

Padova

9 April 2024

EFFECTFUL TRACE SEMANTICS  
VIA  
EFFECTFUL STREAMS

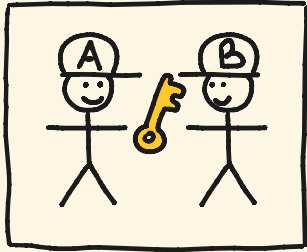
Silippo Bonchi  
Università di Pisa

Elena Di Loreto  
Università di Pisa

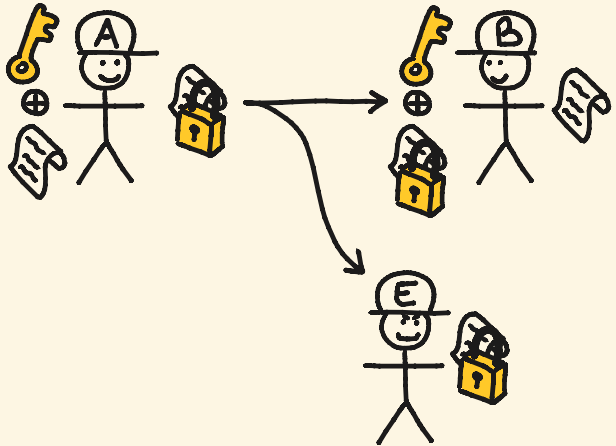
Mario Román  
University of Oxford

# ONE-TIME PAD PROTOCOL

1. share a key through a secure channel



2. send an encrypted message through a public channel





# REPEATING THE ONE-TIME PAD

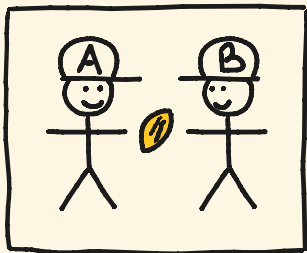
Sending  $n$  messages securely requires  $n$  private keys  
↳ not very useful

- ⇒ • privately share a seed 🍀
- use identical pseudorandom number generators to obtain a new key for each message



# STREAM CIPHER PROTOCOL (1)

1. share a seed through a secure channel

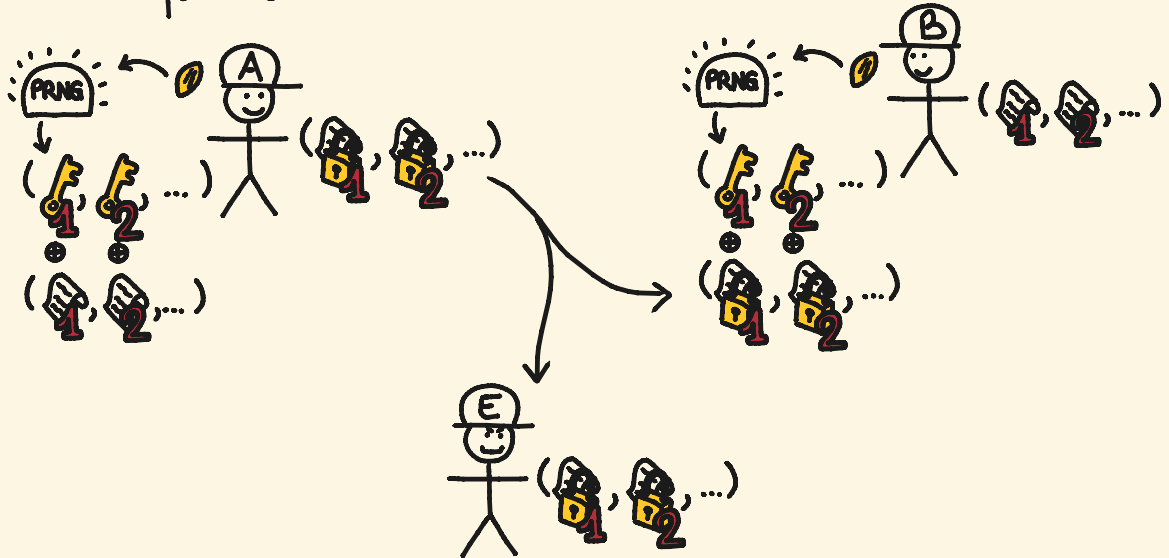


2. share a pseudorandom number generator



# STREAM CIPHER PROTOCOL (2)

3. send a stream of encrypted messages through a public channel



# STREAM CIPHER PROTOCOL COINDUCTIVELY

②

seedGen<sup>(0)</sup> = do  
| rand() → s  
| setSeed(s) ~> ()  
| return()

seedGen<sup>(+0)</sup> = do  
| return()

seedGen<sup>(++)</sup> = seedGen<sup>(+)</sup>



alice(m)<sup>(0)</sup> = do  
| getSeedA() ~> (s)  
| prng(s) → (s', k)  
| return(s', m ⊕ k)

alice(s, m)<sup>(+0)</sup> = do  
| prng(s) → (s', k)  
| return(s', m ⊕ k)

alice(s, m)<sup>(++)</sup> = alice(s, m)<sup>(+)</sup>

# STREAM CIPHER PROTOCOL COINDUCTIVELY



$\text{bob}(m)^{\circ} = \text{do}$   
|  $\text{getSeedB}() \rightsquigarrow (s)$   
|  $\text{prng}(s) \rightarrow (s', k)$   
|  $\text{return}(s', m \oplus k)$

$\text{bob}(s, m)^{\circ\circ} = \text{do}$   
|  $\text{prng}(s) \rightarrow (s', k)$   
|  $\text{return}(s', m \oplus k)$

