Memoryful Geometry of Interaction II

Recursion and Adequacy

Koko Muroya
(Univ. Tokyo)

Naohiko Hoshino
(Kyoto Univ.)

Ichiro Hasuo
(Univ. Tokyo)

POPL (St. Petersburg), January 22, 2016

Geometry of Interaction (GoI)

semantics of { linear logic proofs [Girard '89]
functional programs

token machine semantics

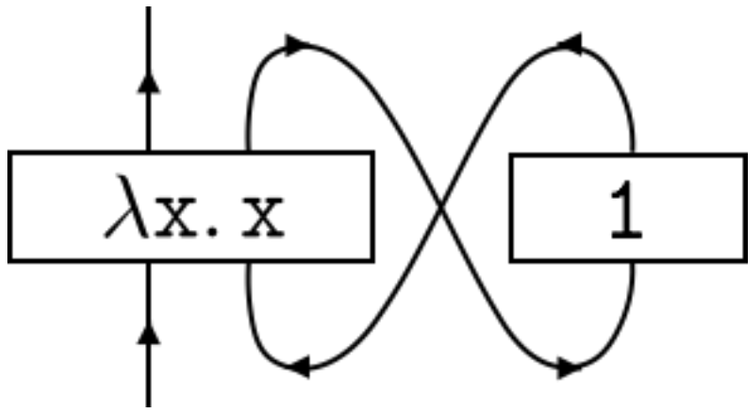
[Mackie '95] [Danos & Renier '99]

