Effectful Trace Semantics via Effectful Streams

Filippo Bonchi[1], Elena Di Lavore[1], Mario Román[2]

[1]Università di Pisa, [2]University of Oxford.

6th April, CMCS '24

ERC Blast project.
EU Estonian IT Academy.

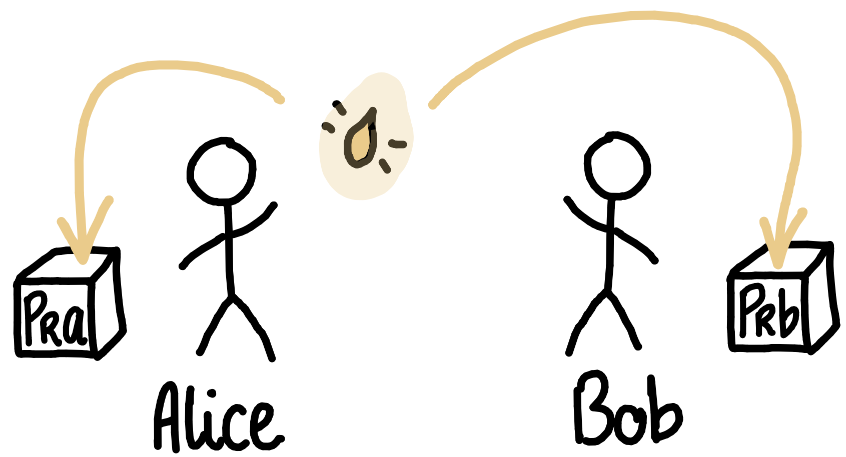
PART 0: MOTIVATION

Example: Stream Cipher

$\mathrm{alice}(m)^{\circ} =$
$\quad \mathrm{seed}() \rightsquigarrow ()$
$\quad \mathrm{rand}_a() \rightsquigarrow k_a$
$\quad \mathrm{return}(m \oplus k_a);$

$\mathrm{alice}(m)^{+\circ} =$
$\quad \mathrm{rand}_a() \rightsquigarrow k_a$
$\quad \mathrm{return}(m \oplus k_a);$

$\mathrm{alice}(m)^{++} = \mathrm{alice}(m)^{+};$



$\mathrm{bob}(n)^{\circ} =$
$\quad \mathrm{rand}_b() \rightsquigarrow k_b$
$\quad \mathrm{return}(n \oplus k_b);$

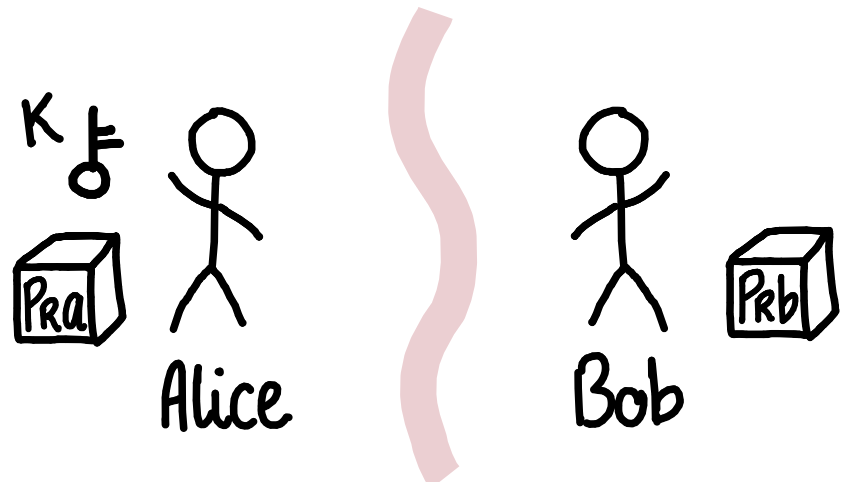
$\mathrm{bob}(n)^{+} = \mathrm{bob}(n);$

Example: Stream Cipher

$\mathrm{alice}(m)^{\circ} =$
$\quad \mathrm{seed}() \rightsquigarrow ()$
$\quad \mathrm{rand}_a() \rightsquigarrow k_a$
$\quad \mathrm{return}(m \oplus k_a);$

$\mathrm{alice}(m)^{+\circ} =$
$\quad \mathrm{rand}_a() \rightsquigarrow k_a$
$\quad \mathrm{return}(m \oplus k_a);$

$\mathrm{alice}(m)^{++} = \mathrm{alice}(m)^{+};$



$\mathrm{bob}(n)^{\circ} =$
$\quad \mathrm{rand}_b() \rightsquigarrow k_b$
$\quad \mathrm{return}(n \oplus k_b);$

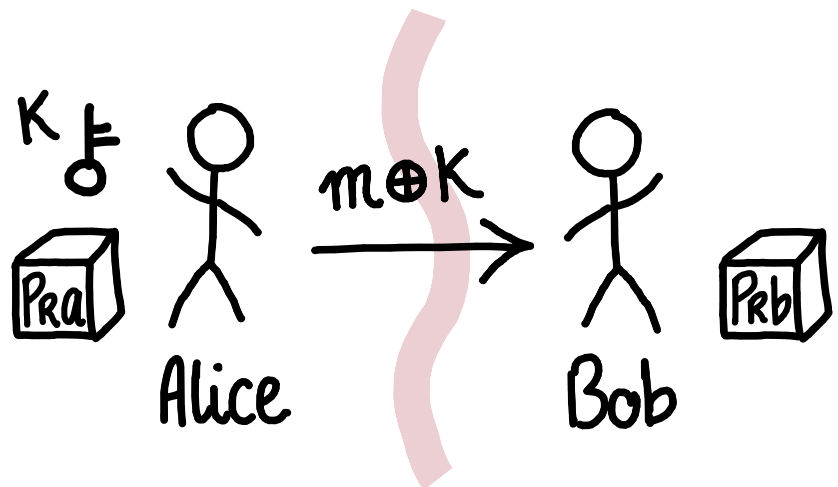
$\mathrm{bob}(n)^{+} = \mathrm{bob}(n);$