$\text{bob}(s, m)^{++} = \text{bob}(s, m)^{+}$



$\text{eve}(m)^{\circ} = \text{do}$   
|  $\text{return}(m)$

$\text{eve}(m)^{+} = \text{eve}(m)$

# OUTLINE

- [ • effectful categories ]
- effectful streams
- effectful trace semantics
- causal processes

# COMPUTATIONS WITH EFFECTS

- Stochastic effects: the distribution monad

$$\mathcal{D}: \text{Set} \rightarrow \text{Set}$$

$$\mathcal{D}(A) := \{ \sigma : A \rightarrow [0,1] \mid \text{supp } \sigma \text{ is finite} \wedge \sum_{a \in A} \sigma(a) = 1 \}$$

- Global state: the state promonad

$$\text{St}: \mathcal{C}^{\text{op}} \times \mathcal{C} \rightarrow \text{Set}$$

$$\text{St}(A, B) := \mathcal{C}(S \otimes A, S \otimes B)$$



# PREMONOIDAL CATEGORIES

Some effects do not interchange.

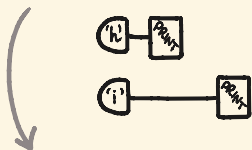
`printHI() = do`

`print('h') ~> ()`  
`print('i') ~> ()`  
`return ()`

$\neq$

`printIH() = do`

`print('i') ~> ()`  
`print('h') ~> ()`  
`return ()`



prints "hi"

$\neq$



prints "ih"

ex state premonads, IO monad



# STREAM CIPHER PROTOCOL (AGAIN)

seedGen( $\cdot$ )<sup>0</sup> = do

④

| rand( $\cdot$ )  $\rightarrow$   $s$   
| setSeed( $s$ )  $\rightsquigarrow$  ( $\cdot$ )  
| return( $\cdot$ )

seedGen( $\cdot$ )<sup>+0</sup> = do

| return( $\cdot$ )

seedGen<sup>++</sup> = seedGen<sup>+</sup>

alice( $m$ )<sup>0</sup> = do



| getSeedA( $\cdot$ )  $\rightsquigarrow$  ( $s$ )  
| prng( $s$ )  $\rightarrow$  ( $s'$ ,  $k$ )  
| return( $s'$ ,  $m \oplus k$ )

alice( $s$ ,  $m$ )<sup>+0</sup> = do

| prng( $s$ )  $\rightarrow$  ( $s'$ ,  $k$ )  
| return( $s'$ ,  $m \oplus k$ )

alice( $s$ ,  $m$ )<sup>++</sup> = alice( $s$ ,  $m$ )<sup>+</sup>

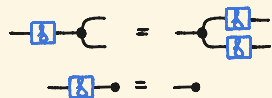


eve( $m$ )<sup>0</sup> = do  
| return( $m$ )

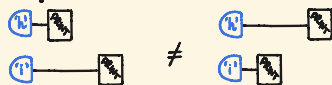
eve( $m$ )<sup>+</sup> = eve( $m$ )

# EFFECTFUL COPY-DISCARD CATEGORIES

Values can be copied and discarded  
(cartesian)

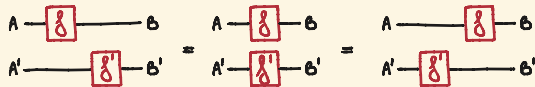


Effectful computations may have global effects  
(premonoidal)



$\mathcal{V} \hookrightarrow \mathcal{L} \hookrightarrow \mathcal{C}$

Local computations interchange  
(monoidal)



ex (Set, Stoch, State)

# OUTLINE

- effectful categories
- [ • effectful streams ]
- effectful trace semantics
- causal processes

# EFFECTFUL STREAMS

An effectful stream  $F: A \rightarrow B$  on  $(\mathcal{U}, \mathcal{L}, \mathcal{C})$  is

- a memory  $M_g \in \mathcal{L}$
- a first action  $g^0: A^0 \rightarrow M_g \otimes B^0$  in  $\mathcal{C}$
- the rest of the action  $F^+: M_g \cdot A^+ \rightarrow B^+$

$$A \text{---} \boxed{F} \text{---} B = A^0 \text{---} \boxed{g^0} \text{---} \overset{M_g}{B^0 A^+} \text{---} \boxed{F^+} \text{---} B^+$$

quotiented by the equivalence relation generated by

$$\begin{cases} \boxed{g^0}; (\pi \otimes \mathbb{1}) = g^0 \\ \boxed{F^+} = \pi \cdot g^+ \end{cases} \quad \text{for } \pi: M_g \rightarrow M_g \text{ in } \mathcal{L}$$

$$\boxed{g^0} \text{---} \boxed{F^+} = \boxed{g^0} \text{---} \boxed{\pi} \text{---} \boxed{F^+} \sim \boxed{g^0} \text{---} \boxed{\pi} \text{---} \boxed{F^+} = \boxed{g^0} \text{---} \boxed{g^+}$$

# COMPOSITIONAL STRUCTURE OF STREAMS

## THEOREM

Effectful streams form an effectful category  $\text{Stream}$ .