Token Machine Semantics

PCF terms

$(\lambda \mathrm{x}.\, \mathrm{x})\ 1$

token machines

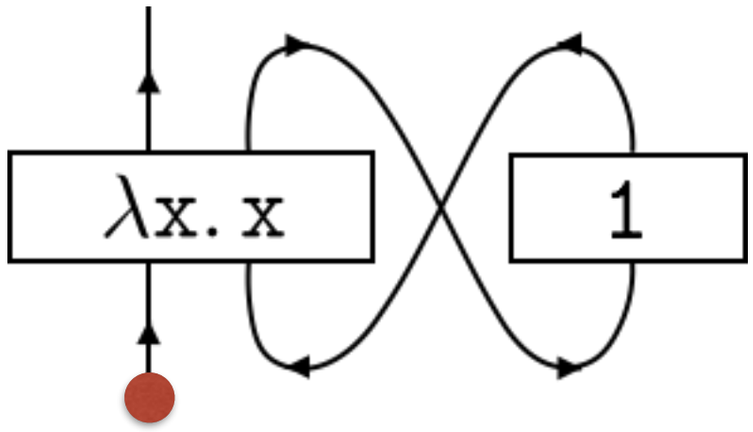


Token Machine Semantics

PCF terms

$(\lambda \mathrm{x}.\, \mathrm{x})\ 1$

token machines

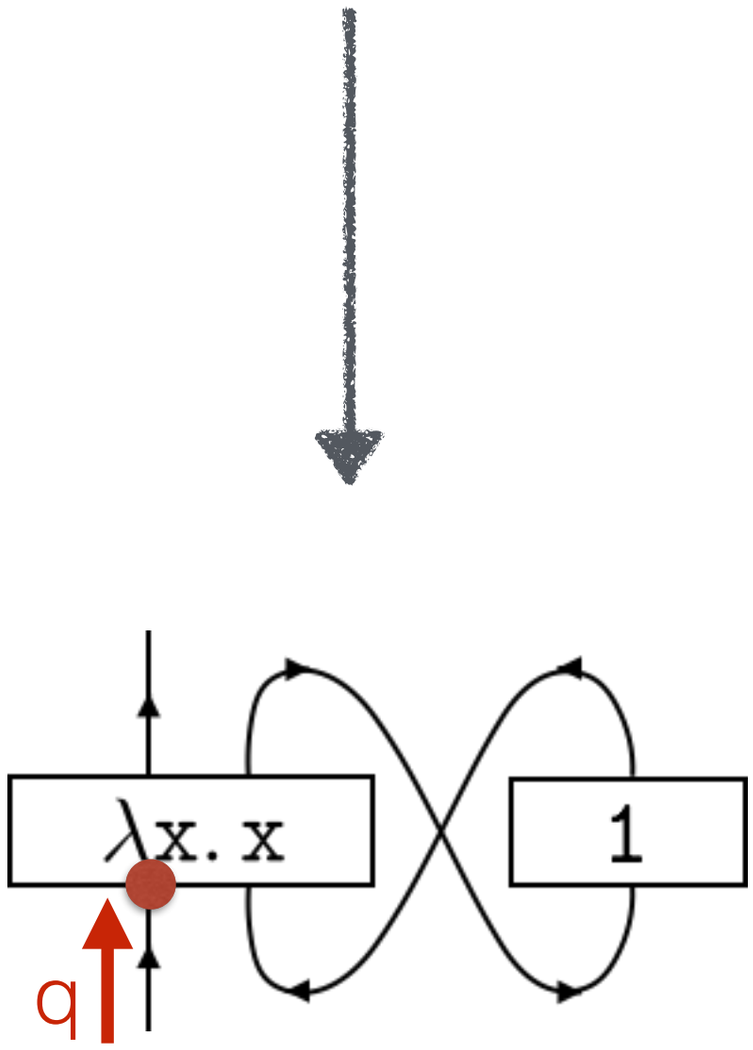


Token Machine Semantics

PCF terms

$(\lambda \mathrm{x}.\, \mathrm{x})\ 1$

token machines

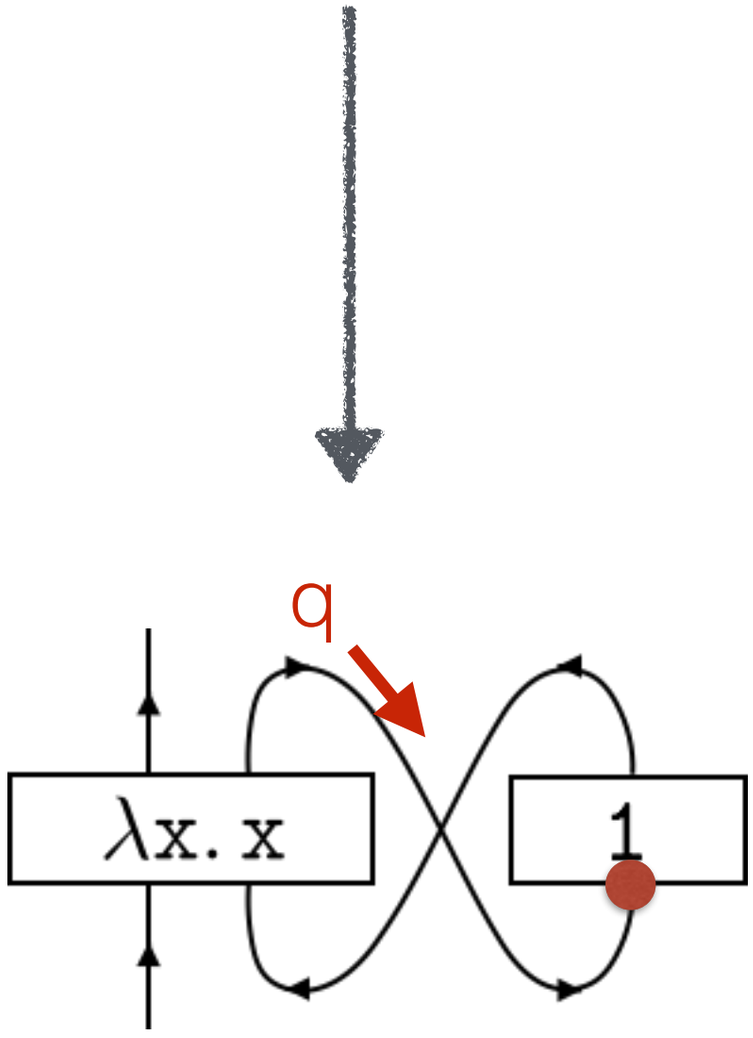


Token Machine Semantics

PCF terms

$(\lambda \mathrm{x}.\, \mathrm{x})\ 1$

token machines

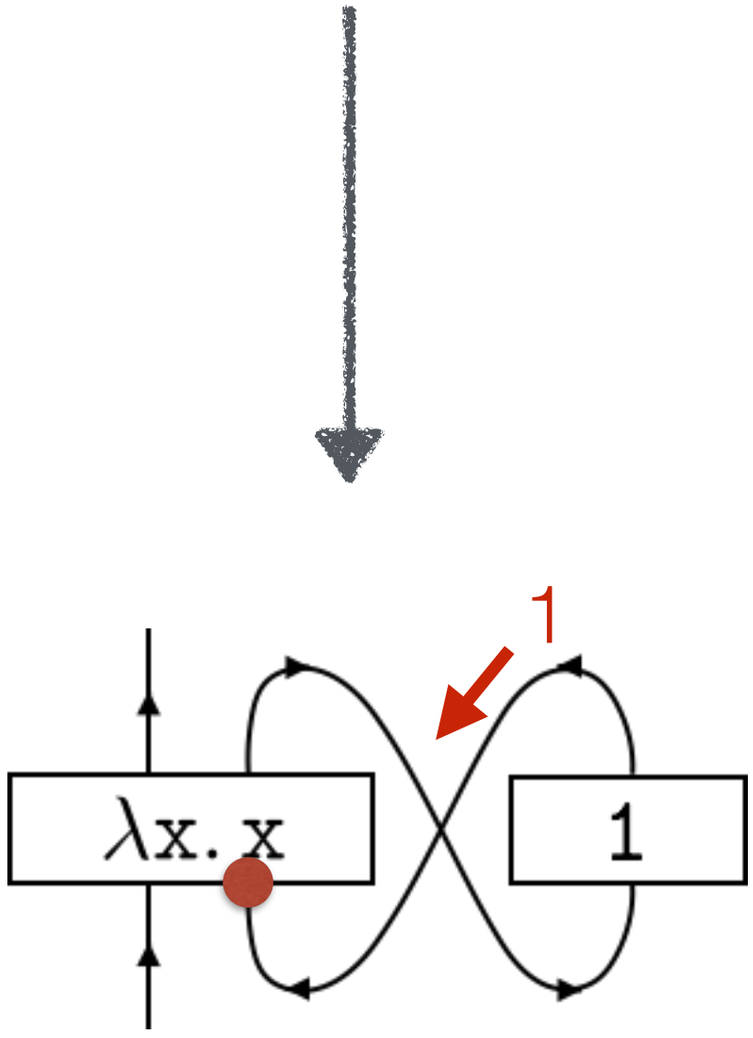


Token Machine Semantics

PCF terms

token machines

$(\lambda \mathrm{x}.\, \mathrm{x})\ 1$

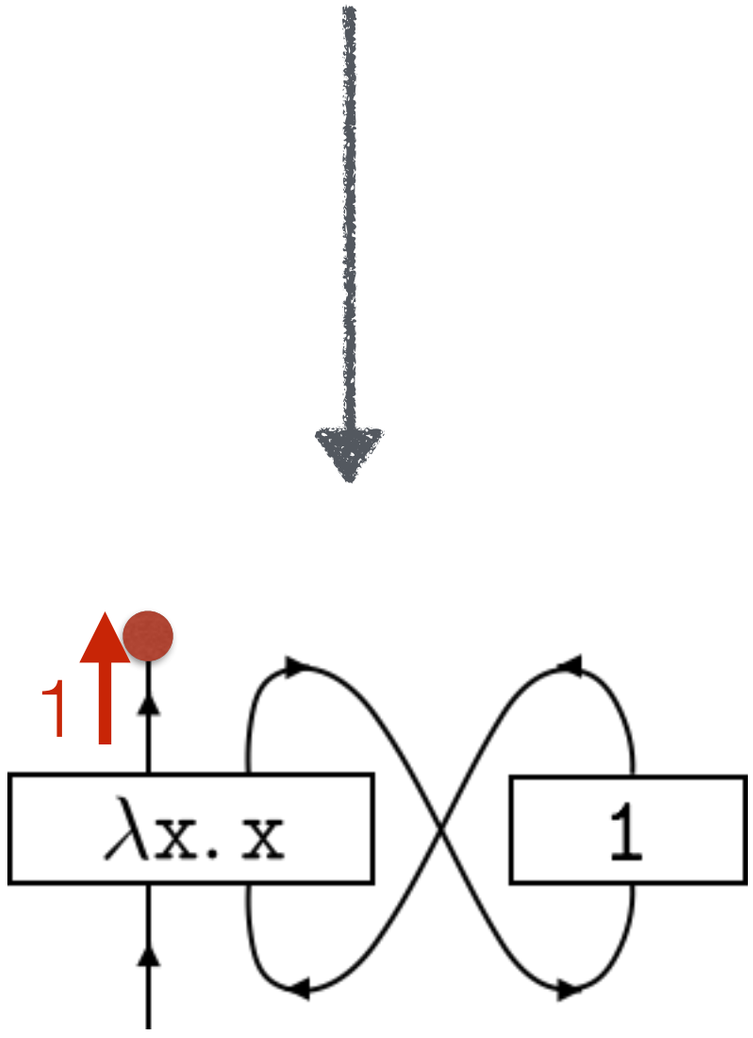


Token Machine Semantics

PCF terms

$(\lambda \mathrm{x}.\, \mathrm{x})\ 1$

token machines

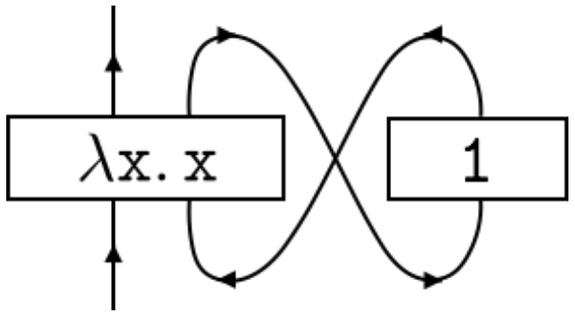


Token Machine Semantics

PCF terms

$(\lambda x.\, x)\, 1$

token machines



- applications to compilation

  [Mackie ’95] [Pinto ’01] [Ghica ’07]

- categorical formalization

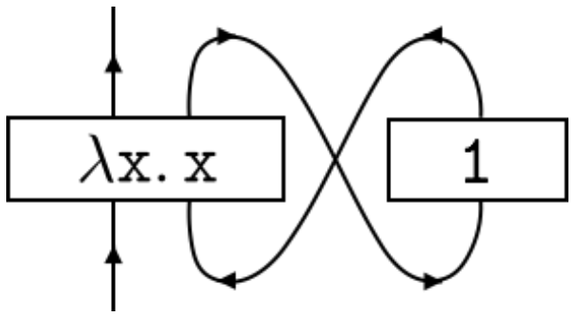
  [Abramsky et al. ’02]

Token Machine Semantics

PCF terms

$(\lambda \mathrm{x}.\, \mathrm{x})\ 1$

token machines



+ computational effects

Token Machine Semantics

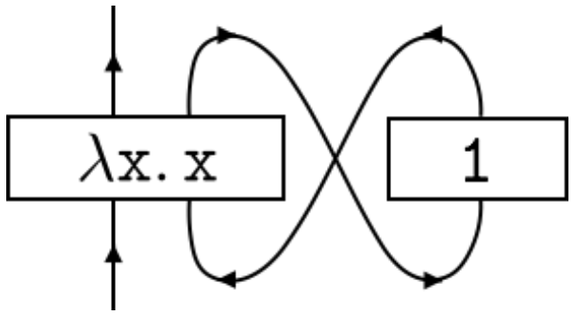
PCF terms

recursion

+ computational effects