EXAMPLE: STREAM CIPHER

$\mathrm{alice}(m)^{\circ} =$
$\quad \mathrm{seed}() \rightsquigarrow ()$
$\quad \mathrm{rand}_a() \rightsquigarrow k_a$
$\quad \mathrm{return}(m \oplus k_a);$

$\mathrm{alice}(m)^{+\circ} =$
$\quad \mathrm{rand}_a() \rightsquigarrow k_a$
$\quad \mathrm{return}(m \oplus k_a);$

$\mathrm{alice}(m)^{++} = \mathrm{alice}(m)^{+};$

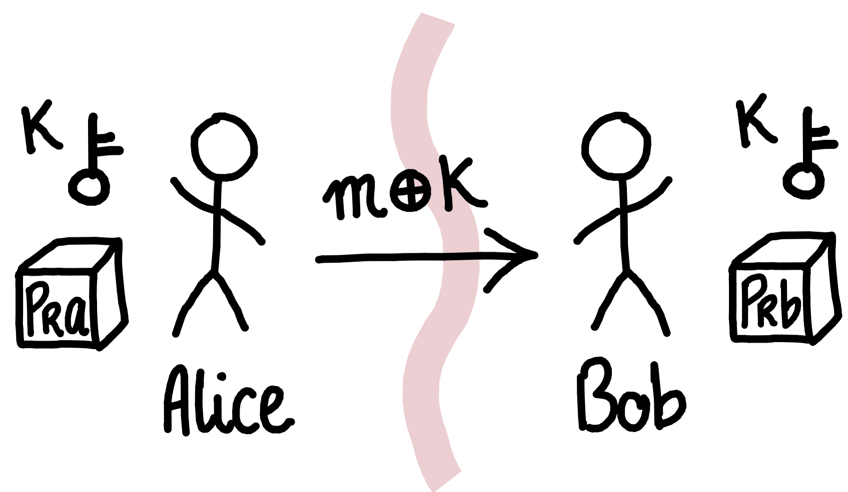


$\mathrm{bob}(n)^{\circ} =$
$\quad \mathrm{rand}_b() \rightsquigarrow k_b$
$\quad \mathrm{return}(n \oplus k_b);$

$\mathrm{bob}(n)^{+} = \mathrm{bob}(n);$

EXAMPLE: STREAM CIPHER

$alice(m)^{\circ} =$
  $seed() \rightsquigarrow ()$
  $rand_a() \rightsquigarrow k_a$
  $return(m \oplus k_a);$
$alice(m)^{+\circ} =$
  $rand_a() \rightsquigarrow k_a$
  $return(m \oplus k_a);$
$alice(m)^{++} = alice(m)^{+};$

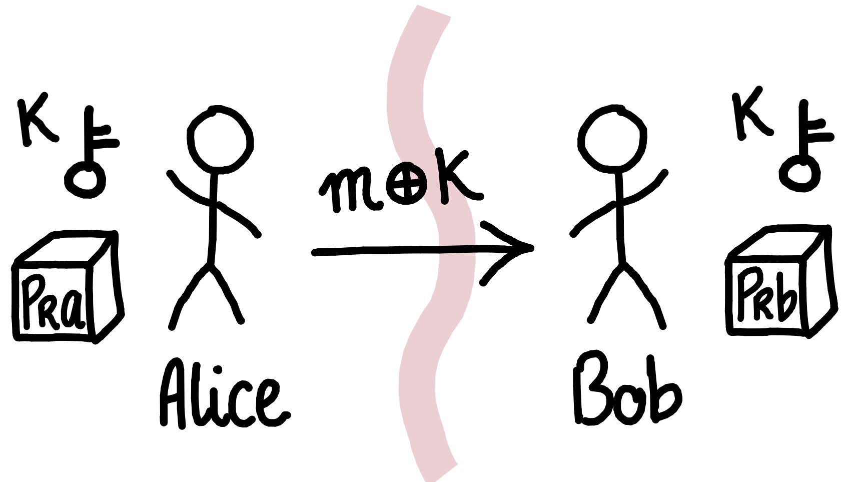


$bob(n)^{\circ} =$
  $rand_b() \rightsquigarrow k_b$
  $return(n \oplus k_b);$

$bob(n)^{+} = bob(n);$

Example: Stream Cipher

$$
\begin{aligned}
&\mathit{alice}(m)^{\circ} = \\
&\quad \mathit{seed}() \rightsquigarrow () \\
&\quad \mathit{rand}_a() \rightsquigarrow k_a \\
&\quad \mathit{return}(m \oplus k_a); \\
&\mathit{alice}(m)^{+\circ} = \\
&\quad \mathit{rand}_a() \rightsquigarrow k_a \\
&\quad \mathit{return}(m \oplus k_a); \\
&\mathit{alice}(m)^{++} = \mathit{alice}(m)^{+};
\end{aligned}
$$



$$
\begin{aligned}
&\mathit{bob}(n)^{\circ} = \\
&\quad \mathit{rand}_b() \rightsquigarrow k_b \\
&\quad \mathit{return}(n \oplus k_b); \\
&\mathit{bob}(n)^{+} = \mathit{bob}(n);
\end{aligned}
$$