- composition and monoidal actions are defined coinductively:  
for  $f: N_f \cdot A \rightarrow B$  and  $g: N_g \cdot B \rightarrow C$ ,

$$\left\{ \begin{array}{l} (f;_N g)^\circ := \begin{array}{c} N_g \\ \text{---} \\ N_f \\ \text{---} \\ A^\circ \end{array} \begin{array}{c} \text{---} \\ \text{---} \\ \text{---} \\ \text{---} \end{array} \begin{array}{c} M_g \\ \text{---} \\ M_f \\ \text{---} \\ C^\circ \end{array} \end{array} \right.$$

$$(f;_N g)^+ := f^+;_M g^+$$

$$\left\{ \begin{array}{l} (X \otimes_N f)^\circ := \begin{array}{c} N_f \\ \text{---} \\ A^\circ \\ \text{---} \\ X^\circ \end{array} \begin{array}{c} \text{---} \\ \text{---} \\ \text{---} \\ \text{---} \end{array} \begin{array}{c} M_f \\ \text{---} \\ B^\circ \\ \text{---} \\ X^\circ \end{array} \end{array} \right.$$

$$(X \otimes_N f)^+ := X^+ \otimes_M f^+$$

# SEMANTICS FOR THE STREAM CIPHER PROTOCOL

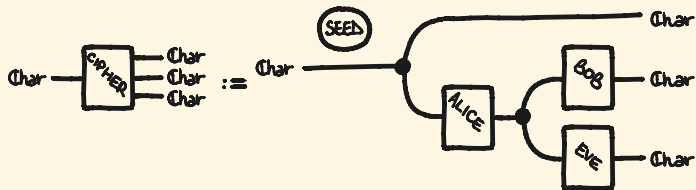
Fix two finite sets Char, Seed  
and take the effectful copy-discard category

(Set, Stoch, SeedStoch)

SeedStoch is the Kleisli category of the monad  
that adds the global state Seed  $\times$  Seed :

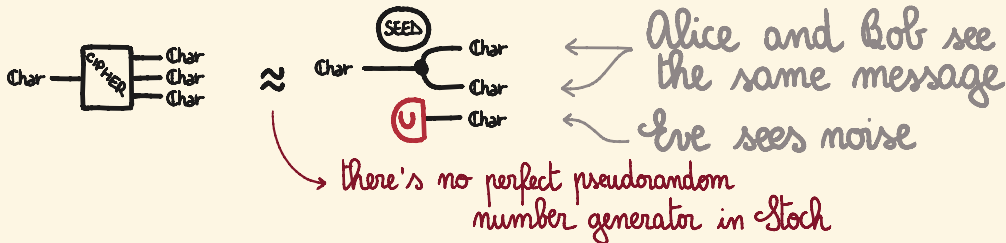
SeedStoch(A, B) := Stoch(Seed  $\times$  Seed  $\times$  A, Seed  $\times$  Seed  $\times$  B)

# SEMANTICS FOR THE STREAM CIPHER PROTOCOL



## THEOREM

The stream cipher protocol is secure.



[cf. Broadbent & Karvonen 2023]

# OUTLINE

- effectful categories
- effectful streams
- [ • effectful trace semantics ]
- causal processes



# EFFECTFUL MEALY MACHINES

A Mealy machine  $(f, S, s_0) : A \rightarrow B$  in  $(\mathcal{U}, \mathcal{L}, \mathcal{C})$   
is a morphism

$$f : S \otimes A \rightarrow S \otimes B$$



with an initial state

$$s_0 : I \rightarrow S$$



A morphism of Mealy machines  $u : (f, S, s_0) \rightarrow (g, T, t_0)$   
is a value morphism  $u : S \rightarrow T$  in  $\mathcal{U}$

such that

$$S \begin{array}{c} \text{---} \\ \text{---} \end{array} \boxed{f} \begin{array}{c} \text{---} \\ \text{---} \end{array} \boxed{u} \begin{array}{c} T \\ B \end{array} = S \begin{array}{c} \text{---} \\ \text{---} \end{array} \boxed{u} \boxed{g} \begin{array}{c} T \\ B \end{array}$$

$$\begin{array}{c} \text{---} \\ \text{---} \end{array} \boxed{s_0} \begin{array}{c} \text{---} \\ \text{---} \end{array} \boxed{u} \begin{array}{c} T \\ \end{array} = \begin{array}{c} \text{---} \\ \text{---} \end{array} \boxed{t_0} \begin{array}{c} T \\ \end{array}$$

[cf. Katis, Sabadini, Walters 1997 ; EDL, Gianola, Román, Sabadini, Sobociński 2022]

# EFFECTFUL CATEGORY OF MEALY MACHINES

Mealy is an effectful category where

- objects are the objects of  $\mathcal{C}$
- morphisms  $(f, S, \rho_0): A \rightarrow B$  are Mealy machines quotiented by **value** isomorphisms  $u: S \xrightarrow{\cong} T$

$$\begin{array}{c} S \\ \text{---} \\ \text{A} \end{array} \text{---} \boxed{f} \text{---} \boxed{u} \text{---} \begin{array}{c} T \\ \text{---} \\ \text{B} \end{array} = \begin{array}{c} S \\ \text{---} \\ \text{A} \end{array} \text{---} \boxed{u} \text{---} \boxed{g} \text{---} \begin{array}{c} T \\ \text{---} \\ \text{B} \end{array}$$

$$\begin{array}{c} \rho_0 \\ \text{---} \\ \text{A} \end{array} \text{---} \boxed{u} \text{---} T = \begin{array}{c} \tau_0 \\ \text{---} \\ \text{A} \end{array} \text{---} T$$

- composition tensors the state spaces

$$\begin{array}{c} S \\ \text{---} \\ \text{T} \\ \text{---} \\ \text{A} \end{array} \text{---} \boxed{f} \text{---} \boxed{g} \text{---} \begin{array}{c} S \\ \text{---} \\ \text{T} \\ \text{---} \\ \text{C} \end{array} = \begin{array}{c} \rho_0 \\ \text{---} \\ \text{A} \end{array} \text{---} S \quad \begin{array}{c} \tau_0 \\ \text{---} \\ \text{A} \end{array} \text{---} T$$