$(\lambda \mathrm{x}.\, \mathrm{x})\ 1$

token machines



Token Machine Semantics

PCF terms

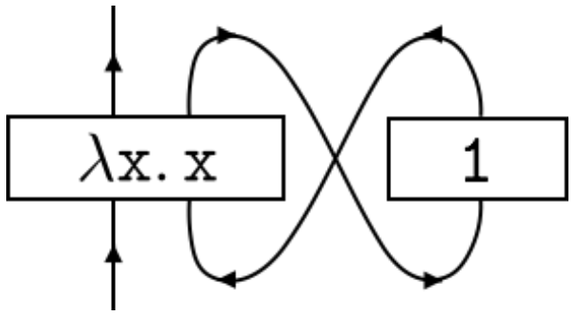
recursion

$(\lambda x.\, x)\ 1$

+ computational effects

- memoryful GoI [Hoshino, — & Hasuo ’14]

token machines



Token Machine Semantics

PCF terms

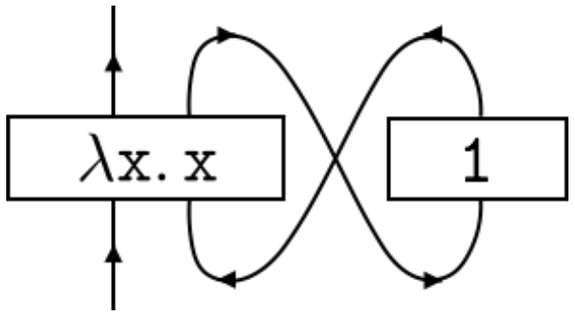
recursion

$(\lambda x.\, x)\ 1$

+ computational effects

- memoryful GoI [Hoshino, — & Hasuo ’14]
- memoryful GoI II

token machines



Memoryful GoI

effectful PCF terms

call-by-value PCF terms

with algebraic operations [Plotkin & Power '01]

- nondeterministic choice $\texttt{choose}(\texttt{M}, \texttt{N})$
- probabilistic choice $\texttt{choose}_p(\texttt{M}, \texttt{N})$
- actions on global states $\texttt{lookup}_\ell(\texttt{M}_{v_0}, \ldots, \texttt{M}_{v_n})$

$\texttt{update}_{\ell,v}(\texttt{M})$

transducers

Memoryful GoI

effectful PCF terms

call-by-value PCF terms

with algebraic operations [Plotkin & Power '01]

$$(\lambda \mathtt{x}.\, \mathtt{M})\, \mathtt{V} \to \mathtt{M}[\mathtt{V}/\mathtt{x}]$$

$$\mathtt{rec}(\mathtt{f},\mathtt{x}).\,\mathtt{M} \to (\lambda \mathtt{x}.\,\mathtt{M})\,[\mathtt{rec}(\mathtt{f},\mathtt{x}).\,\mathtt{M}/\mathtt{f}]$$

$$\mathtt{op}(\mathtt{M}_1,\ldots,\mathtt{M}_n) \xrightarrow{\mathtt{op}_i} \mathtt{M}_i$$