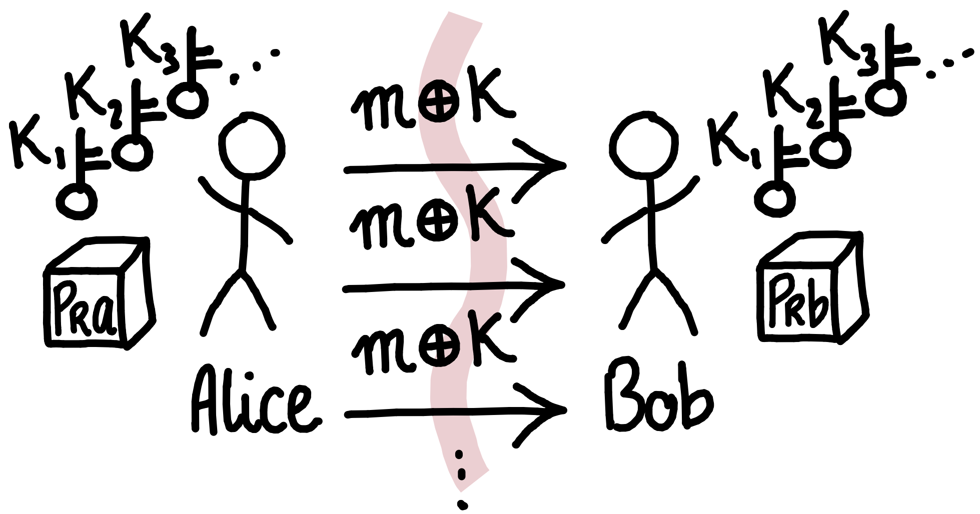
$$n \oplus K = (m \oplus K) \oplus K = m$$

Example: Stream Cipher

$alice(m)^{\circ} =$
$\quad seed() \rightsquigarrow ()$
$\quad rand_a() \rightsquigarrow k_a$
$\quad return(m \oplus k_a);$

$alice(m)^{+\circ} =$
$\quad rand_a() \rightsquigarrow k_a$
$\quad return(m \oplus k_a);$

$alice(m)^{++} = alice(m)^{+};$



$bob(n)^{\circ} =$
$\quad rand_b() \rightsquigarrow k_b$
$\quad return(n \oplus k_b);$

$bob(n)^{+} = bob(n);$

Part 1: Effectful Copy-Discard Categories

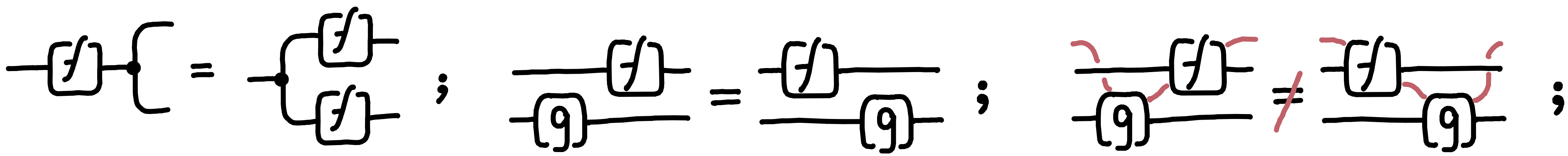
Effectful Copy-Discard

$$\mathbb{V} \xrightarrow{\text{i.i.o}} \mathbb{P} \xrightarrow{\text{i.i.o}} \mathbb{C}$$

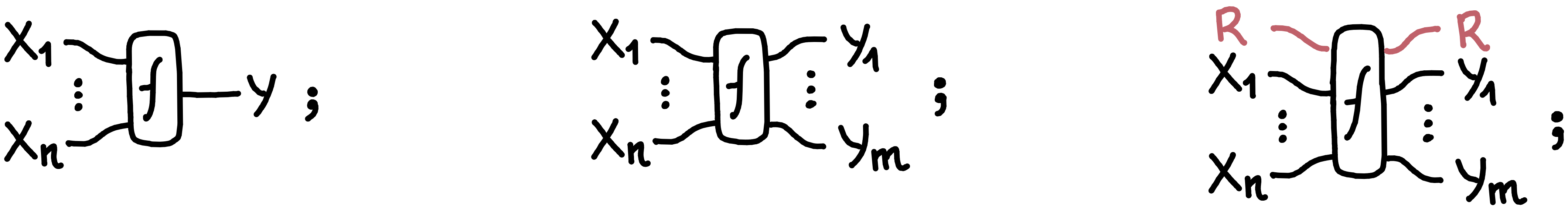
$\mathbb{V}$	$\mathbb{P}$	$\mathbb{C}$
cartesian category	sym. monoidal category	sym. premonoidal category
"values"	"pure effects" "local effects"	"global effects"

- Kleisli categories of strong (pro)monads: $(\mathbb{V}, \mathcal{Z}(\mathrm{Kl}(T)), \mathrm{Kl}(T))$.

Effectful Copy-Discard



Cartesian. Monoidal. Premonoidal.



Jeffrey (1998). Premonoidal Categories and a Graphical View of Programs.