# COMPOSITIONAL TRACE SEMANTICS

## THEOREM

There is an effectful functor

$\text{Tr} : \text{Mealy} \rightarrow \text{Stream}$

$A \mapsto (A) = (A, A, \dots)$

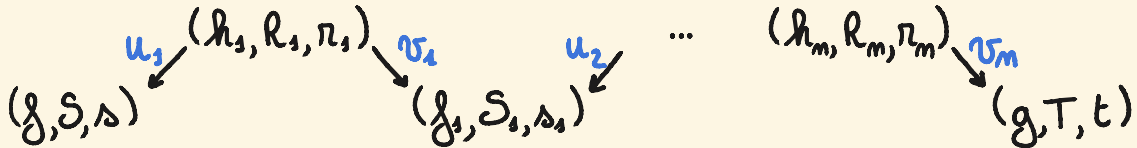
$\begin{array}{c} S \\ A \end{array} \text{---} \boxed{\delta} \text{---} \begin{array}{c} S \\ B \end{array} \mapsto A \text{---} \overset{A}{\curvearrowright} \boxed{\delta} \text{---} B \text{---} (A) \text{---} \boxed{(\delta)} \text{---} (B)$

$= A \text{---} \overset{A}{\curvearrowright} \boxed{\delta} \text{---} B \text{---} A \text{---} \boxed{\delta} \text{---} B \text{---} A \text{---} \boxed{\delta} \text{---} B \dots$

$\leadsto$  in Rel these traces coincide with the classical traces

# BISIMULATION

Two effectful Mealy machines  $(g, S, s), (g, T, t) : A \rightarrow B$  are bisimilar if they belong to the same connected component in  $\text{Mealy}(A, B)$ :



## THEOREM

For Mealy machines in  $(\mathcal{V}, \mathcal{L}, \mathcal{C})$ ,  
bisimulation implies trace equivalence.

PROOF: By coinduction.  $\square$

# COALGEBRAIC BISIMULATION

## PROPOSITION

When  $\mathcal{C} = \text{Kl}(M)$ , for a commutative monad preserving weak pullbacks, then  $(f, S, s)$  and  $(g, T, t)$  are bisimilar iff they have the same bisimulation quotient, i.e. there is  $(h, Q, q)$  with morphisms

$$(f, S, s) \xrightarrow{u} (h, Q, q) \xleftarrow{v} (g, T, t) .$$

## EXAMPLES

- Set
- Rel
- pStoch

# OUTLINE

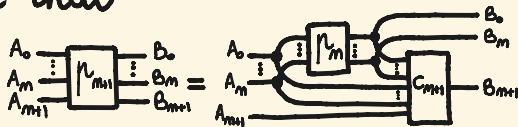
- effectful categories
- effectful streams
- effectful trace semantics
- [ • causal processes ]

# CAUSAL PROCESSES

A causal process  $\mu: A \rightarrow B$  in a copy-discard category  $\mathcal{C}$  is a family of morphisms

$$\mu_m : A_0 \otimes \dots \otimes A_m \rightarrow B_0 \otimes \dots \otimes B_m$$

such that



for some  $c_{m+1} : B_0 \otimes \dots \otimes B_m \otimes A_0 \otimes \dots \otimes A_m \otimes A_{m+1} \rightarrow B_{m+1}$

## THEOREM

causal processes form a monoidal category Proc when  $\mathcal{C}$  has quasi-total conditionals.

[cf. Ramey 1958 ; Springer & Katsumata 2019]

# CAUSAL PROCESSES ARE STREAMS

## THEOREM

Consider  $(\text{func } \mathcal{C}, \text{tot } \mathcal{C}, \mathcal{C})$ .

If  $\mathcal{C}$  has quasi-total conditionals and ranges,  
 $\text{Proc} \approx \text{Stream}$ .

## EXAMPLES

- Set
- Rel
- pStoch
- Par
- Stoch



# TRACES ARE EFFECTFUL TRACES

compute the traces of a Mealy machine

$$(f, S, s) : A \rightarrow B$$

in some known cases.

$(b_0, \dots, b_m)$  is a trace of  $(a_0, \dots, a_m)$

Set if  $s_0 = s$  and  $\forall k \leq m$   $(s_{k+1}, b_k) = f(s_k, a_k)$

Rel if  $\exists (s_0, \dots, s_{m+1})$   $s_0 \in S$   
and  $\forall k \leq m$   $(s_{k+1}, b_k) \in f(s_k, a_k)$

stoch with probability  $\sum_{(s_0, \dots, s_{m+1})} s(s_0 | *) \cdot \prod_{k \leq m} f(s_{k+1}, b_k | s_k, a_k)$

# SUMMARY

- formal compositional semantics for effectful stream computations
- trace equivalence and bisimulation of effectful Mealy machines
- characterisation as causal stream processes

# FUTURE WORK

- coinduction up-to dinaturality

$$\boxed{g^0} \text{---} \boxed{f^+} = \boxed{g^0} \text{---} \boxed{\eta} \text{---} \boxed{f^+} \sim \boxed{g^0} \text{---} \boxed{\eta} \text{---} \boxed{f^+} = \boxed{g^0} \text{---} \boxed{g^+}$$

- Rel with explicit failure
- equality in StC implies bisimulation

$$\boxed{\delta} \text{---} \boxed{\delta} = \boxed{g} \text{---} \boxed{g} \Rightarrow \boxed{\delta} \text{---} \boxed{\delta} \approx \boxed{g} \text{---} \boxed{g}$$

↕ ? ↕

- distance instead of equivalence relation for security


$$\text{Seed} \text{---} \boxed{U} \text{---} \boxed{PR} \text{---} \text{Char} \approx \text{Seed} \text{---} \boxed{U} \text{---} \text{Char}$$

ε ?