$$\mathtt{op}() \downarrow_{\mathtt{op}}$$

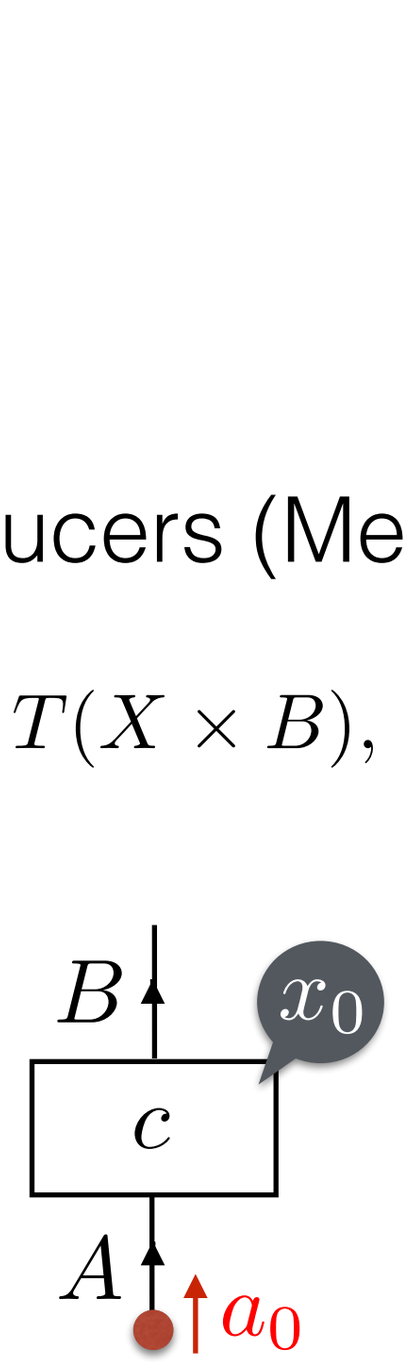
transducers

Memoryful GoI

effectful PCF terms

stream transducers (Mealy machines)

$$(X,\ c\colon X \times A \to T(X \times B),\ x_0 \in X)\colon A \nrightarrow B$$



transducers

Memoryful GoI

effectful PCF terms

transducers

stream transducers (Mealy machines)

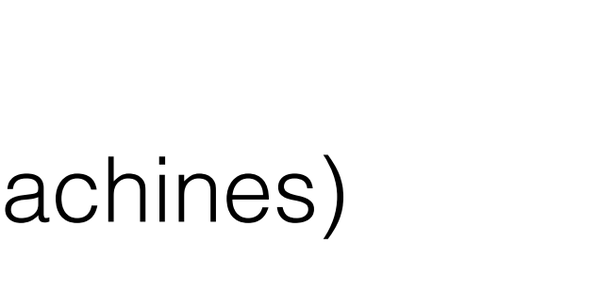
$$(X,\ c\colon X \times A \to T(X \times B),\ x_0 \in X)\colon A \nrightarrow B$$

- nondeterministic transitions

$$TX = \mathcal{P}(X)$$

- probabilistic transitions

$$TX = \mathcal{D}_{\leq 1}(X)$$

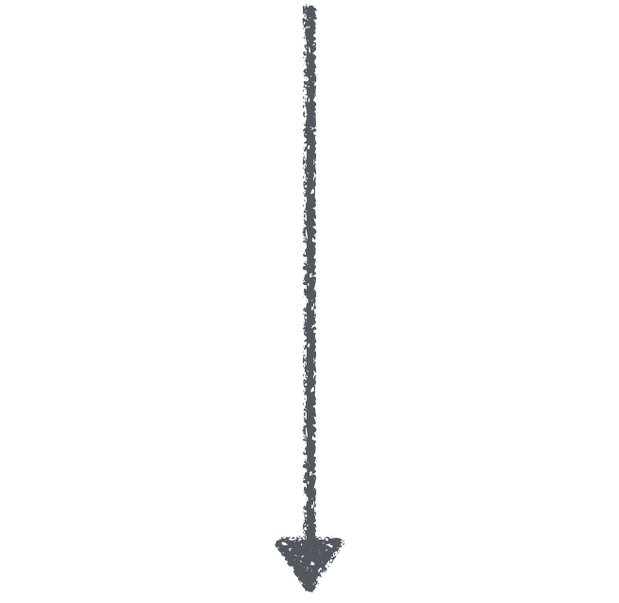


- transitions with global states

$$TX = (1 + X \times S)^S$$

Memoryful GoI

effectful PCF terms



transducers

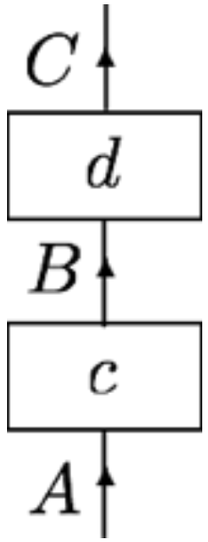
compositional translation

via coalgebraic component calculus

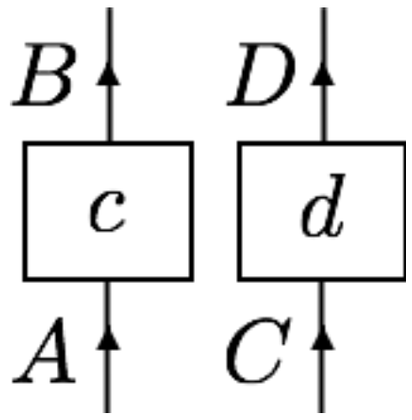
[Barbosa ’03] [Hasuo, Jacobs ’11]