Effectful Copy-Discard: Do-Notation

$$\frac{}{\Gamma \vdash x : X}\ (x \in \Gamma)$$

$$\frac{\Gamma \vdash t_1 : X_1 \quad \dots \quad \Gamma \vdash t_n : X_n}{\Gamma \vdash f(t_1, \dots, t_n) : Y}$$

Lawvere theory syntax.

$$\frac{}{\Gamma \Vdash \mathrm{return}\,(t_1, \dots, t_n) : X_1, \dots, X_n}$$

$$\frac{y_1, \dots, y_m, \Gamma \Vdash \mathrm{prog} : Z_1, \dots, Z_m}{\Gamma \Vdash g(t_1, \dots, t_n) \to y_1, \dots, y_m\ \mathrm{prog} : Z_1, \dots, Z_m}$$

$$\frac{y_1, \dots, y_m, \Gamma \Vdash \mathrm{prog} : Z_1, \dots, Z_m}{\Gamma \Vdash h(t_1, \dots, t_n) \rightsquigarrow y_1, \dots, y_m\ \mathrm{prog} : Z_1, \dots, Z_m}$$

For each tuple of values, $\Gamma \vdash t_1 : X_1 \dots \Gamma \vdash t_n : X_n$.

Do-notation generators.

EFFECTFUL COPY-DISCARD: DO-NOTATION

THEOREM. Do-notation derivations form the free strict effectful copy-discard over a signature.

$$\begin{tikzcd} \mathsf{EcdSig} \arrow[r, bend left, "\mathsf{Do}"] \arrow[r, phantom, "\perp"] & \mathsf{EcdCat} \arrow[l, bend left, "\mathsf{Forget}"] \end{tikzcd}$$

EXAMPLE. Signature

$$\Sigma = \left\{ \begin{array}{l} (\oplus) : (2^n, 2^n) \longrightarrow 2^n \\ \mathrm{seed} : () \rightsquigarrow () \\ \mathrm{rand}_a : () \rightsquigarrow (2^n) \\ \mathrm{rand}_b : () \rightsquigarrow (2^n) \end{array} \right\}$$

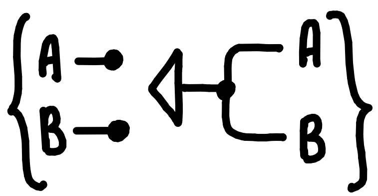
EXAMPLE. Program.

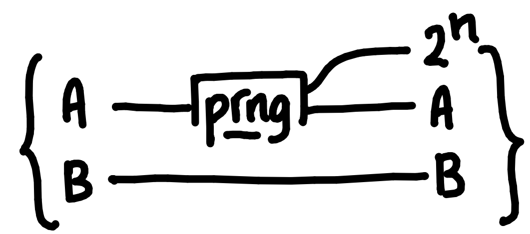
$$\begin{array}{l} \mathrm{alice}(m) = \\ \quad \mathrm{seed}() \rightsquigarrow () \\ \quad \mathrm{rand}_a() \rightsquigarrow k_a \\ \quad \mathrm{rand}_b() \rightsquigarrow k_b \\ \quad \mathrm{return}((m \oplus k_b) \oplus k_a); \end{array} \in \mathrm{Do}(\Sigma)(2^n; 2^n)$$

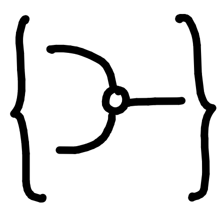
EFFECTFUL COPY-DISCARD: SEMANTICS

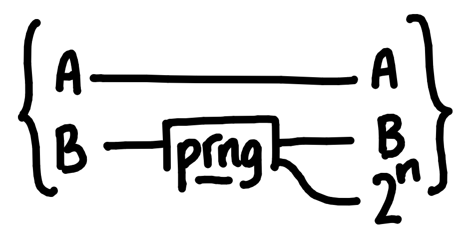
We get semantics on any effectful copy-discard category.

EXAMPLE. $[\![\cdot]\!] : \mathrm{Do}(\Sigma) \longrightarrow (\mathsf{Set}, \mathsf{Stoch}, \mathsf{StStoch}_{2^n \times 2^n})$

$[\![\mathrm{seed}]\!] =$  ;

$[\![\mathrm{rand}_A]\!] =$  ;

