

Padova

9 April 2024

EFFECTFUL TRACE SEMANTICS VIA EFFECTFUL STREAMS

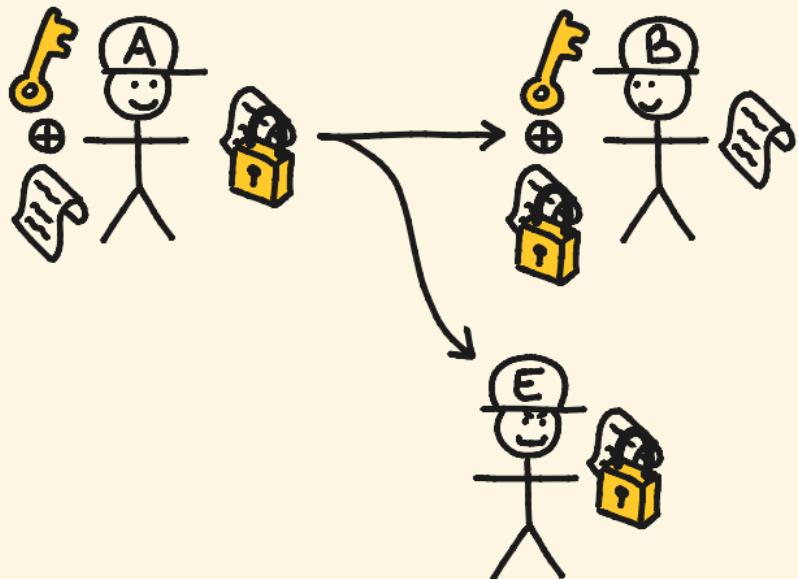
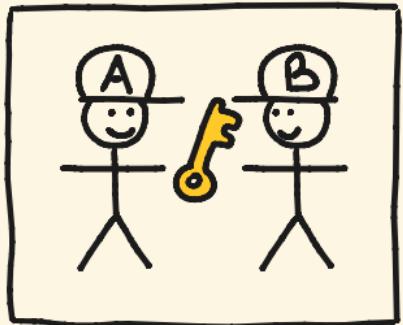
Filippo Bonchi
Università di Pisa

Elena Di Stefano
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

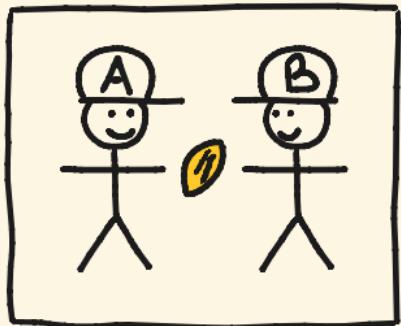
Sending n messages securely requires n private keys
↳ not very useful

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



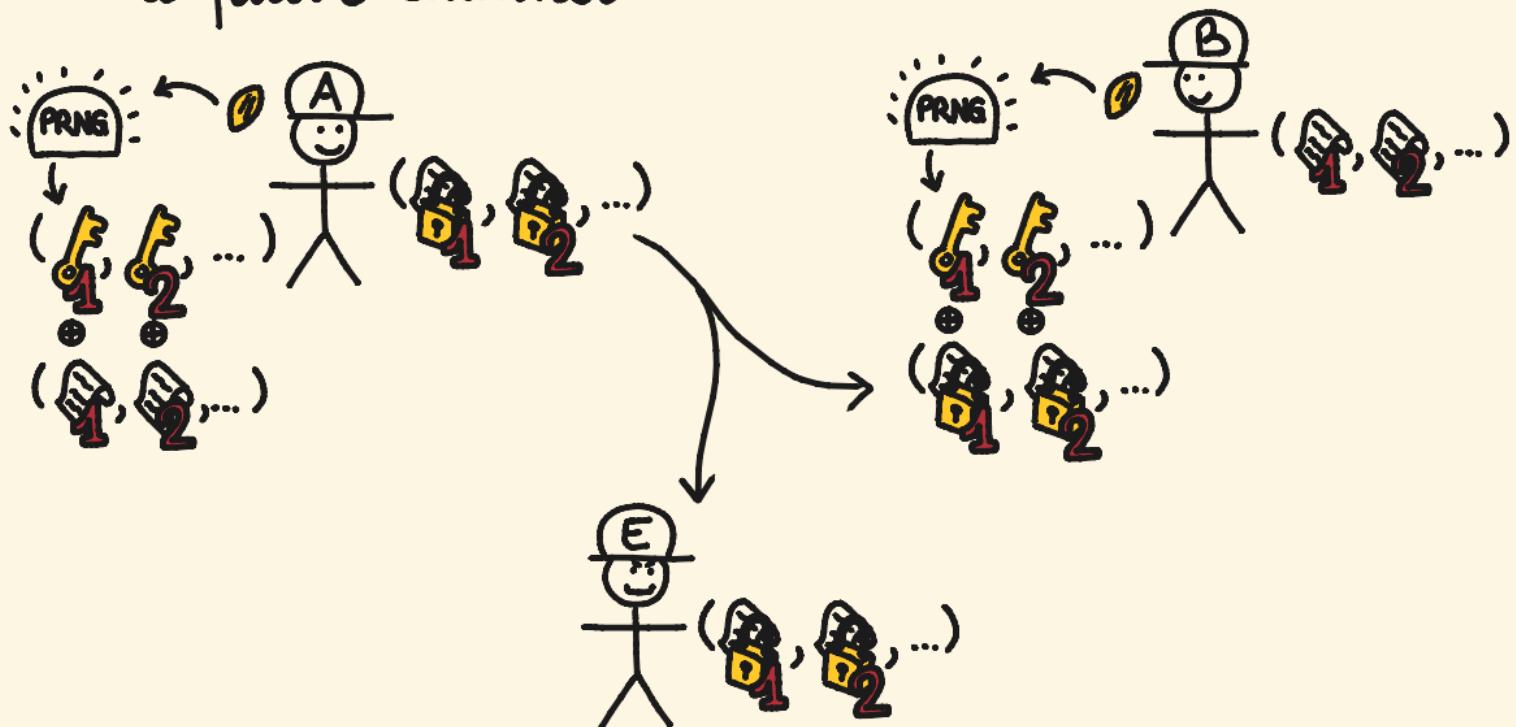
STREAM CIPHER PROTOCOL (1)