Component Calculus

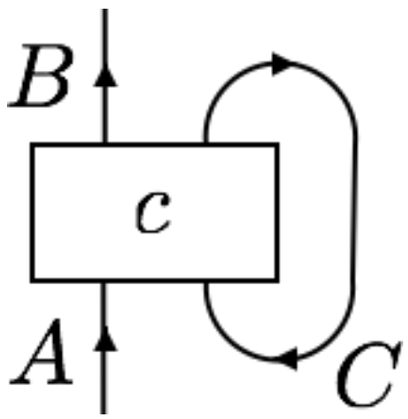
sequential composition



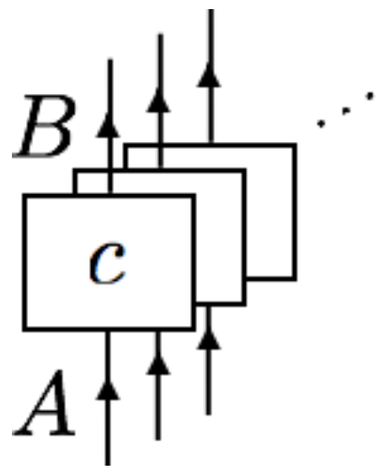
parallel composition



trace operator

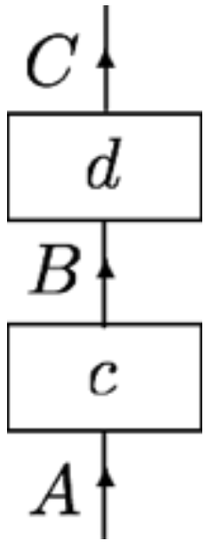


countable copy operator

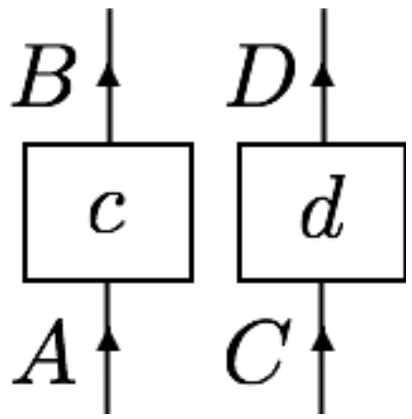


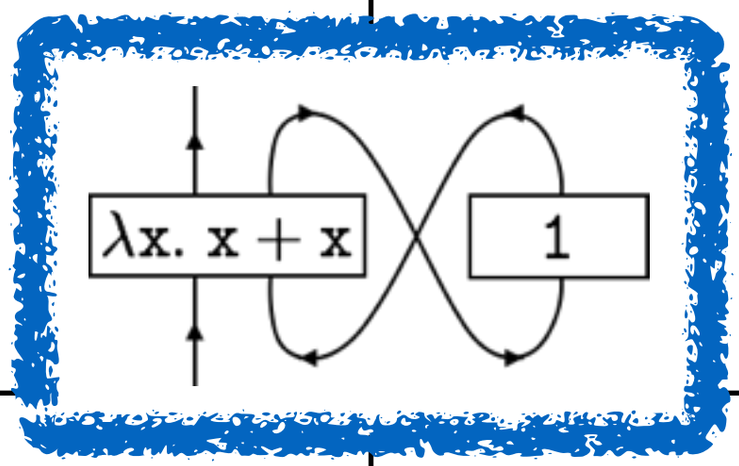
Component Calculus

sequential composition