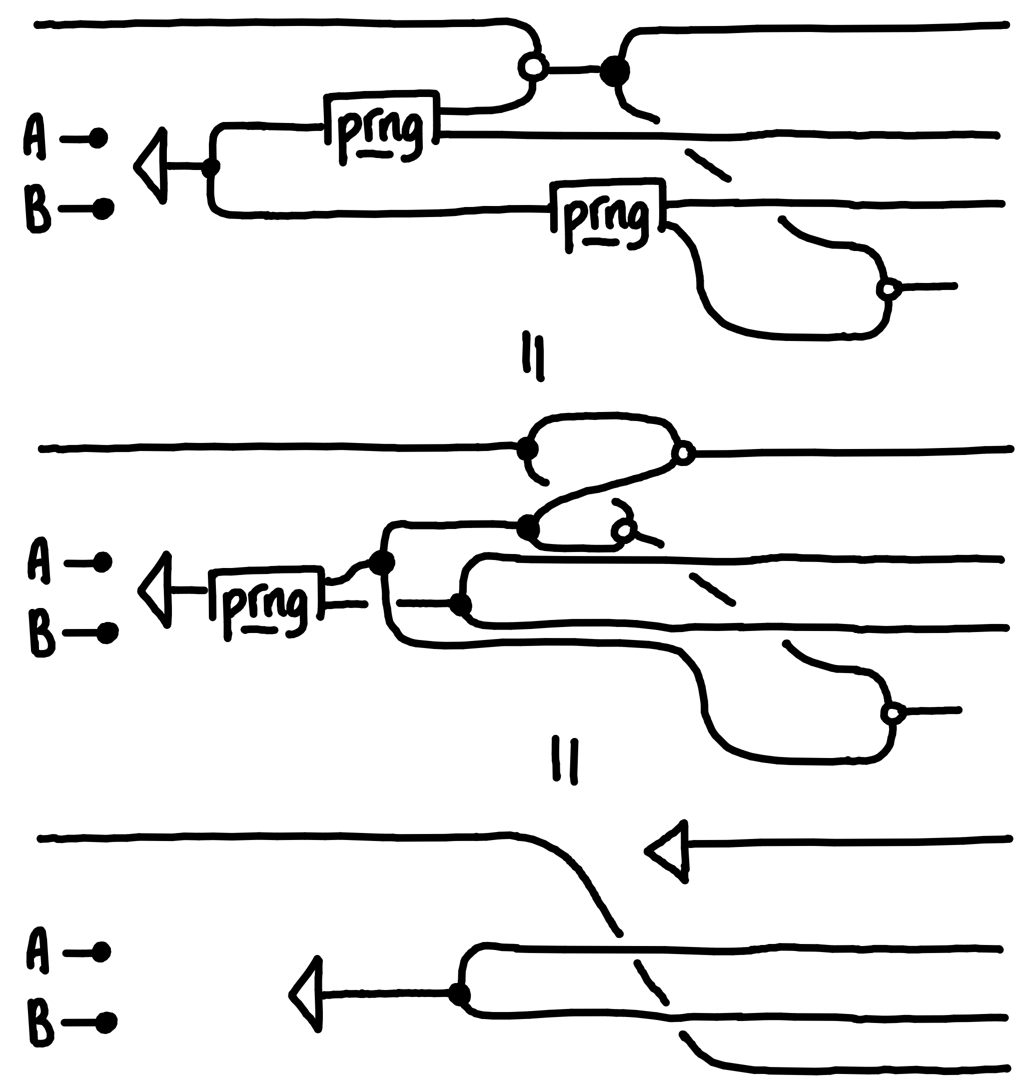
$[\![\oplus]\!] =$  ;

$[\![\mathrm{rand}_B]\!] =$  ;

Effectful Copy-Discard: Semantics

Example: One step of the stream cipher is correct, assuming reasonable axioms for our PRNG.

$\mathrm{seed}() \rightsquigarrow ()$
$\mathrm{rand}_a() \rightsquigarrow k_a$
$\mathrm{rand}_b() \rightsquigarrow k_b$
$\mathrm{return}\,((m \oplus k_b) \oplus k_a, m \oplus k_b);$



☐ Broadbent & Karvonen (2023). Categorical Composable Cryptography.

PART 2 : Streams

Effectful Streams

Let $(\mathbb{V}, \mathbb{P}, \mathbb{C})$ be an effectful copy-discard category; we write $\mathbb{A}, \mathbb{B}, \dots,$ for streams of objects, with head $A^\circ \in \mathbb{C}_{obj}$ and tail $\mathbb{A}^+ \in \mathbb{C}_{obj}^{\omega}$. We define $(M \cdot \mathbb{A})^\circ = M \otimes A^\circ$ and $(M \cdot \mathbb{A})^+ = \mathbb{A}^+$.

Definition. An effectful stream $f : \mathbb{A} \to \mathbb{B}$ is

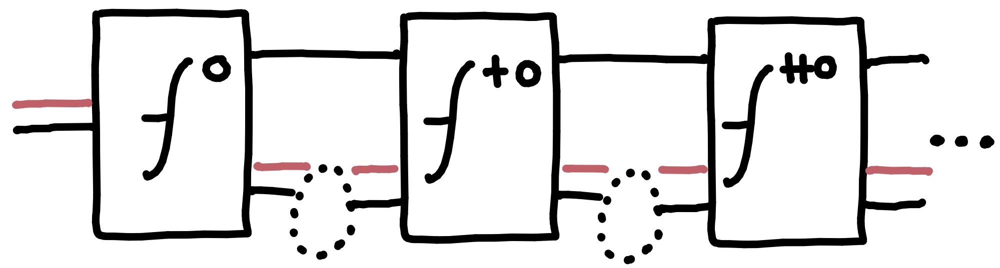
- a memory, $M \in \mathbb{C}_{obj}$;
- an effectful morphism $f^\circ : A^\circ \rightsquigarrow M \otimes B^\circ$;
- an effectful stream $f^+ : M \cdot \mathbb{A}^+ \to \mathbb{B}^+$.

Effectful Streams

Effectful streams form an effectful category. Composition interleaves.

$$
\begin{array}{l}
\mathrm{alice}(m)^{\circ} = \\
\quad \mathrm{seed}() \rightsquigarrow () \\
\quad \mathrm{rand}_a() \rightsquigarrow k_a \\
\quad \mathrm{return}(m \oplus k_a); \\
\mathrm{alice}(m)^{+\circ} = \\
\quad \mathrm{rand}_a() \rightsquigarrow k_a \\
\quad \mathrm{return}(m \oplus k_a); \\
\mathrm{alice}(m)^{++} = \mathrm{alice}(m)^{+};
\end{array}
\quad ; \quad
\begin{array}{l}
\mathrm{bob}(n)^{\circ} = \\
\quad \mathrm{rand}_b() \rightsquigarrow k_b \\
\quad \mathrm{return}(n \oplus k_b); \\
\\
\mathrm{bob}(n)^{+} = \mathrm{bob}(n);
\end{array}
\quad = \quad
\begin{array}{l}
\mathrm{comp}^{\circ}(m) = \\
\quad \mathrm{seed}() \rightsquigarrow () \\
\quad \mathrm{rand}_a() \rightsquigarrow k_a \\
\quad \mathrm{rand}_b() \rightsquigarrow k_b \\
\quad \mathrm{return}((m \oplus k_b) \oplus k_a); \\
\\
\mathrm{comp}^{+}(m) = \\
\quad \mathrm{rand}_a() \rightsquigarrow k_a \\
\quad \mathrm{rand}_b() \rightsquigarrow k_b \\
\quad \mathrm{return}((m \oplus k_b) \oplus k_a); \\
\\
\mathrm{comp}(m)^{++} = \mathrm{comp}(m)^{+};
\end{array}
$$