1. share a seed through a secure channel
2. share a pseudorandom number generator



STREAM CIPHER PROTOCOL (2)

- send a stream of encrypted messages through a public channel



STREAM CIPHER PROTOCOL CONDUCTIVELY

seedGen⁰ = do

rand() → s

setSeed(s) ↳ ()

return()

alice(m)⁰ = do

getSeedA() ↳ (s)

prng(s) → (s', k)

return(s', m ⊕ k)

seedGen⁺⁰ = do

return()

seedGen⁺⁺ = seedGen⁺



alice(s, m)⁺⁰ = do

prng(s) → (s', k)

return(s', m ⊕ k)

alice(s, m)⁺⁺ = alice(s, m)⁺

STREAM CIPHER PROTOCOL CONDUCTIVELY



$\text{bob}(m)^\circ = \text{do}$

$\begin{cases} \text{getSeedB()} \rightsquigarrow (s) \\ \text{prng}(s) \rightarrow (s', k) \\ \text{return } (s', m \oplus k) \end{cases}$

$\text{bob}(s, m)^{+0} = \text{do}$

$\begin{cases} \text{prng}(s) \rightarrow (s', k) \\ \text{return } (s', m \oplus k) \end{cases}$

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



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

$\begin{cases} \text{return } (m) \end{cases}$

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

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 ()`

`printIH() = do`

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

\neq



\neq



points "hi"

points "ih"

ex state promonads, IO monad

STREAM CIPHER PROTOCOL (AGAIN)

seedGen⁽⁰⁾ = do

rand() → s

setSeed(s) ↳ ()

return()

alice(m)⁰ = do

getSeedA() ↳ (s)

prng(s) → (s', k)

return(s', m ⊕ k)

eve(m)⁰ = do

return(m)

seedGen⁽⁺⁰⁾ = do

return()

seedGen⁺⁺ = seedGen⁺

alice(s, m)⁺⁰ = do

prng(s) → (s', k)

return(s', m ⊕ k)

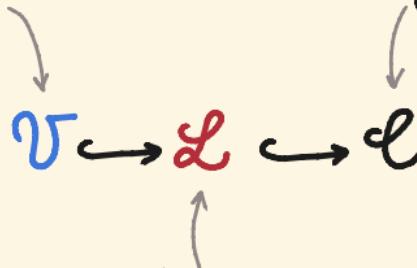
alice(s, m)⁺⁺ = alice(s, m)⁺

eve(m)⁺ = eve(m)

EFFECTFUL COPY-DISCARD CATEGORIES

Values can be copied and discarded (cartesian)

$$\begin{array}{c} \text{---} \square \curvearrowleft = \text{---} \square \\ \text{---} \square \bullet = \text{---} \end{array}$$



Effectful computations may have global effects (premonoidal)

$$\begin{array}{c} \text{---} \square \curvearrowright \\ \text{---} \square \bullet \end{array} \neq \begin{array}{c} \text{---} \square \bullet \\ \text{---} \square \curvearrowright \end{array}$$

local computations interchange (monoidal)

$$\begin{array}{ccc} A - \boxed{g} - B & = & A - \boxed{g} - B \\ A' - \boxed{g'} - B' & = & A' - \boxed{g'} - B' \\ \end{array} = \begin{array}{ccc} A - \boxed{g} - B & = & A - \boxed{g'} - B \\ A' - \boxed{g'} - B' & = & A' - \boxed{g} - B' \end{array}$$

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 $g^\circ: A^\circ \rightarrow M_g \otimes B^\circ$ in \mathcal{C}
- the rest of the action $f^+: M_g \cdot A^+ \rightarrow B^+$

$$A - \boxed{f} - B = A^\circ - \boxed{g^\circ} - B^\circ \xrightarrow{M_g} A^+ - \boxed{f^+} - B^+$$

quotiented by the equivalence relation generated by

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

$$-\boxed{g^\circ} - \boxed{f^+} - = -\boxed{g^\circ} - \boxed{\pi} - \boxed{f^+} - \sim -\boxed{g^\circ} - \boxed{\pi} - \boxed{f^+} - = -\boxed{g^\circ} - \boxed{g^+} -$$

COMPOSITIONAL STRUCTURE OF STREAMS

THEOREM

Effectful streams form an effectful category Stream.

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

$$\begin{cases} (F;_N g)^\circ := \begin{array}{c} Ng \\ \xrightarrow{\quad g \circ \quad} \\ A^\circ \end{array} \quad \begin{array}{c} Ng \\ \xrightarrow{\quad g \circ \quad} \\ B^\circ \end{array} \quad \begin{array}{c} Mg \\ \xrightarrow{\quad g \circ \quad} \\ C^\circ \end{array} \\ (F;_N g)^+ := F^+;_M g^+ \end{cases}$$

$$\begin{cases} (\mathbb{X} \otimes_N F)^\circ := \begin{array}{c} Ng \\ \xrightarrow{\quad g \circ \quad} \\ A^\circ \end{array} \quad \begin{array}{c} Ng \\ \xrightarrow{\quad g \circ \quad} \\ B^\circ \end{array} \\ \qquad \qquad \qquad \xrightarrow{\quad x^\circ \quad} \qquad \qquad \qquad x^\circ \\ (\mathbb{X} \otimes_N F)^+ := \mathbb{X}^+ \otimes_M F^+ \end{cases}$$

SEMANTICS FOR THE STREAM CIPHER PROTOCOL

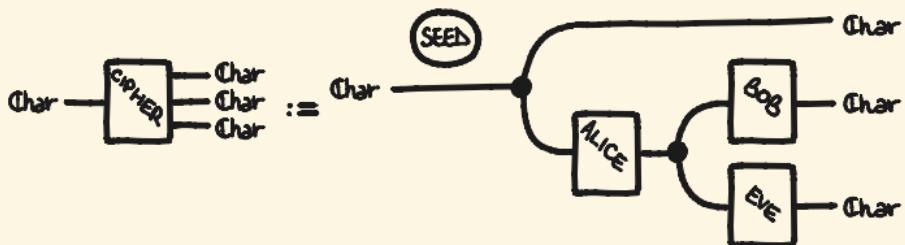
Fix two finite sets Char, Seed
and take the effectful copy-discard category

(cSet, cStoch, cSeedStoch)

cSeedStoch is the Kleisli category of the promonad
that adds the global state $\text{Seed} \times \text{Seed}$:

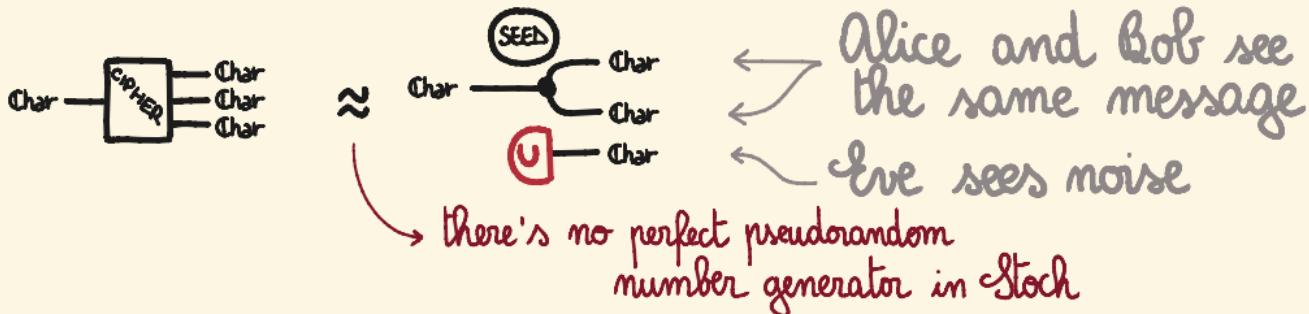
$\text{cSeedStoch}(A, B) := \text{cStoch}(\text{Seed} \times \text{Seed} \times A, \text{Seed} \times \text{Seed} \times B)$

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

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

$$S_A = \boxed{f} = S_B$$

with an initial state

$$s_0 : I \rightarrow S$$

$$\textcircled{A} - S$$

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

$$S_A = \boxed{f} \xrightarrow{u} T_B = S_A = \boxed{g} \xrightarrow{T} T_B$$

$$\textcircled{A} \xrightarrow{u} T = \textcircled{t_0} - T$$

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

EFFECTFUL CATEGORY OF MEALY MACHINES

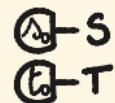
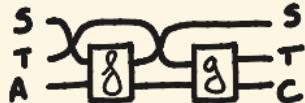
Mealy is an effectful category where

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

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

$$\begin{array}{c} \text{---} \\ \text{---} \end{array} \xrightarrow{\quad u \quad} \begin{array}{c} T \\ \text{---} \end{array} = \begin{array}{c} \text{---} \\ \text{---} \end{array} \xrightarrow{\quad t_0 \quad} T$$

- composition tensors the state spaces



COMPOSITIONAL TRACE SEMANTICS

THEOREM

There is an effectful functor

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

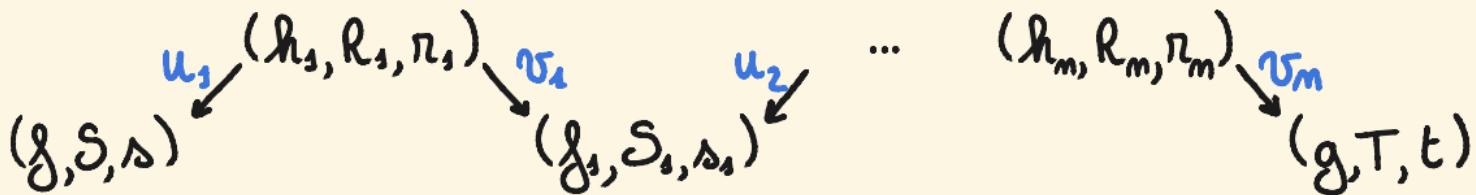
$$A \mapsto (A) = (A, A, \dots)$$

$$\begin{aligned} S_A &= \boxed{A} \xrightarrow{S_B} \boxed{B} \mapsto A \xrightarrow{\quad} \boxed{A} \xrightarrow{S_B} \boxed{(A)} \xrightarrow{S_B} \boxed{(B)} \\ &= A \xrightarrow{\quad} \boxed{A} \xrightarrow{S_B} \boxed{B} \xrightarrow{A} \boxed{A} \xrightarrow{S_B} \boxed{B} \xrightarrow{A} \boxed{A} \xrightarrow{S_B} \boxed{B} \dots \end{aligned}$$

→ in Rel these traces coincide with the classical traces

BISIMULATION

Two effectful Mealy machines $(f, S, s), (g, T, t) : A \rightarrow B$ 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
- probstoch

OUTLINE

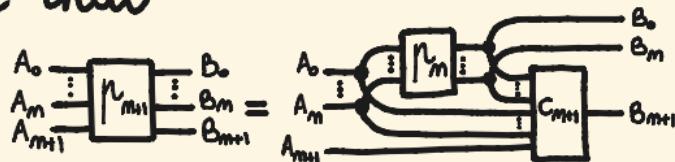
- effectful categories
- effectful streams
- effectful trace semantics
- [• causal processes]

CAUSAL PROCESSES

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

$$p_m : A_0 \otimes \cdots \otimes A_m \rightarrow B_0 \otimes \cdots \otimes B_m$$

such that



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

THEOREM

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

[cf. Ramey 1958; Sprunger & Katsumata 2019]

CAUSAL PROCESSES ARE STREAMS

THEOREM

Consider $(\text{funcl}, \text{tot cl}, cl)$.

If cl has quasi-total conditionals and ranges,
 $\text{Proc} \simeq \text{Stream}$.

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 $\forall k \leq m \quad (s_{k+1}, b_k) = f(s_k, a_k)$

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

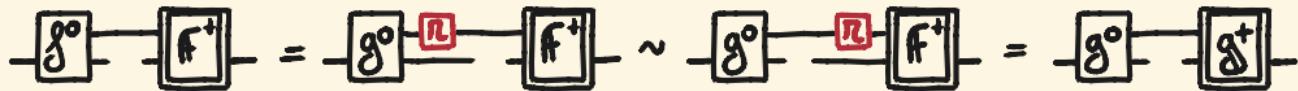
pStoch with probability $\sum_{(s_0, \dots, s_{m+1})} s(s_0 | *) \cdot \prod_{k \leq m} f(s_{k+1}, b_k | s_k, a_k)$

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



- Rel with explicit failure
- equality in cStL implies bisimulation



- distance instead of equivalence relation for security



APPENDIX

SEMANTICS FOR THE STREAM CIPHER PROTOCOL

Semantics for values and local computations.