# APPENDIX

# SEMANTICS FOR THE STREAM CIPHER PROTOCOL

Semantics for *values* and *local* computations.

$\llbracket - \oplus - \rrbracket := \text{xor} : \text{Char} \times \text{Char} \rightarrow \text{Char} \rightsquigarrow \text{bitwise xor}$   
 in *Set*

$\llbracket \text{rand} \rrbracket := \text{unif} : 1 \rightarrow \mathcal{D}(\text{Seed}) \rightsquigarrow \text{uniform distribution}$   
 in *Stoch*

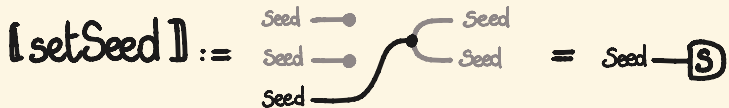
$\llbracket \text{prng} \rrbracket : \text{Seed} \rightarrow \text{Seed} \times \mathcal{D}(\text{Char}) \rightsquigarrow \text{use the seed to generate a key}$   
 in *Stoch*

# SEMANTICS FOR THE STREAM CIPHER PROTOCOL

Semantics for effectful computations.

$\llbracket \text{setSeed} \rrbracket : \text{Seed}^3 \rightarrow \text{Seed}^2$   
 $\text{Seed} \rightsquigarrow 1$

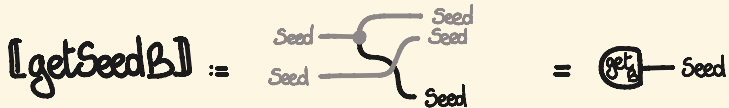
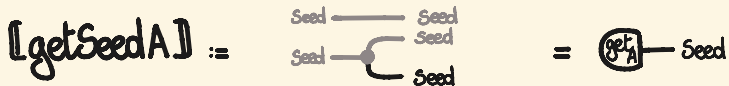
$\rightsquigarrow$  copy the seed to the global state



in SeedStock

$\llbracket \text{getSeedA} \rrbracket, \llbracket \text{getSeedB} \rrbracket : \text{Seed}^2 \rightarrow \text{Seed}^3$   
 $1 \rightsquigarrow \text{Seed}$

$\rightsquigarrow$  alice and bob get their seeds



# SEMANTICS FOR THE STREAM CIPHER PROTOCOL

•  $\text{seedGen} = \textcircled{\text{SEED}} : \mathbb{I} \rightarrow \mathbb{I}$  in Stream

$$\text{seedGen}^{\circ} := \begin{array}{c} \text{Seed} \quad \bullet \\ \text{Seed} \quad \bullet \\ \textcircled{\text{U}} \quad \text{---} \quad \bullet \\ \text{Seed} \quad \text{---} \quad \bullet \\ \text{Seed} \quad \text{---} \quad \bullet \end{array} = \textcircled{\text{U}} \text{---} \textcircled{\text{S}}$$

$$\text{seedGen}^{+\circ} := \begin{array}{c} \text{Seed} \text{---} \text{Seed} \\ \text{Seed} \text{---} \text{Seed} \end{array} = \square$$

$$\text{seedGen}^{++} = \text{seedGen}^{+}$$

•  $\text{eve} = \text{Char} \text{---} \boxed{\text{EVE}} \text{---} \text{Char} : \text{Char} \rightarrow \text{Char}$  in Stream

$$\text{eve}^{\circ} := \begin{array}{c} \text{Seed} \text{---} \text{Seed} \\ \text{Seed} \text{---} \text{Seed} \\ \text{Char} \text{---} \text{Char} \end{array} = \text{Char} \text{---} \text{Char}$$

$$\text{eve}^{+} = \text{eve}$$

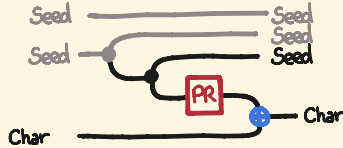
# SEMANTICS FOR THE STREAM CIPHER PROTOCOL

•  $\text{alice} = \text{Char} \rightarrow \boxed{\text{Alice}} \rightarrow \text{Char} : \text{Char} \rightarrow \text{Char}$  in Stream

•  $\text{bob} = \text{Char} \rightarrow \boxed{\text{Bob}} \rightarrow \text{Char} : \text{Char} \rightarrow \text{Char}$  in Stream

$\text{alice}^\circ$

$:=$

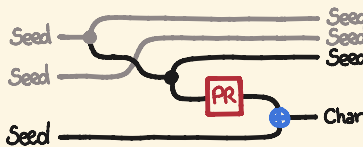


$=$



$\text{bob}^\circ$

$:=$



$=$



$\text{alice}^{+\circ} = \text{bob}^{+\circ}$

$:=$



$=$



$\text{alice}^{++} = \text{alice}^+$

$\text{bob}^{++} = \text{bob}^+$



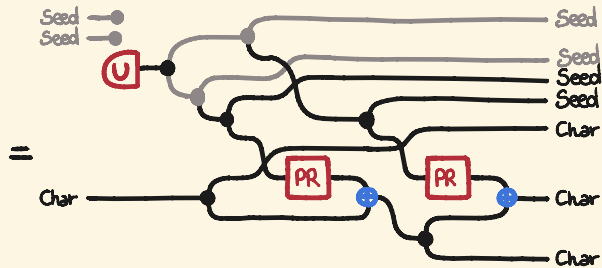
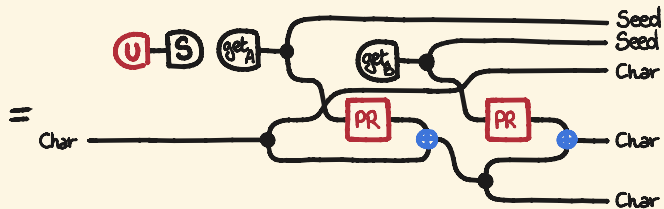
# STREAM CIPHER IS SECURE

Proceed by coinduction.

