X} \in [\mathbb{N}, \mathbb{C}]$ for $X_0, X_1, \dots$; write $\mathbb{X}^+$ for $X_1, X_2, \dots$; and write $M \cdot \mathbb{X}$ for $M\otimes X_1, X_2, X_3, \dots$; the equation becomes

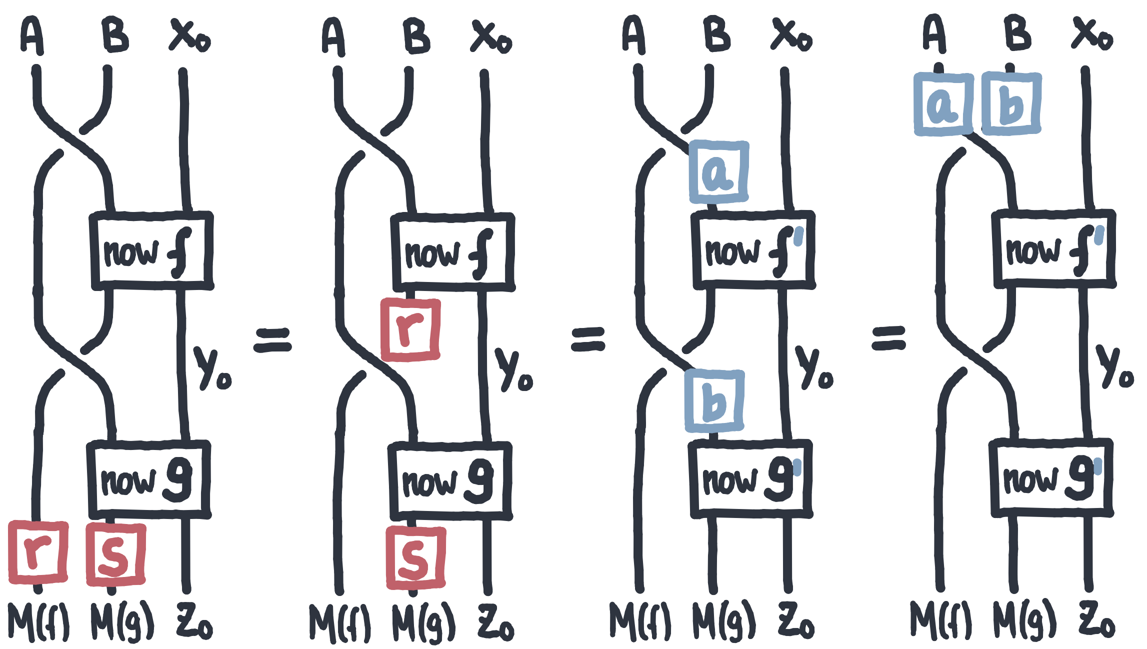
$$S(\mathbb{X},\mathbb{Y}) \cong \int^{M:\mathbb{C}} \hom(X_0, M\otimes Y_0) \times S(M\cdot\mathbb{X}^+; \mathbb{Y}^+).$$

The coend connects the M to the next step.



Coinductive Monoidal Streams

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



By coinduction,

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

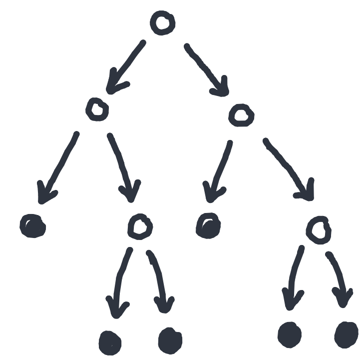
using $\mathrm{later}(f) \approx r \cdot \mathrm{later}(f')$,
$\mathrm{later}(g) \approx s \cdot \mathrm{later}(g')$.

When $a = id_A$, $b = id_B$, we have $(f \overset{r}{\approx} f') \wedge (g \overset{s}{\approx} g') \Rightarrow f^{A} ; g^{B} \overset{r \otimes s}{\approx} f'^{A} ; g'^{B}$.

ALGEBRA

Finite lists.

$L \cong A \times L + 1$



Finite trees.