$[- \oplus -] := \text{xor} : \text{Char} \times \text{Char} \rightarrow \text{Char}$ \rightsquigarrow bitwise xor



$[\text{rand}] := \text{unif} : 1 \rightarrow \mathcal{D}(\text{Seed})$ \rightsquigarrow uniform distribution



$[\text{prng}] : \text{Seed} \rightarrow \text{Seed} \times \mathcal{D}(\text{Char})$



\rightsquigarrow use the seed to generate a key

SEMANTICS FOR THE STREAM CIPHER PROTOCOL

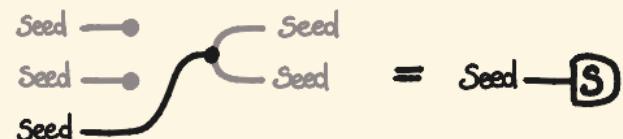
Semantics for effectful computations.

[setSeed] : Seed³ → Seed²

Seed ↼ 1

↝ copy the seed to
the global state

[setSeed] :=



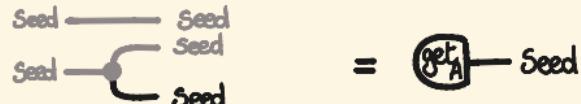
in `SeedStock`

[getSeedA], [getSeedB] : Seed² → Seed³

1 ↼ Seed

↝ alice and bob
get their seeds

[getSeedA] :=



[getSeedB] :=



SEMANTICS FOR THE STREAM CIPHER PROTOCOL

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

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

$$\text{seedGen}^{+0} := \begin{array}{c} \text{Seed} \\ \text{Seed} \end{array} \xrightarrow{\quad} \begin{array}{c} \text{Seed} \\ \text{Seed} \end{array} = \square$$

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

- $\text{eve} = \text{Char} \xrightarrow{\text{eve}} \text{Char} : \text{Char} \rightarrow \text{Char}$ in Stream

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

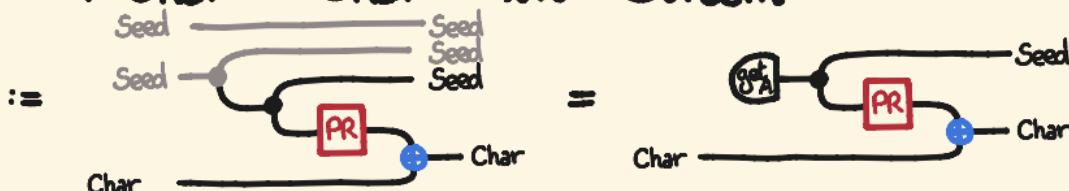
$$\text{eve}^+ = \text{eve}$$

SEMANTICS FOR THE STREAM CIPHER PROTOCOL

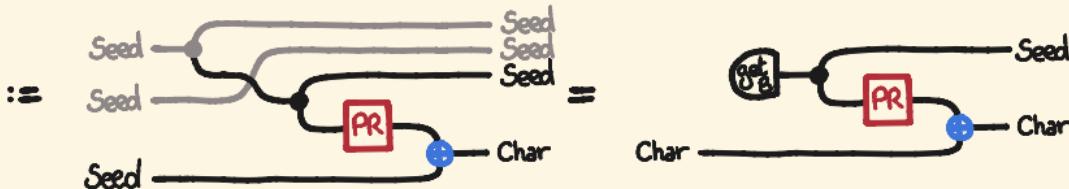
• $\text{alice} = \text{Char} \xrightarrow{\text{ALICE}} \text{Char} : \text{Char} \rightarrow \text{Char}$ in Stream

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

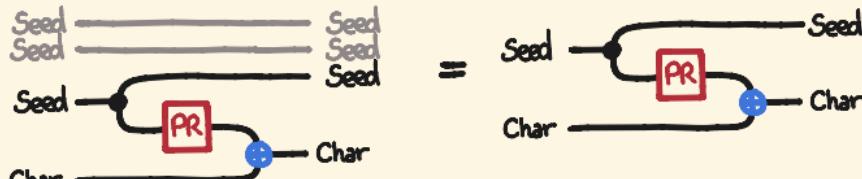
alice°



bob°



$\text{alice}^{+0} = \text{bob}^{+0} :=$



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

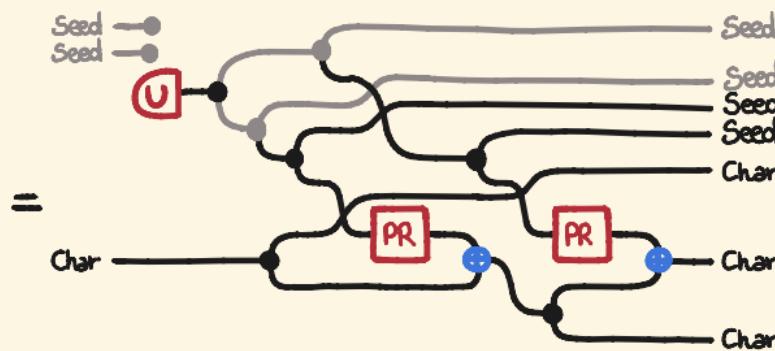
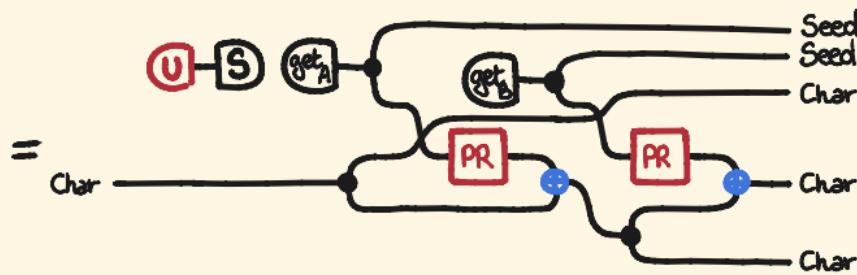
STREAM CIPHER IS SECURE

Proceed by coinduction.