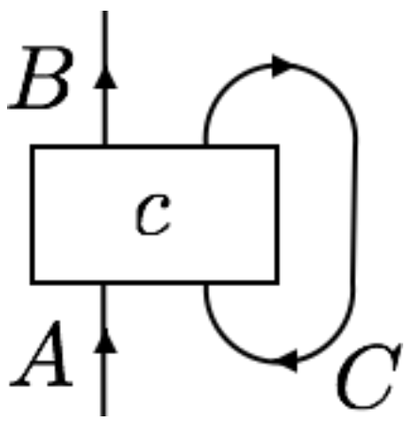


parallel composition

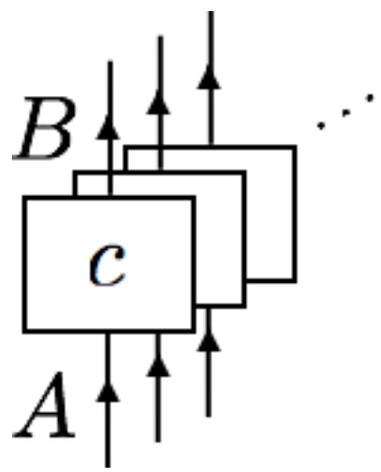




trace operator

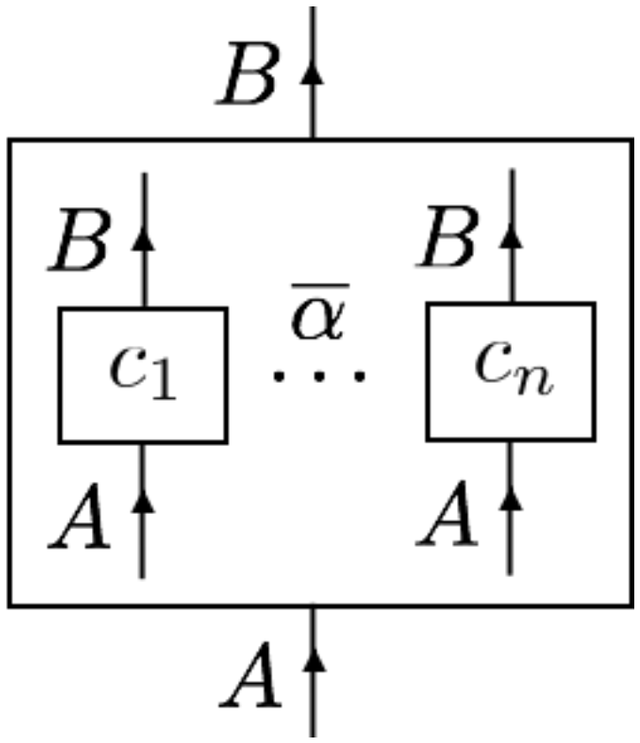


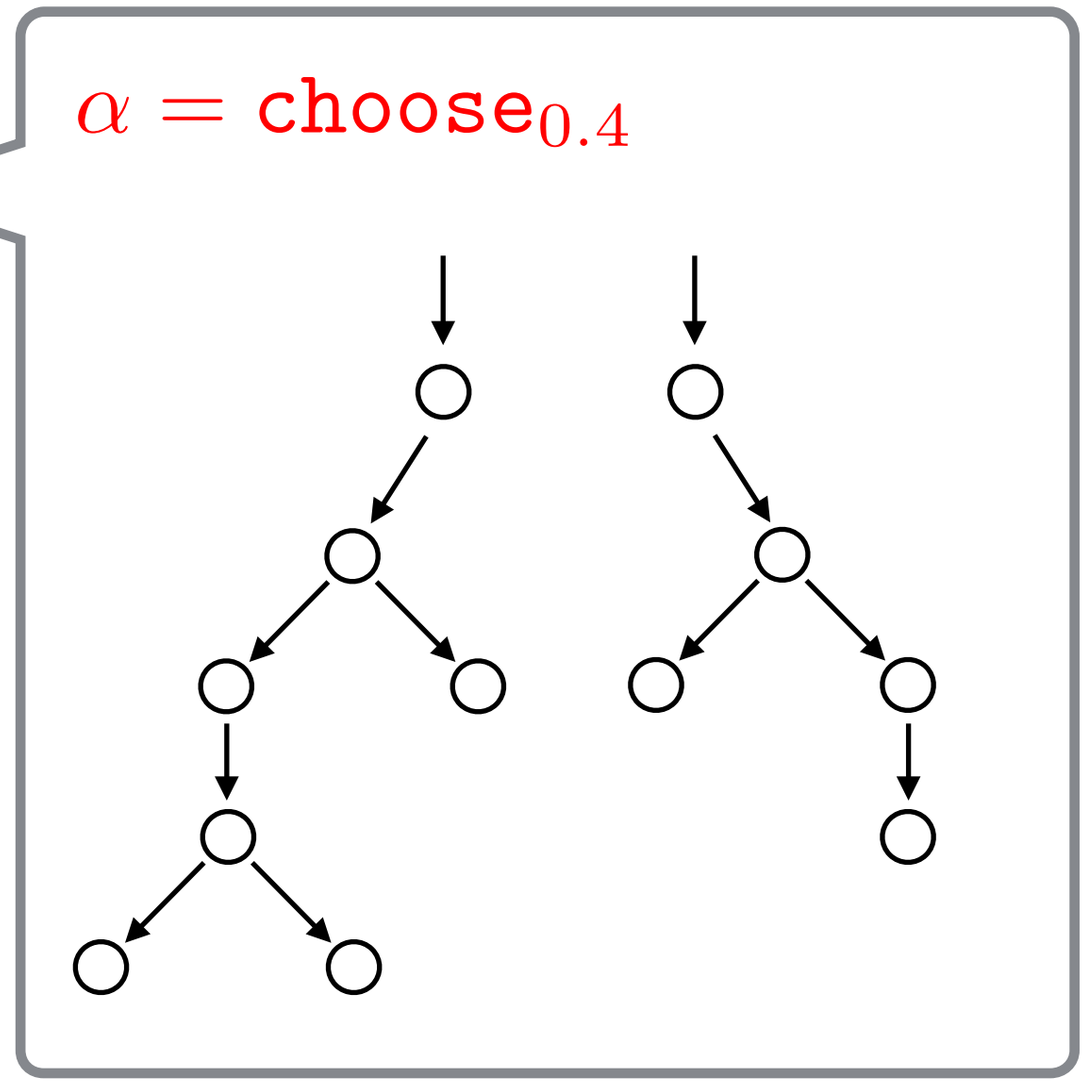
countable copy operator



Component Calculus for Effects

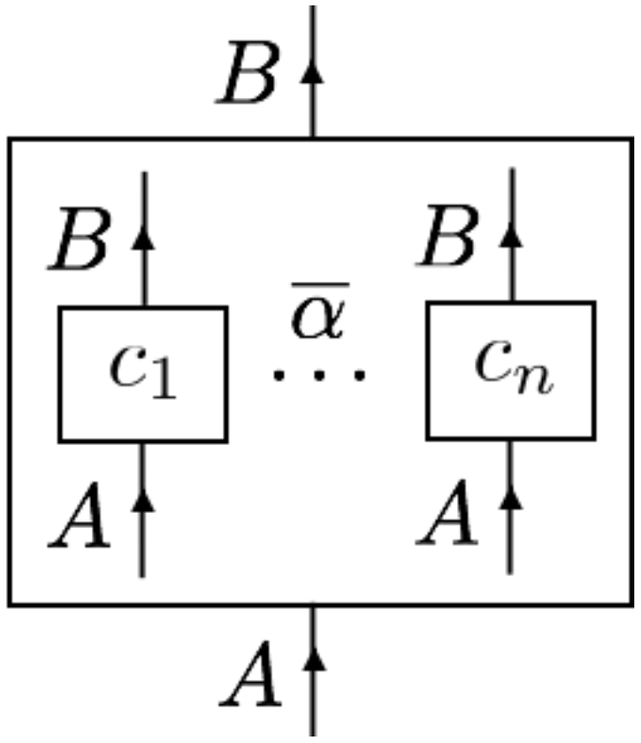
lifted algebraic operations

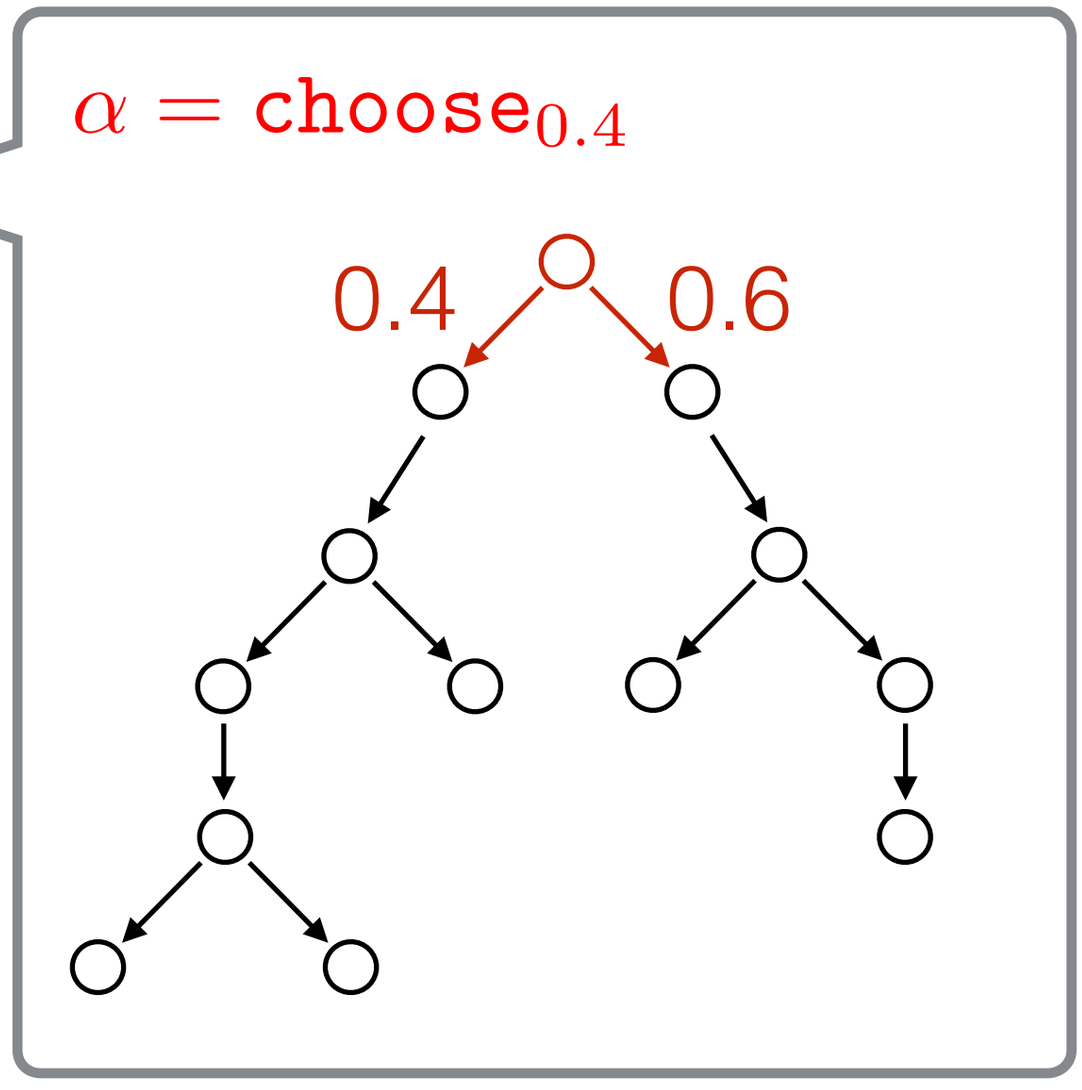