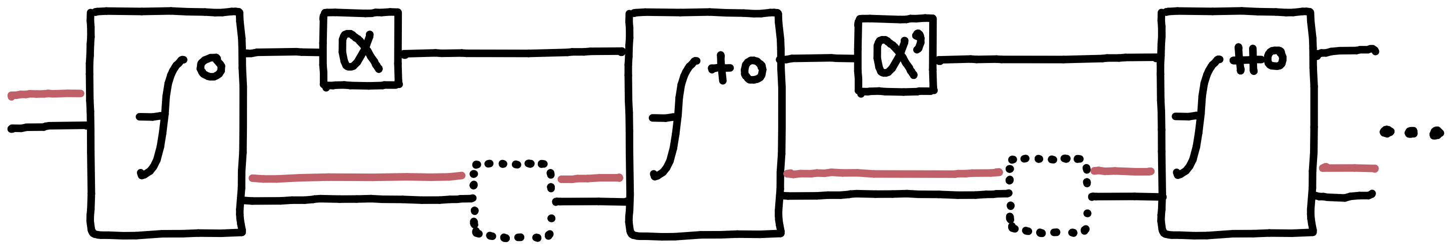
EFFECTFUL STREAMS: DINATURALITY



DEFINITION. An effectful stream $f : \mathbb{A} \to \mathbb{B}$ is

- a memory, $M \in \mathbb{C}_{obj}$;
- an effectful morphism $f^{\circ} : A^{\circ} \rightsquigarrow M \otimes B^{\circ}$;
- an effectful stream $f^{+} : M \cdot \mathbb{A}^{+} \to \mathbb{B}^{+}$.

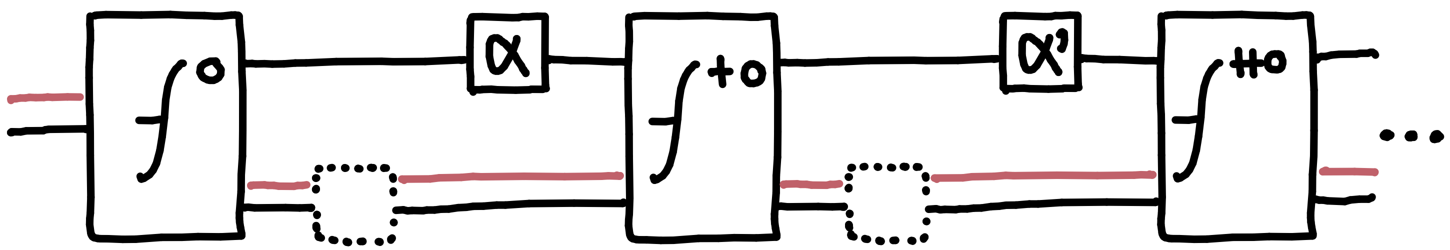
Effectful Streams: Dinaturality



DEFINITION. Dinatural equivalence is the minimal equivalence relation equating

$$
\left|\begin{array}{l}
lhs^{\circ}(x) = \\
\quad f^{\circ}(x) \rightsquigarrow m, y \\
\quad \alpha(m) \to n \\
\quad \mathrm{return}\ (n, y) \\
lhs^{+}(n, y) = \\
\quad f^{+}(n, y)
\end{array}\right.
=
\left|\begin{array}{l}
rhs^{\circ}(x) = \\
\quad f^{\circ}(x) \rightsquigarrow m, y \\
\quad \mathrm{return}\ (m, y) \\
rhs^{+}(m, y) = \\
\quad \alpha(m) \to n \\
\quad f^{+}(n, y)
\end{array}\right.
$$

EFFECTFUL STREAMS: DINATURALITY



DEFINITION. Dinatural equivalence is the minimal equivalence relation equating

$$
\left|\begin{array}{l}
\mathrm{lhs}^{\circ}(x) = \\
\quad f^{\circ}(x) \rightsquigarrow m, y \\
\quad \alpha(m) \to n \\
\quad \mathrm{return}\,(n, y) \\
\mathrm{lhs}^{+}(n, y) = \\
\quad f^{+}(n, y)
\end{array}\right.
=
\left|\begin{array}{l}
\mathrm{rhs}^{\circ}(x) = \\
\quad f^{\circ}(x) \rightsquigarrow m, y \\
\quad \mathrm{return}\,(m, y) \\
\mathrm{rhs}^{+}(m, y) = \\
\quad \alpha(m) \to n \\
\quad f^{+}(n, y)
\end{array}\right.
$$