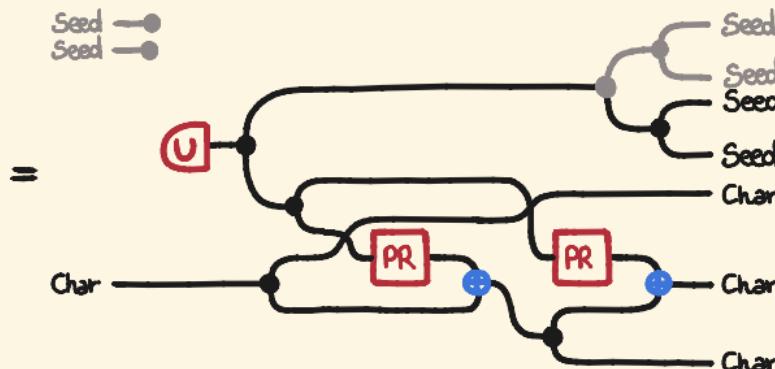
cipher⁰

= $\begin{cases} \text{cipher}(m)^0 = \text{do} \\ \quad \text{rand}() \rightarrow s \\ \quad \text{setSeed}(s) \rightsquigarrow () \\ \quad \text{getSeedA}() \rightsquigarrow s_A' \\ \quad \text{prng}(s_A') \rightarrow (s_A', k_A) \\ \quad \text{getSeedB}() \rightsquigarrow s_B' \\ \quad \text{prng}(s_B') \rightarrow (s_B', k_B) \\ \quad \text{return } (m, s_A', m \oplus k_A \oplus k_B, s_B', m \oplus k_A) \end{cases}$