Component Calculus for Effects

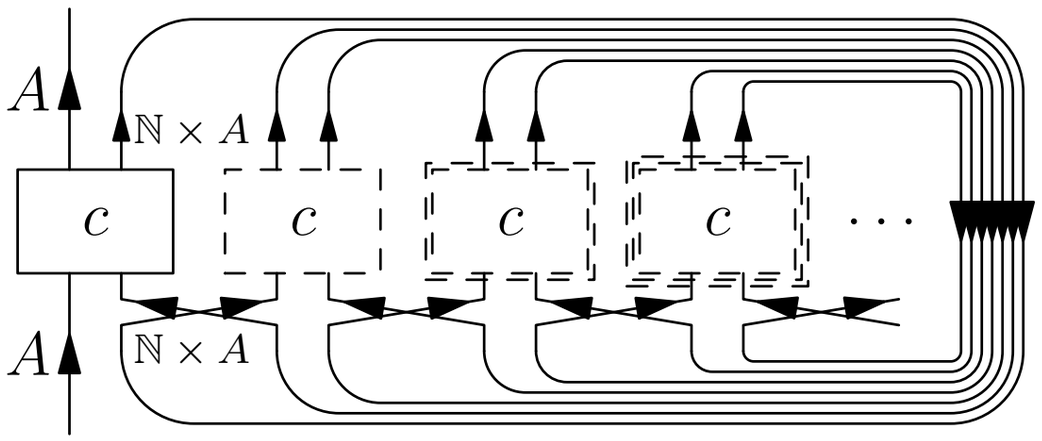
lifted algebraic operations



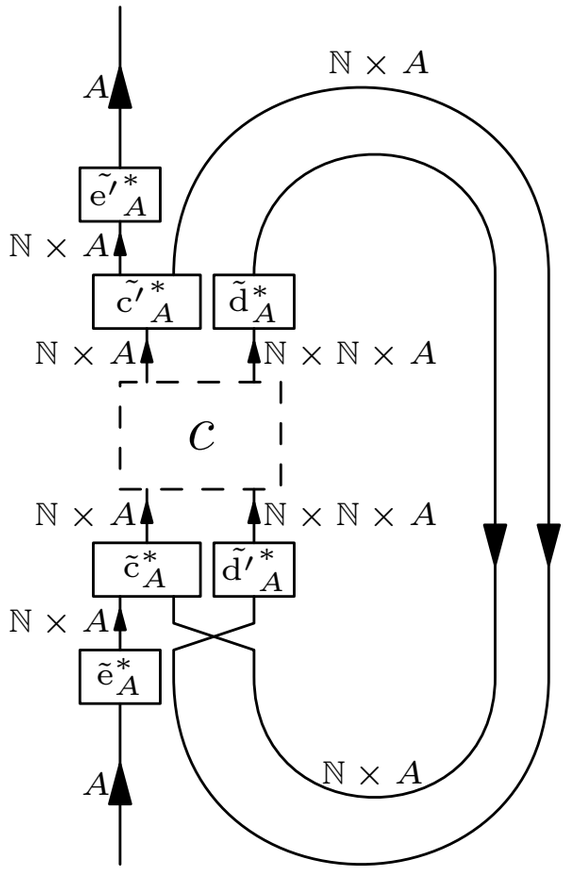


Component Calculus for Recursion

Girard style fixed point operator

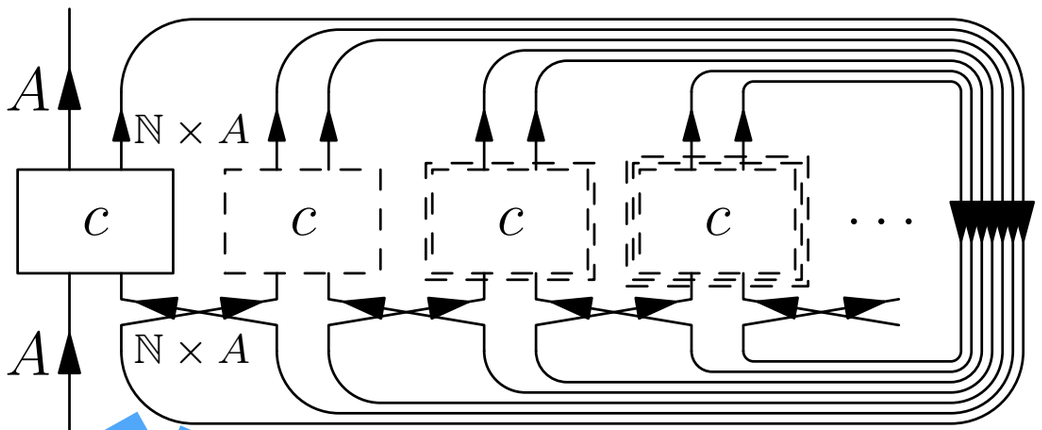


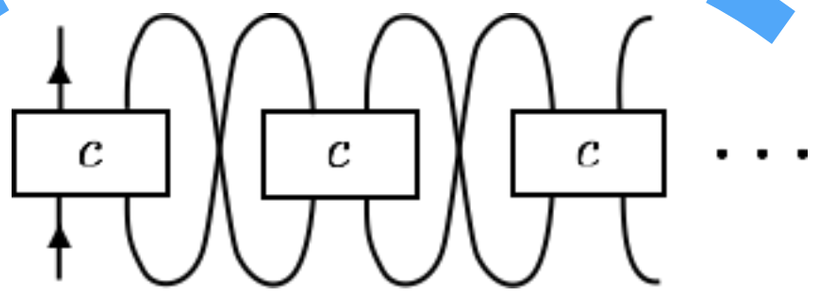
Mackie style fixed point operator



Component Calculus for Recursion