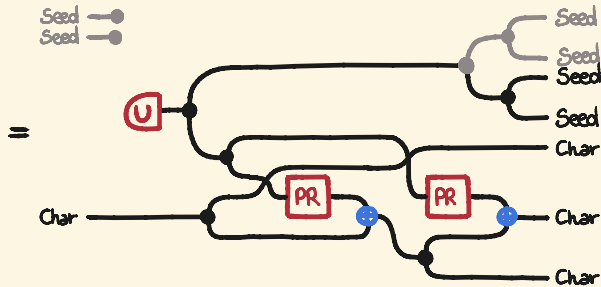
cipher<sup>o</sup>

```
cipher(m)o = do
= [ rand() → s
  setSeed(s) ~> ()
  getSeedA() ~> sA
  prng(sA) → (s'A, kA)
  getSeedB() ~> sB
  prng(sB) → (s'B, kB)
  return(m, s'A, m ⊕ kA ⊕ kB, s'B, m ⊕ kA)
```

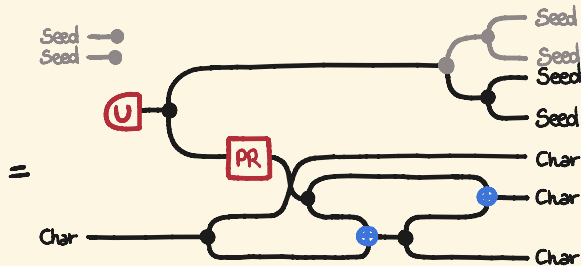
# STREAM CIPHER IS SECURE



# STREAM CIPHER IS SECURE

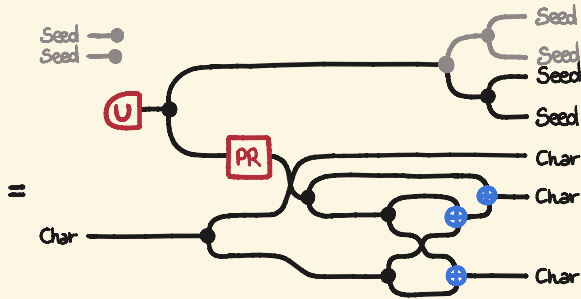


*by associativity of copy*

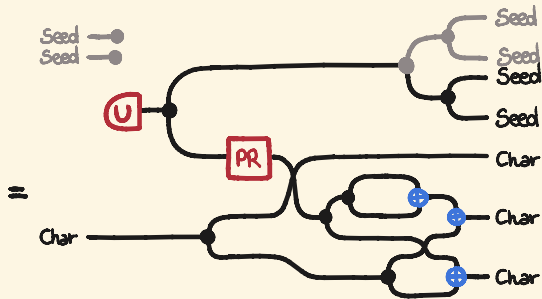


*pseudorandom is deterministic*

# STREAM CIPHER IS SECURE

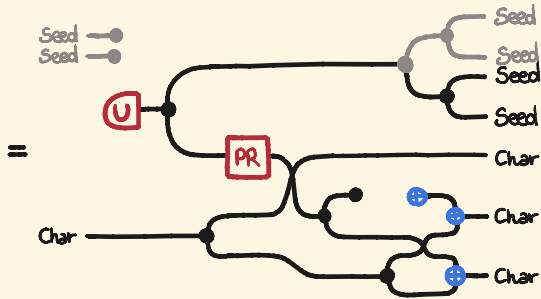


xor is deterministic

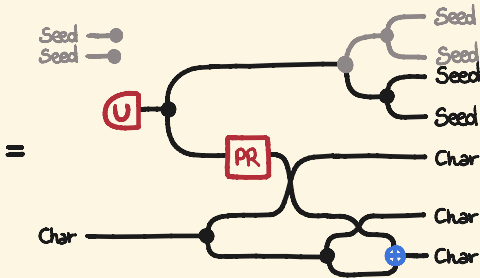


by associativity of copy  
and xor

# STREAM CIPHER IS SECURE

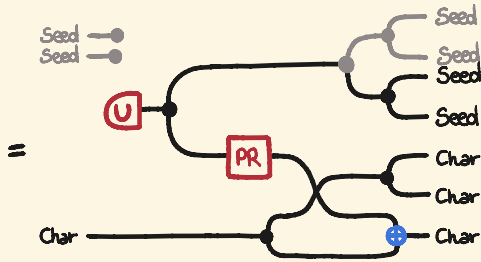


xor is nilpotent

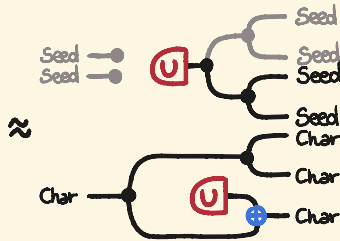


by unitality of copy  
and xor