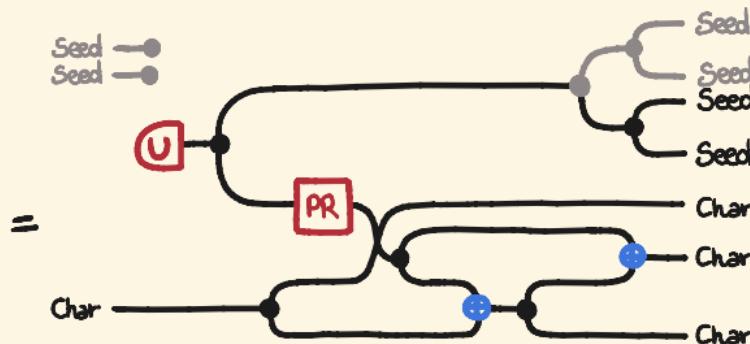
STREAM CIPHER IS SECURE



STREAM CIPHER IS SECURE

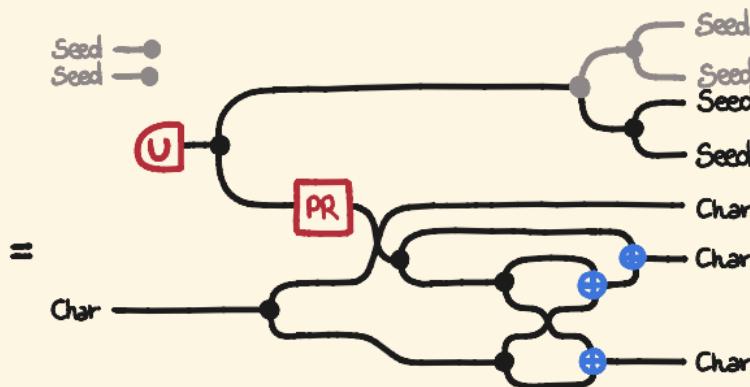


by associativity of copy

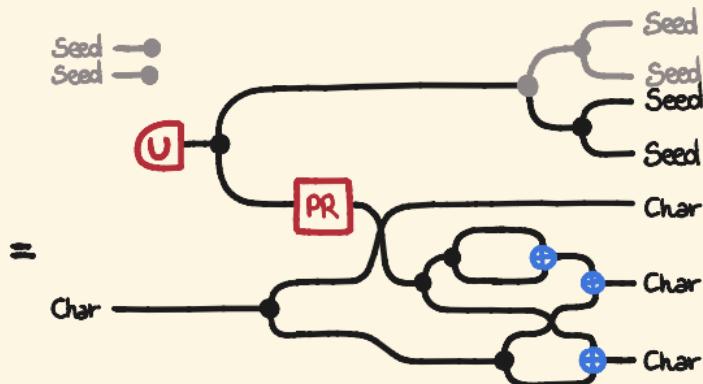


pseudorandom is deterministic

STREAM CIPHER IS SECURE

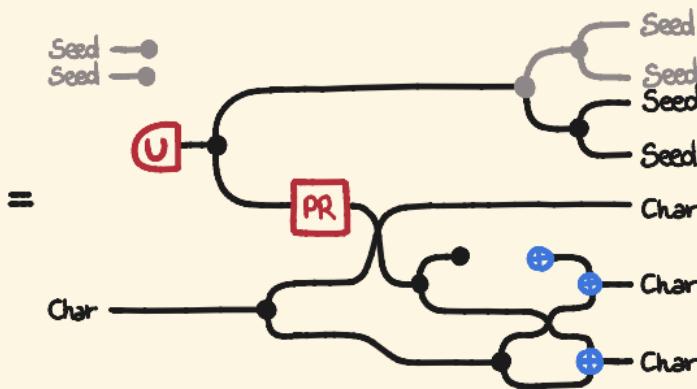


xor is deterministic

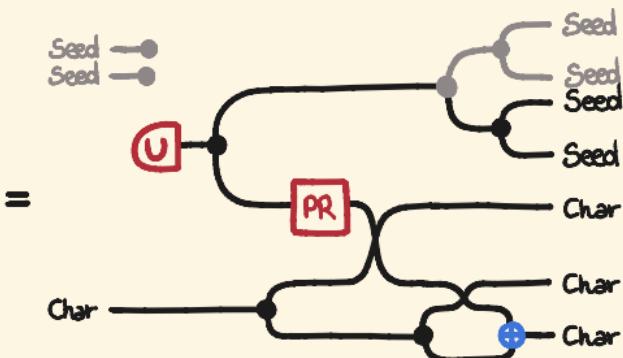


by associativity of copy
and xor

STREAM CIPHER IS SECURE

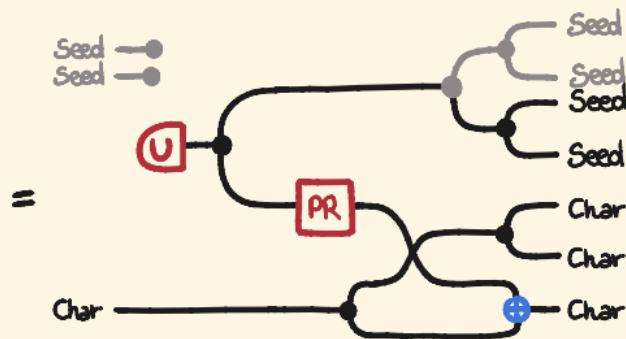


xor is nihilpotent

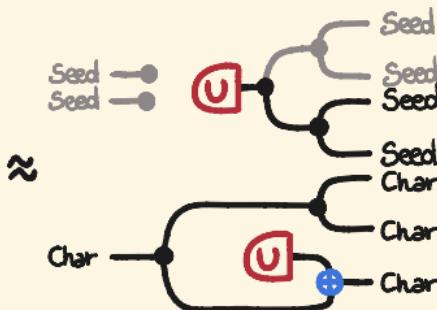


by unitality of copy
and xor

STREAM CIPHER IS SECURE

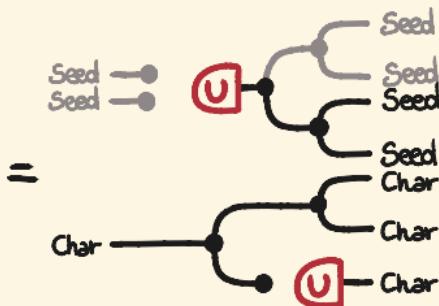


by associativity of copy

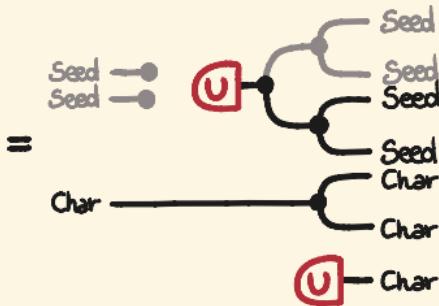


by assumption

STREAM CIPHER IS SECURE



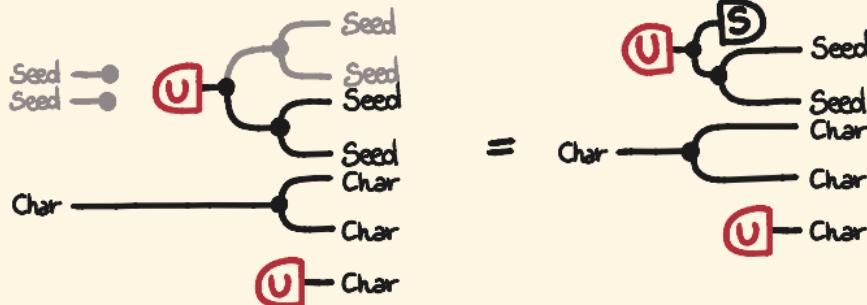
the uniform distribution is a
Sweedler integral for xor