$T \cong T^2 + 1$

1
2
3
⋮

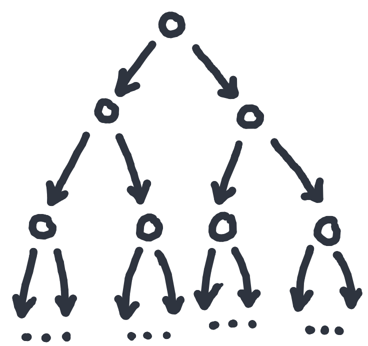
Natural numbers.

$N \cong N + 1$

Mathematics of syntax.

- Find the initial fixpoint of a functor, $FX \cong X$.
- Reason inductively.

Coalgebra



1
2
3
⋮
∞

Infinite lists.
$L \cong A \times L$

Infinite trees.
$T \cong T^2$

Conatural numbers.
$N \cong N + 1$

Algebra of state and transitions.

- Find the final fixpoint of a functor, $FX \cong X$.
- Reason coinductively.

COALGEBRA

THEOREM (LAMBEK). If the final coalgebra exists, it is a final fixpoint.
THEOREM (ADAMEK). If the following limit is a fixpoint, it is final.

$$\lim_{n\in\mathbb{N}}\left(1 \xleftarrow{!} F1 \xleftarrow{F!} FF1 \xleftarrow{FF!} FFF1 \longleftarrow \dots\right).$$

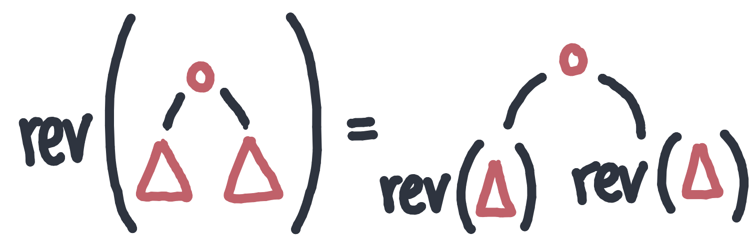
$$\begin{tikzcd} X \arrow[r, "f"] \arrow[d, "\alpha"'] \arrow[dr, phantom, "="] & U \arrow[d, "\cong"] \\ FX \arrow[r, "Ff"'] & FU \end{tikzcd}$$

That is, to compute a fixpoint, repeatedly apply F and it will converge.
Hopefully you will arrive to the fixpoint.
Initial algebras work too, but they will be less interesting.

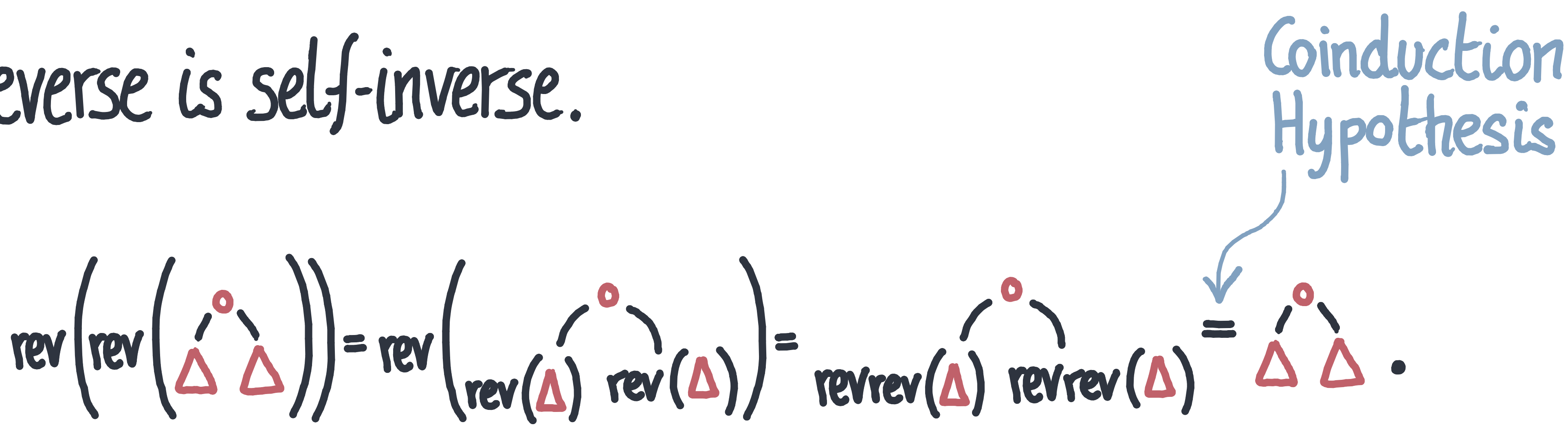
COINDUCTION

Reverse an infinite tree.

$$\mathrm{reverse}(a; l, r) := (a, \mathrm{reverse}(l), \mathrm{reverse}(r))$$



Reverse is self-inverse.