# STREAM CIPHER IS SECURE

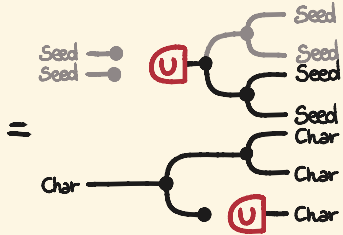


by associativity of copy

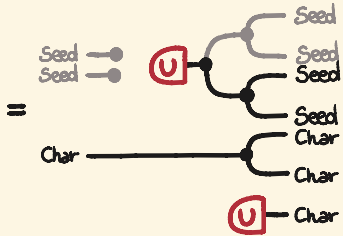


by assumption

# STREAM CIPHER IS SECURE



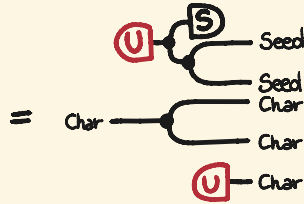
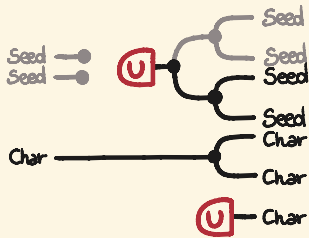
the uniform distribution is a  
Sweedler integral for xor



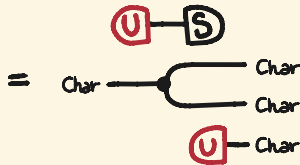
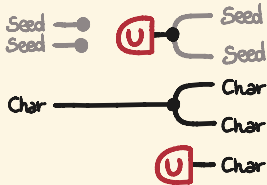
by unitality of copy

# STREAM CIPHER IS SECURE

cipher<sup>o</sup> ≈



secure<sup>o</sup> :=

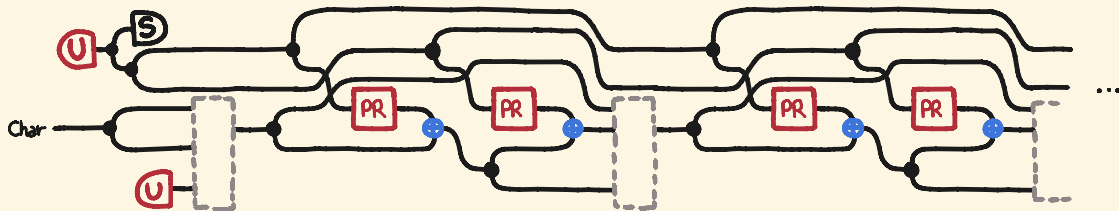




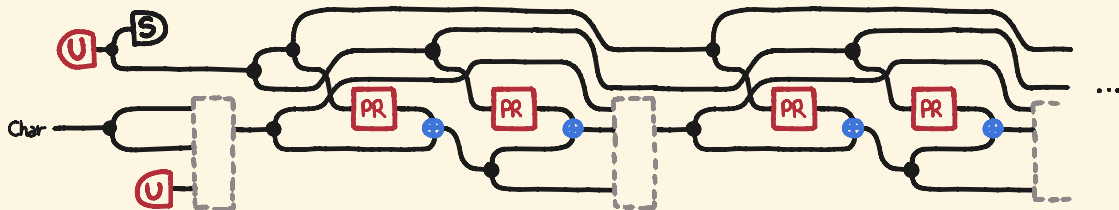
# STREAM CIPHER IS SECURE

cipher

=

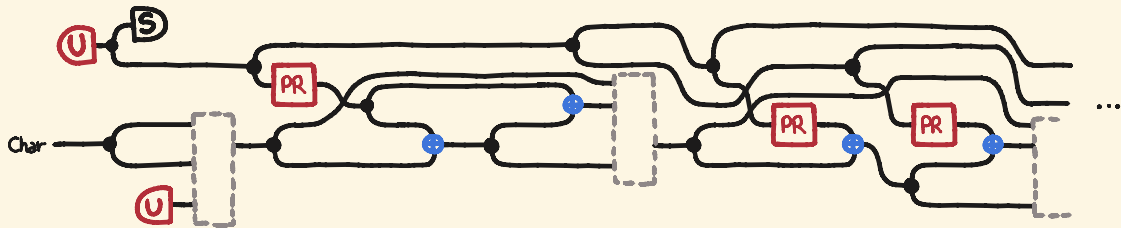


= (by sliding)

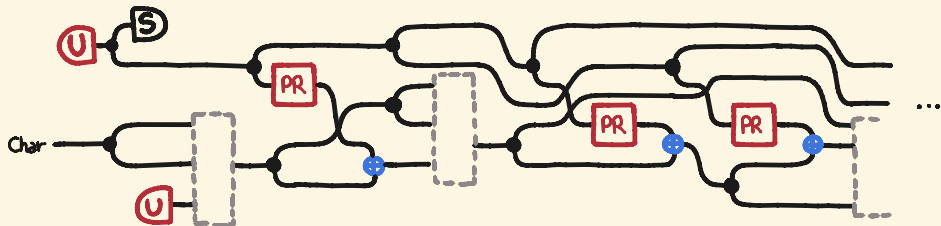


# STREAM CIPHER IS SECURE

= (pseudorandom is deterministic)

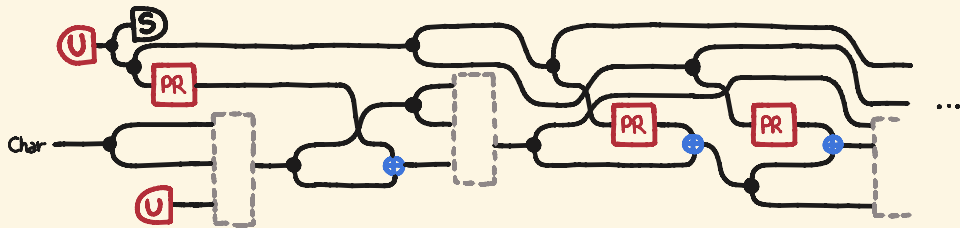


= (xor is deterministic and nilpotent)