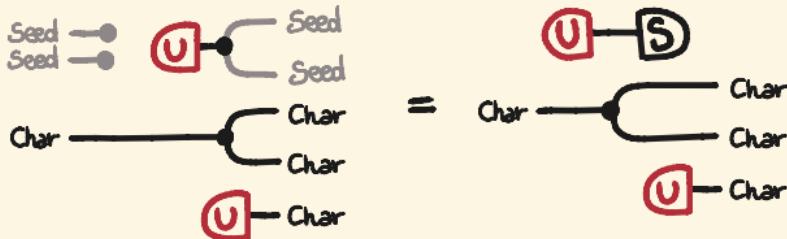
by unitality of copy

STREAM CIPHER IS SECURE

$\text{cipher}^\circ \approx$



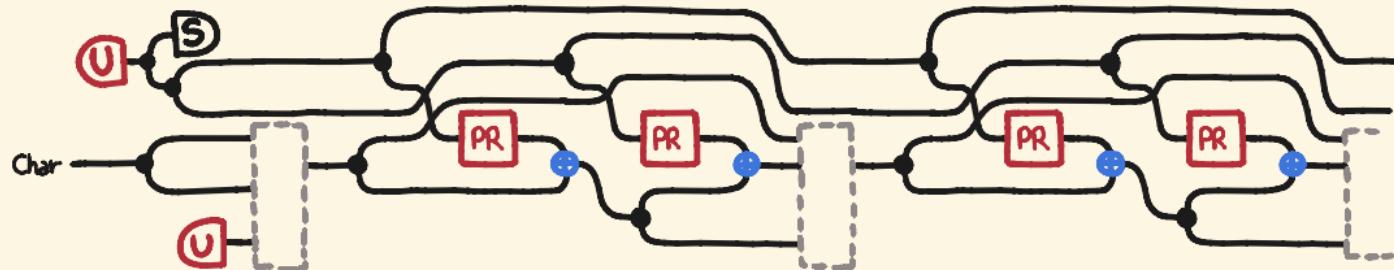
$\text{secure}^\circ :=$



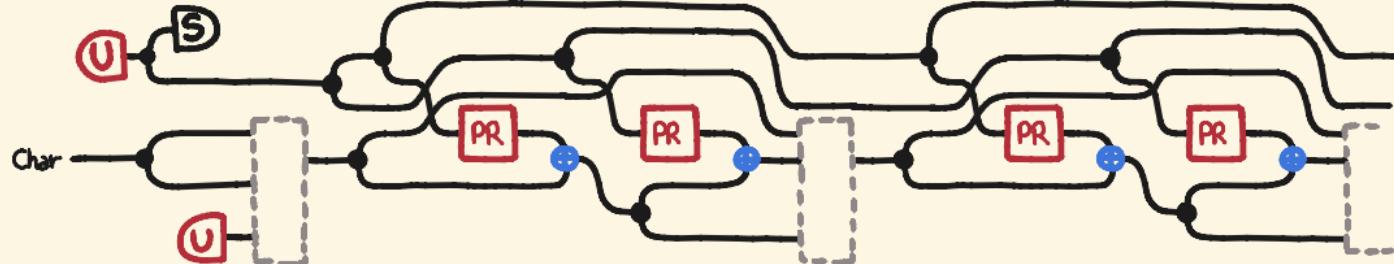
STREAM CIPHER IS SECURE

cipher

=

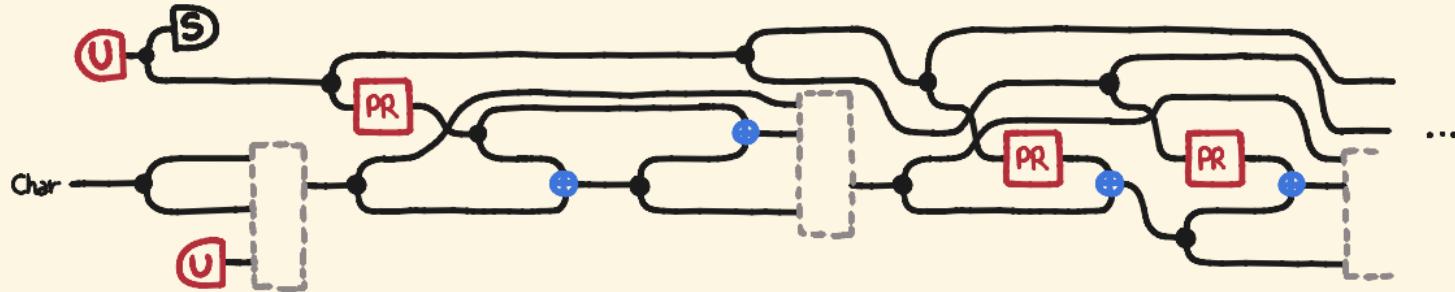


= (by sliding)

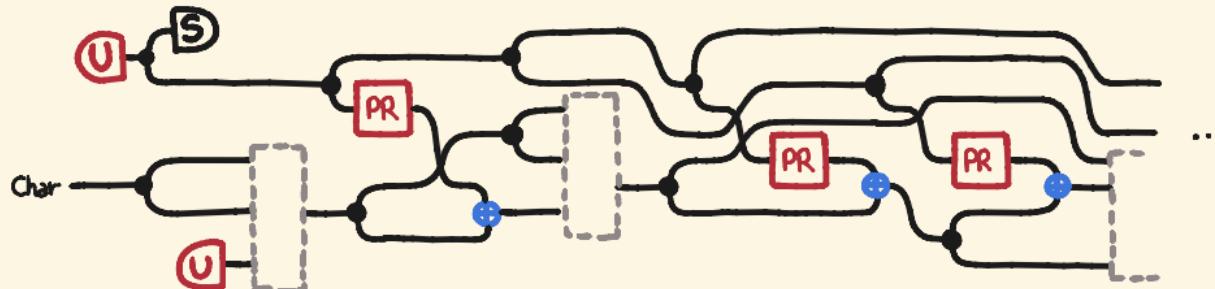


STREAM CIPHER IS SECURE

= (pseudorandom is deterministic)

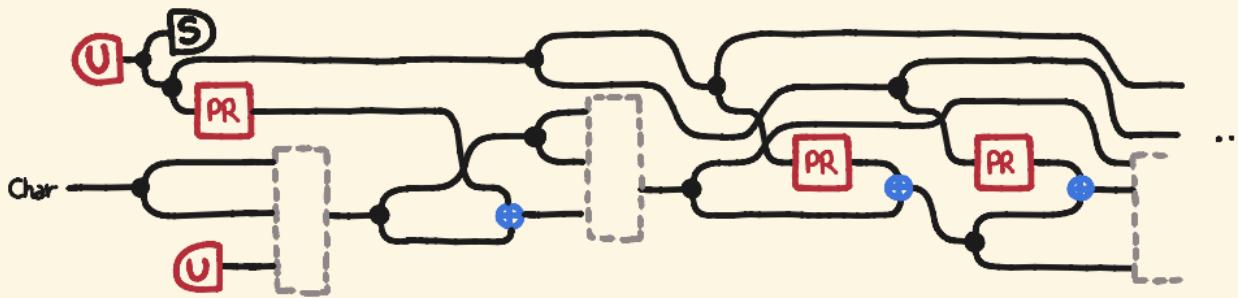


= (xor is deterministic and nihilpotent)

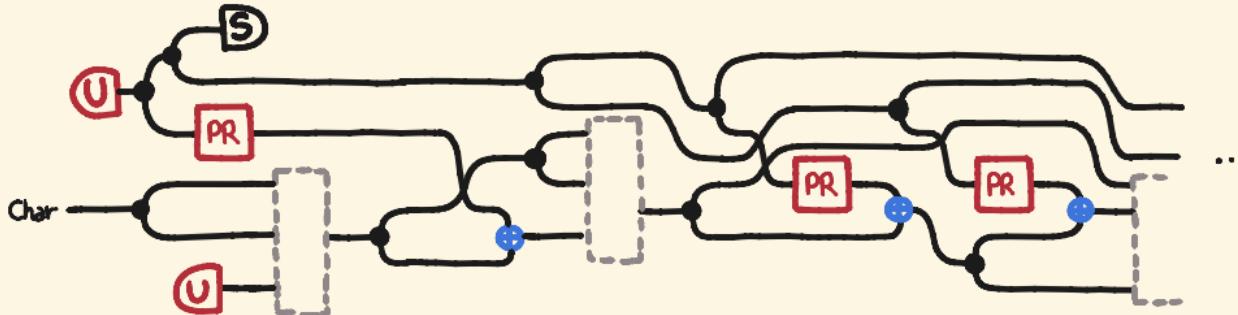


STREAM CIPHER IS SECURE

= (by sliding)

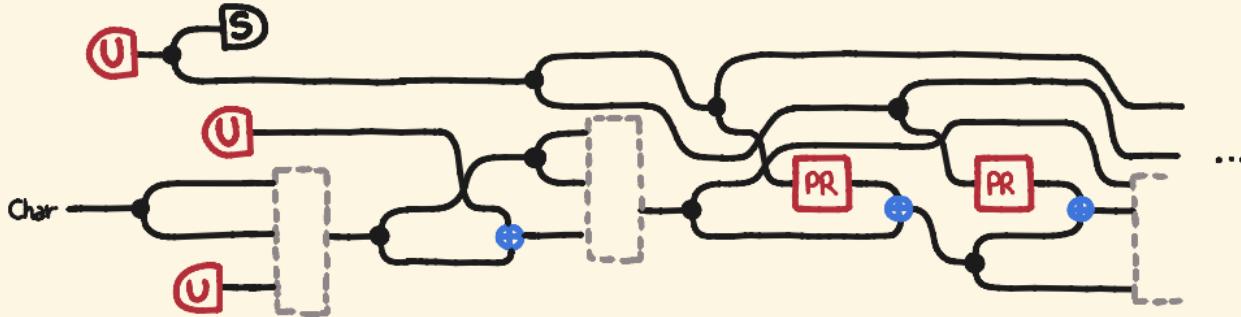


= (by associativity)