Extra : Observational Streams

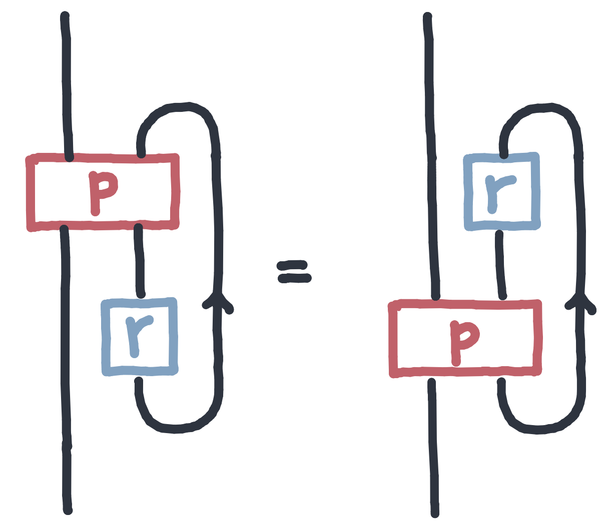
Process Interpretation

We think of functors $\mathbb{C}^{op} \times \mathbb{C} \to \mathrm{SET}$ as indexing families of processes, $P(M,N)$, by input M and output N. How to plug the output to the input?

Given $r : N \to M$,

- $P(id, r)(p) \in P(M,M)$, translate after reading,
- $P(r, id)(p) \in P(N,N)$, translate before writing.

These are "morally the same": dinaturally equivalent.



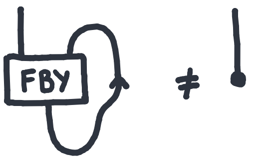
We write the set $\int^{X \in \mathbb{C}_{obj}} P(X,X)$ for $\sum_{X \in \mathbb{C}_{obj}} P(X,X)/\sim$, the coend of P,

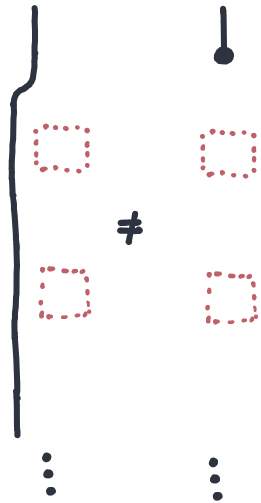
- $P(id,r)(p) \sim P(r,id)(p)$.

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.

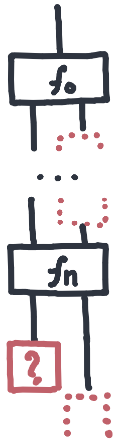




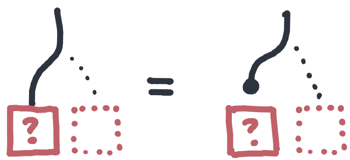
Observational Equivalence

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

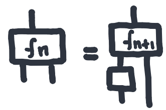
Intensional streams were the canonical fixpoint of the equation

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

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

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

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.

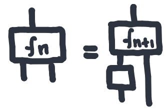
$$\int^{M:\mathbb{C}} \hom(X_0, M\otimes Y_0) \times \lim^{n\in\mathbb{N}} \int^{M_1,\dots,M_n} \prod_{i=1}^{n} \hom(M_{i-1}\otimes X_i, M_i\otimes Y_i)$$

$\cong$ ✓ fine

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

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

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

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

.