EFFECTFUL STREAMS

THEOREM. Streams quotiented by dinaturality are the final fixpoint of the following equation of profunctors,

$$\mathrm{Stream}(\mathbb{A};\mathbb{B}) = \int^{M \in \mathbb{C}} \hom_{\mathbb{C}}(A^{\circ}; M \otimes B^{\circ}) \times \mathrm{Stream}(M \cdot \mathbb{A}^{+}; \mathbb{B}^{+}).$$

In other words, the final coalgebra of the functor

$$\Phi(Q)(\mathbb{A};\mathbb{B}) = \int^{M \in \mathbb{C}} \hom_{\mathbb{C}}(A^{\circ}; M \otimes B^{\circ}) \times Q(M \cdot \mathbb{A}^{+}; \mathbb{B}^{+}),$$

of type $\Phi : [(\mathbb{C}^{\omega})^{op} \times (\mathbb{C}^{\omega}), \mathrm{SET}] \longrightarrow [(\mathbb{C}^{\omega})^{op} \times (\mathbb{C}^{\omega}), \mathrm{SET}]$.

☐ c.f. Di Lavore, de Felice, Román (2022). ☐ c.f. Profunctor Optics.

Part 3: From Causal Functions to Effectful Streams

CARTESIAN STREAMS (Sprunger, Katsumata)

THEOREM. In a cartesian monoidal category, streams $f_i : \mathbb{A} \to \mathbb{B}$ are causal extensional sequences,

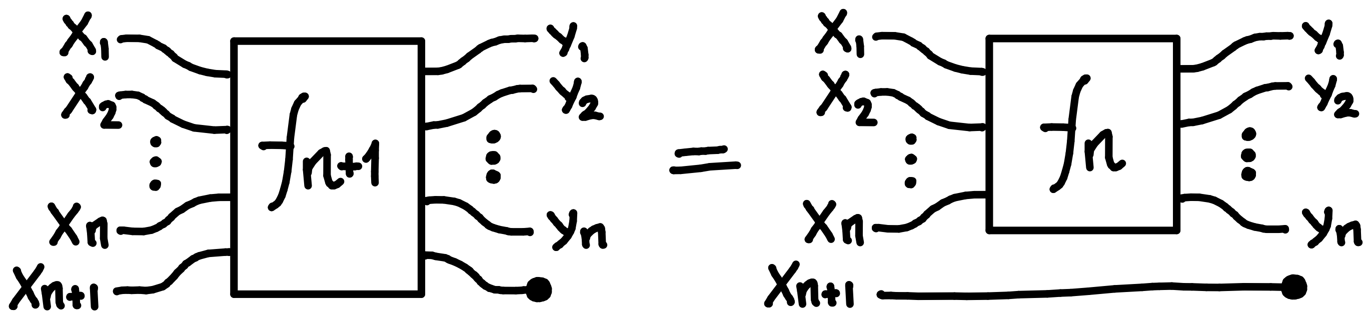
$$f_n : X_1 \times \cdots \times X_n \longrightarrow Y_n.$$

- Sprunger, Katsumata (2019). Differentiable Causal Computations via Delayed Trace.
- Uustalu, Vene (2008). Comonadic Notions of Computation.

PROBABILISTIC STREAMS

THEOREM. In a Markov category with conditionals and ranges, streams $f_i : \mathbb{A} \to \mathbb{B}$ are stochastic processes,

$$f_n : X_1 \otimes \cdots \otimes X_n \longrightarrow Y_1 \otimes \cdots \otimes Y_n .$$

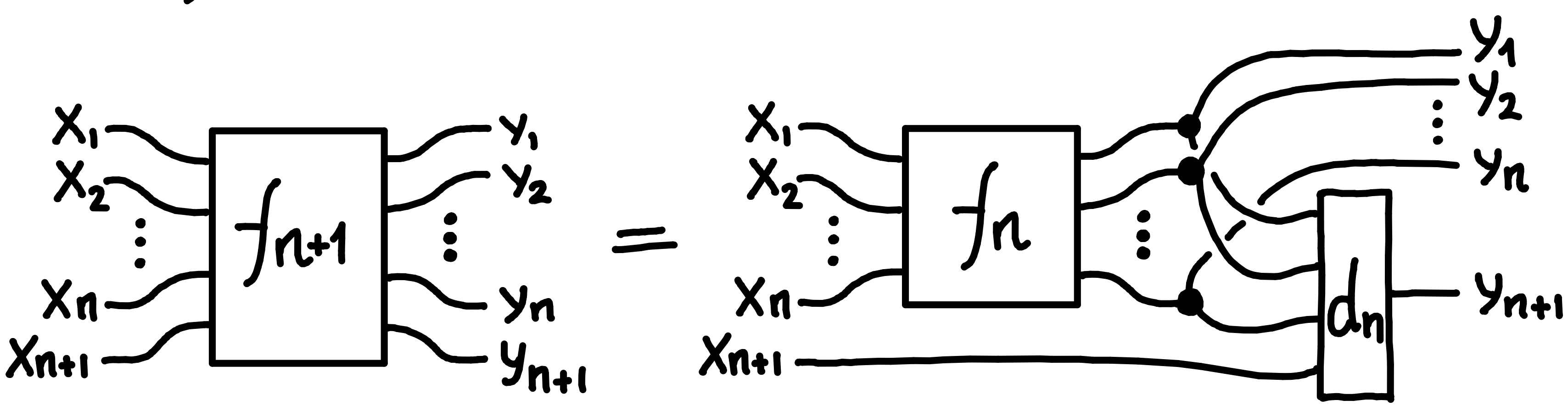


- Carette, de Visme, Perdrix (2021). Graphical Language with Delayed Trace.
- Di Lavore, de Felice, Román (2022). Monoidal Streams for Dataflow Programming.