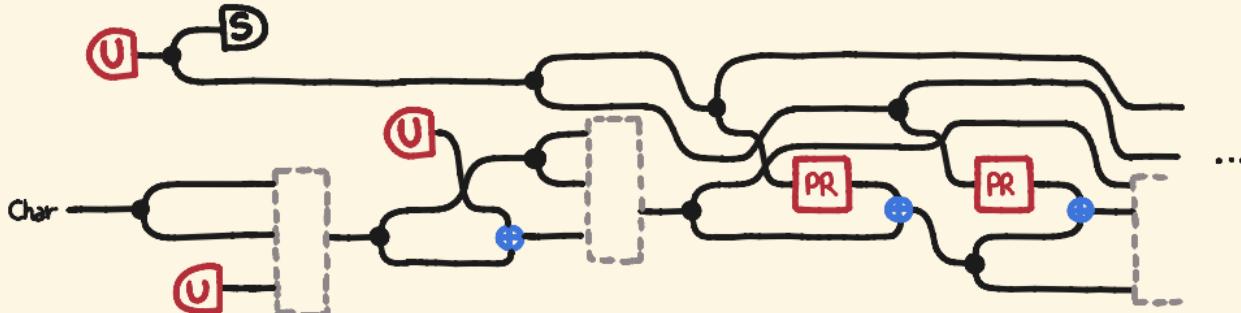


STREAM CIPHER IS SECURE

≈ (by assumption on pseudorandom)

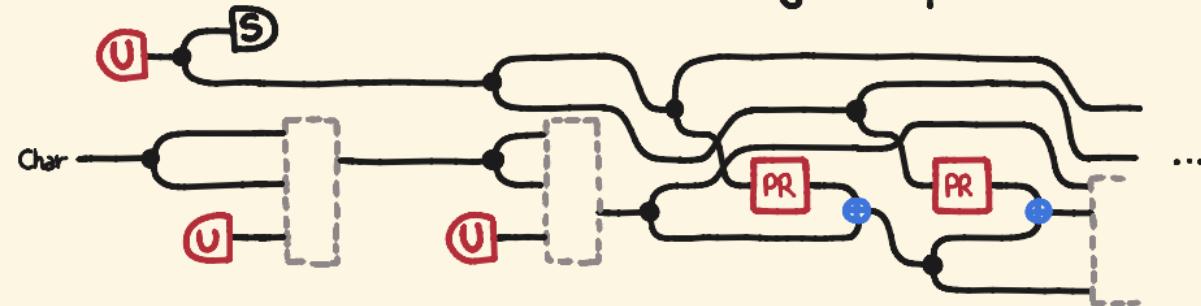


= (by sliding)

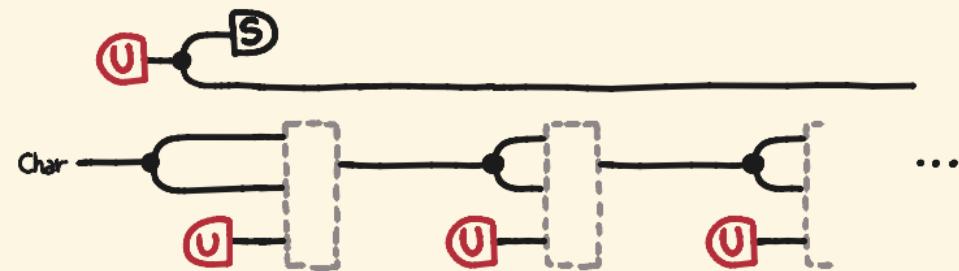


STREAM CIPHER IS SECURE

= (unif is a Sweedler integral for xor)

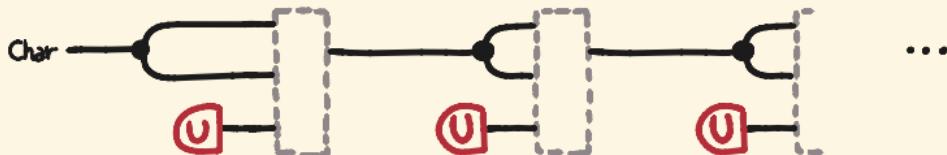


≈ (by coinduction)

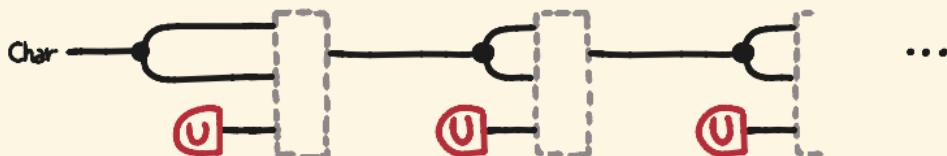


STREAM CIPHER IS SECURE

= (by coinduction)



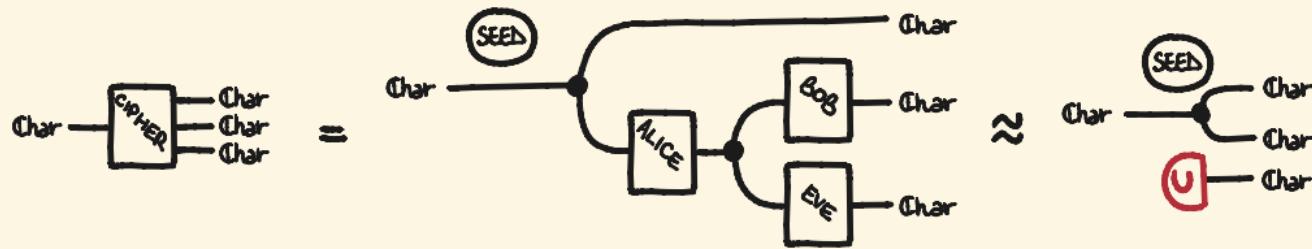
= (by unitality)



= secure

STREAM CIPHER IS SECURE

We have shown



using sliding and coinduction.