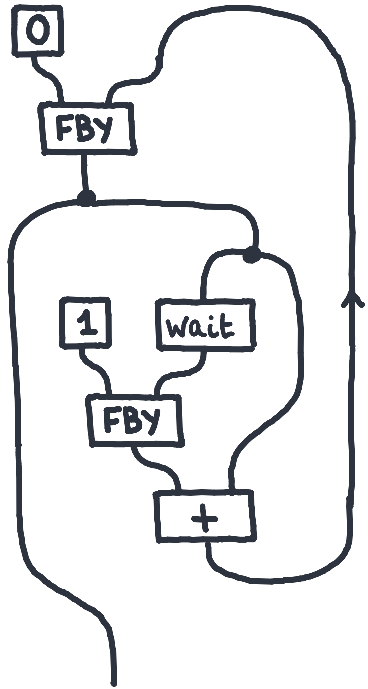
Nothing to worry in semicartesian, compact closed and freely generated monoidals.

Extra : EXAMPLE

Motivation: Dataflow Programming

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

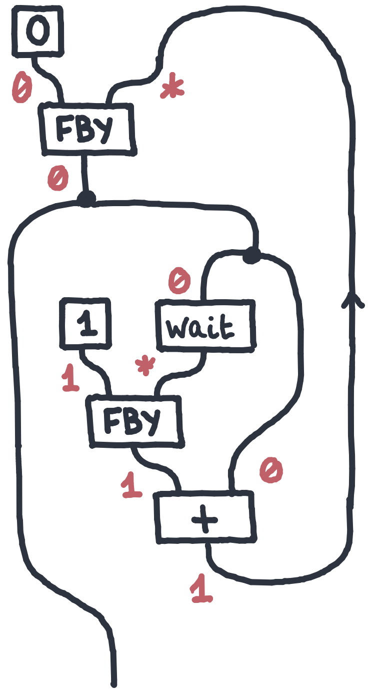
Time.	fib	WAIT(fib)	1 FBY WAIT fib
0			
1			
2			
3			
4			
⋮			



Motivation: Dataflow Programming

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

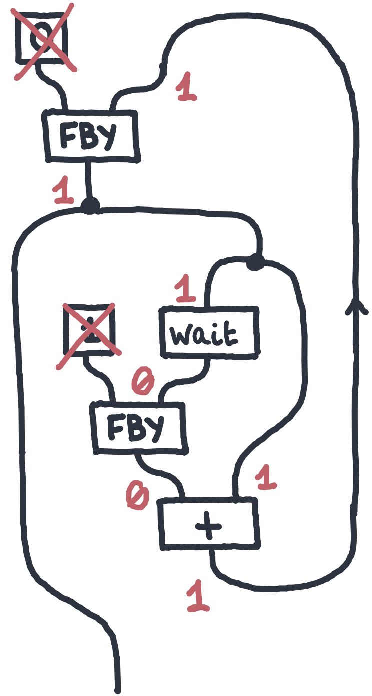
Time.	fib	WAIT(fib)	1 FBY WAIT fib
0	0	*	1
1			
2			
3			
4			
⋮			



Motivation: Dataflow Programming

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

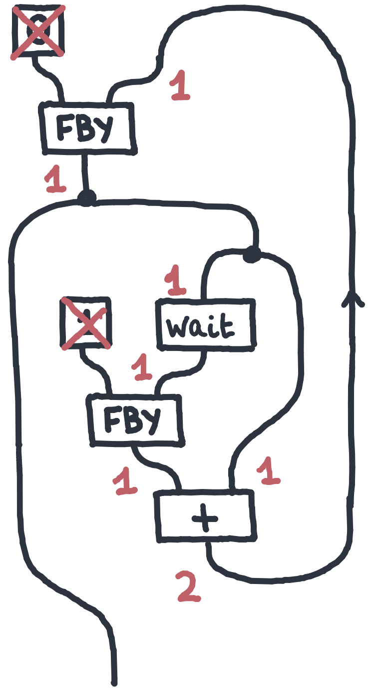
Time.	fib	WAIT(fib)	1 FBY WAIT fib
0	0	*	1
1	1	0	0
2			
3			
4			
⋮			