Partial and Relational Streams

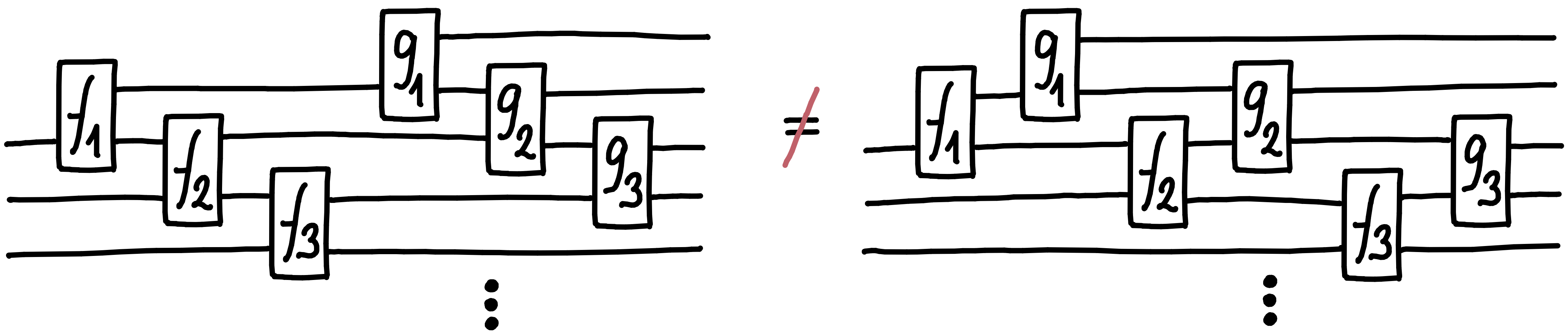
THEOREM. In a copy-discard category with quasi-total conditionals and ranges, streams $f_i : A \to B$ are causal processes,

$$f_n : X_1 \otimes \cdots \otimes X_n \longrightarrow Y_1 \otimes \cdots \otimes Y_n .$$



Stateful Streams

Causal process composition works for monoidals but not for premonoidals: it needs interleaving.



CONCLUSION. Effectful streams coincide with causal processes in all cases and they moreover add the stateful premonoidal case.

PART 4: TRACES

Effectful Mealy Machines (transducers)

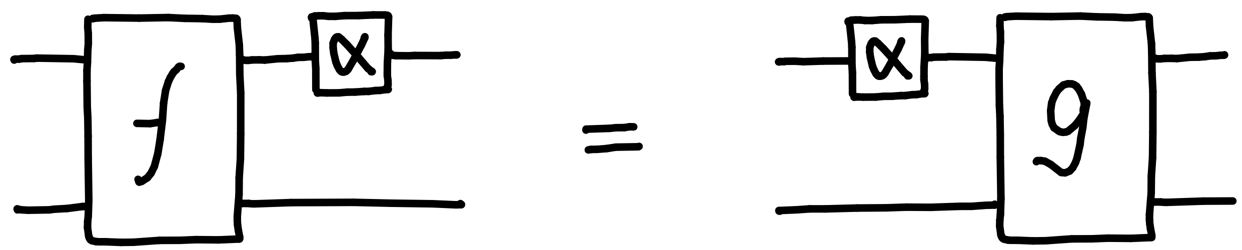
DEFINITION. An *effectful Mealy machine* in an effectful copy-discard category $(\mathbb{V}, \mathbb{P}, \mathbb{C})$, with input on $A \in \mathbb{C}_{obj}$ and with outputs on $B \in \mathbb{C}_{obj}$, consists of

- a state space, $U \in \mathbb{C}_{obj}$;
- an initial space, $i : I \rightsquigarrow U$;
- a transition morphism, $f : U \otimes A \rightsquigarrow U \otimes B$.

- Hoshino, Muroya, Hasuo (2014). Memoryful Geometry of Interaction.
- Katis, Sabadini, Walters (1997). Bicategories of Processes.

Bisimulation

A homomorphism of effectful Mealy machines, $\alpha\colon (U,i,f) \Rightarrow (V,j,g)$, is a value morphism $\alpha\colon U \to V$ such that $i ; \alpha = j$ and $f ; (\alpha \otimes id) = (\alpha \otimes id) ; g$,

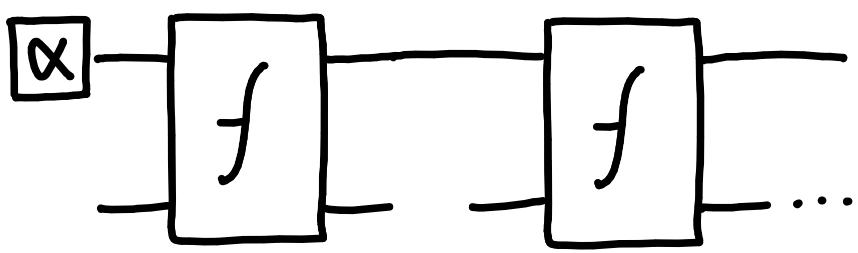


Bisimulation is connectedness by homomorphisms.

- Hoshino, Muroya, Hasuo (2014). Memoryful Geometry of Interaction.
- Katis, Sabadini, Walters (1997). Bicategories of Processes.

TRACES

An "unrolling" functor transforms Mealy machines into the effectful stream they generate by repetition.



Object A	Constant seq. A, A, A
Mealy machine	Stream
Mealy homomorphism	Equality

COROLLARY. Bisimulation implies trace equivalence.

End