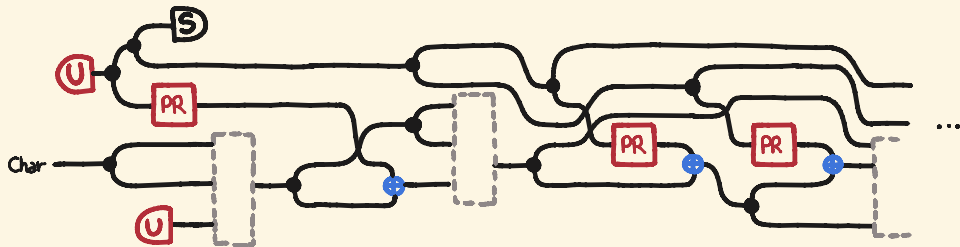


# STREAM CIPHER IS SECURE

= (by sliding)

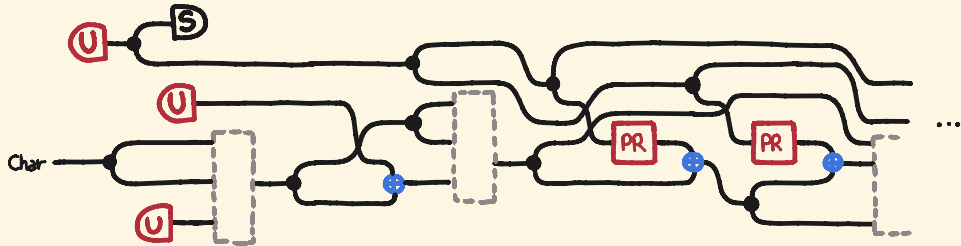


= (by associativity)

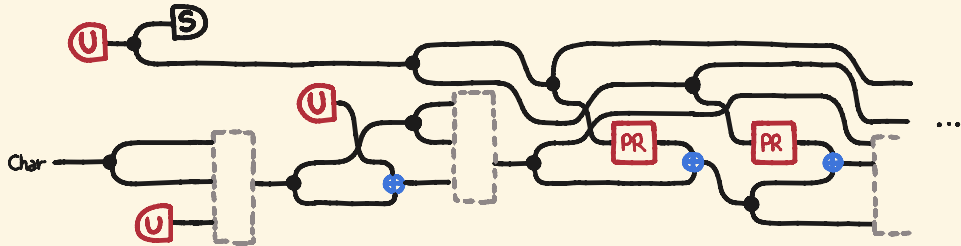


# STREAM CIPHER IS SECURE

$\approx$  (by assumption on pseudorandom)

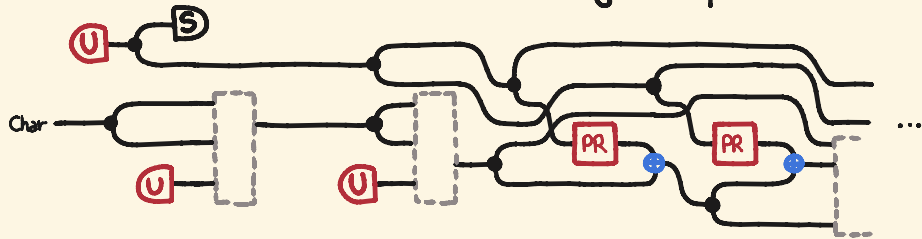


= (by sliding)

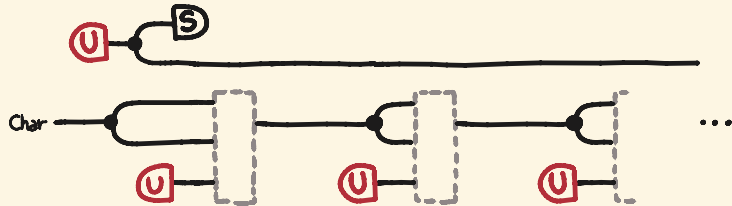


# STREAM CIPHER IS SECURE

= (unif is a Sweedler integral for xor)



$\approx$  (by coinduction)

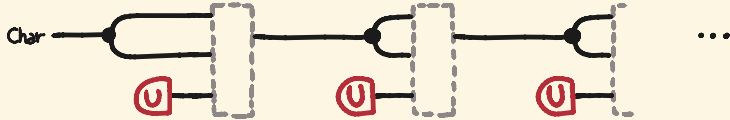


# STREAM CIPHER IS SECURE

= (by coinduction)



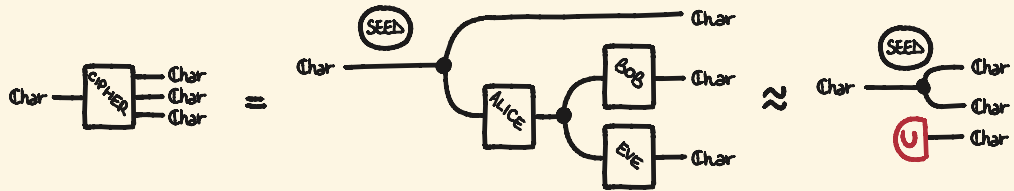
= (by unitality)



= secure

# STREAM CIPHER IS SECURE

We have shown



using sliding and coinduction.