Motivation: Dataflow Programming

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

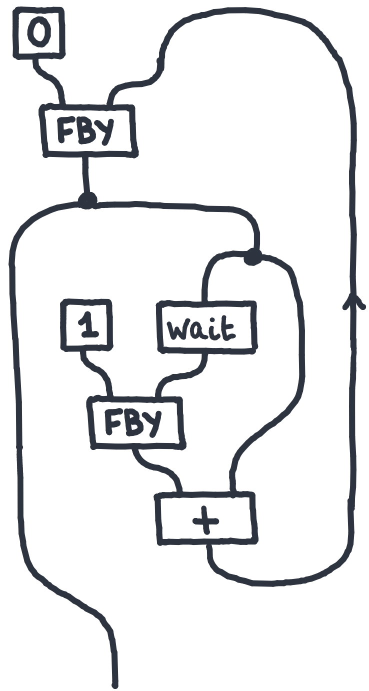
Time.	fib	WAIT(fib)	1 FBY WAIT fib
0	0	*	1
1	1	0	0
2	1	1	1
3			
4			
⋮			



Motivation: Dataflow Programming

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

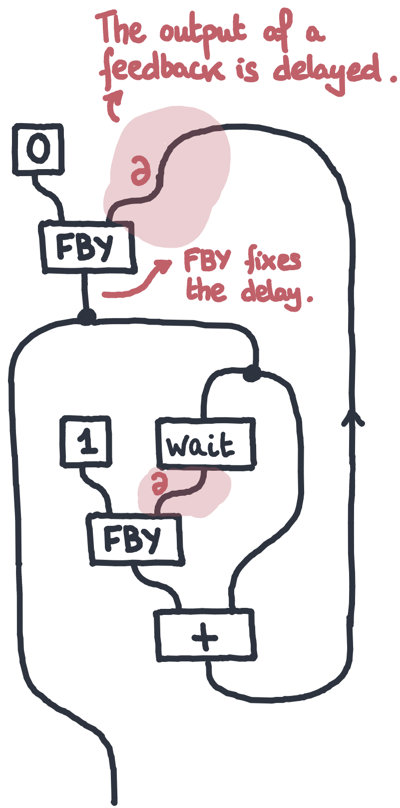
Time.	fib	WAIT(fib)	1 FBY WAIT fib
0	0	*	1
1	1	0	0
2	1	1	1
3	2	1	1
4	3	2	2
⋮	⋮	⋮	⋮



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.

Extra : Lenses

Dinaturality and Coends

Given $P : \mathbb{C}^{op} \times \mathbb{C} \to \textsc{Set}$, consider the set of all processes: $\sum_{X \in \mathbb{C}_{obj}} P(X,X)$.
Dinatural equivalence is the smallest equivalence relation generated by

$$P(id,r)(p) \sim P(r,id)(p).$$

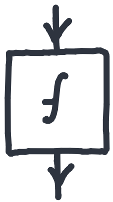
We write the set $\int^{X \in \mathbb{C}_{obj}} P(X,X)$ for $\sum_{X \in \mathbb{C}_{obj}} P(X,X)/\sim$, the coend of P.

"The coend, $\int^{X \in \mathbb{C}_{obj}}$, connects both X's."

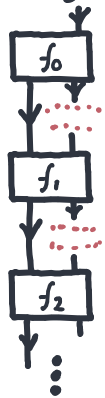
MOTIVATION

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.

Adámek's Theorem

THEOREM. Let $\mathbb{C}$ be a category with a terminal object and let F be any endofunctor $F : \mathbb{C} \to \mathbb{C}$. If the following limit

$$L \cong \lim_{n \in \mathbb{N}} \left(1 \xleftarrow{!} F1 \xleftarrow{F!} FF1 \xleftarrow{FF!} FFF1 \leftarrow \dots \right)$$

exists and is preserved by F, then it carries a final coalgebra structure.

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

Initial algebras work too, but they will be less interesting in our case.

MOTIVATION

Lenses can be used to describe a single exchange,

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

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

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

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

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

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

Causal Stream Functions

DEFINITION. A causal stream function $f : \mathbb{X} \to \mathbb{Y}$ is a family of functions

$f_n : X_0 \times \dots \times X_n \to Y_n$.

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

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

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

We have

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

Uustalu, Vene   Sprunger, Katsumata 19

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

OTHER THEORIES OF PROCESSES

These will be monoidal streams.

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

THEOREM (DLdFR) The non-empty list functor

$$\mathrm{List}^{+}(\mathbb{X})_n = \bigotimes_{i=0}^{n} X_i$$

is a monoidal comonad if and only if $(\otimes)$ is a cartesian product.

So far, so good with deterministic functions.