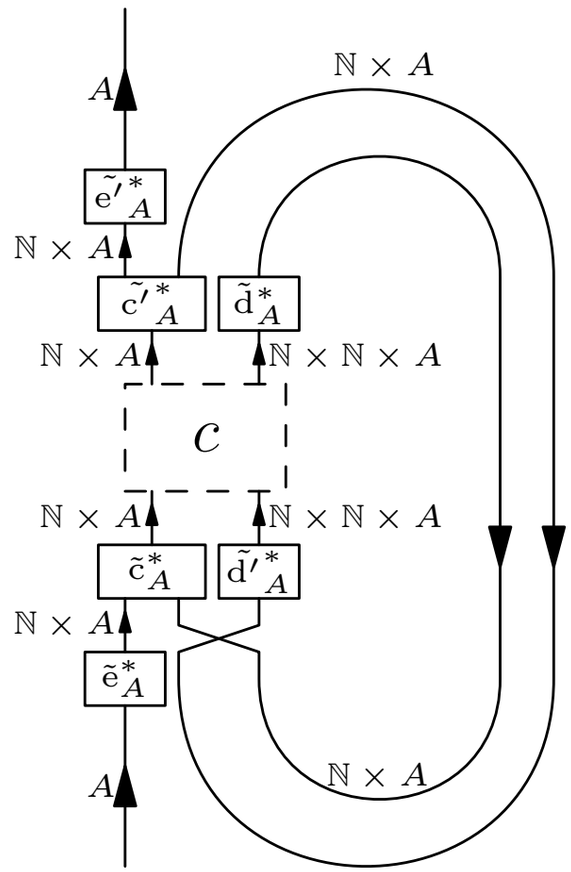
Girard style fixed point operator





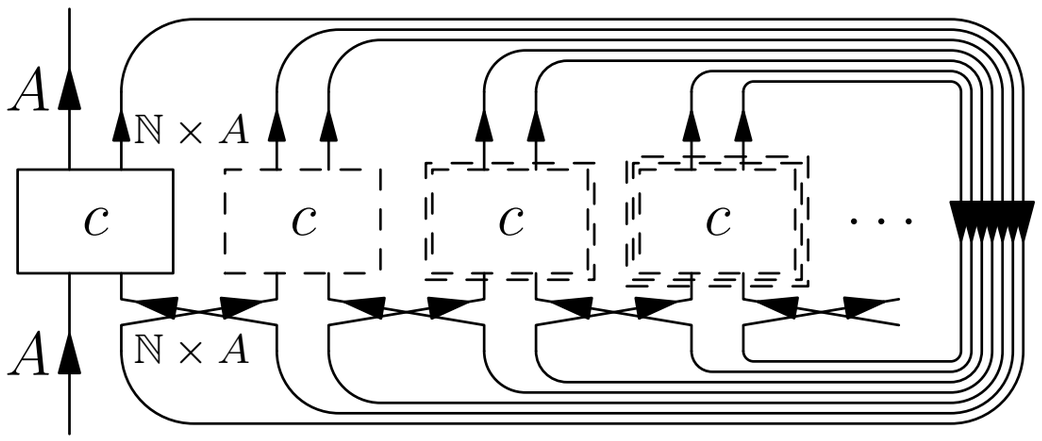
$$\mathrm{fix}(F) = F(F(F(\cdots)))$$

Mackie style fixed point operator

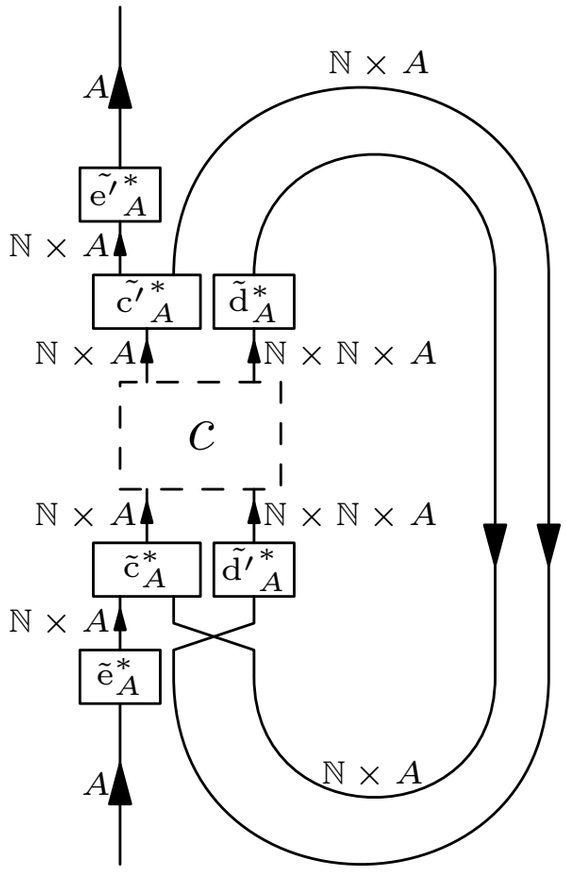


Component Calculus for Recursion

Girard style fixed point operator

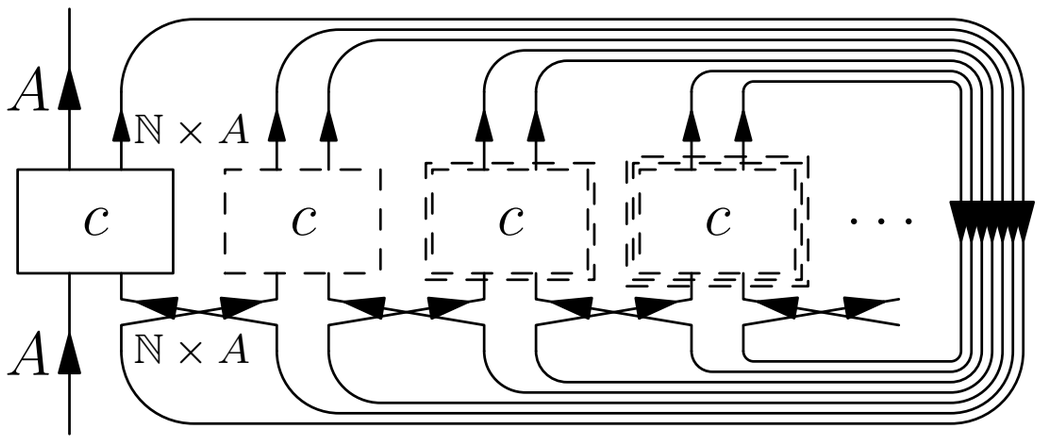


Mackie style fixed point operator



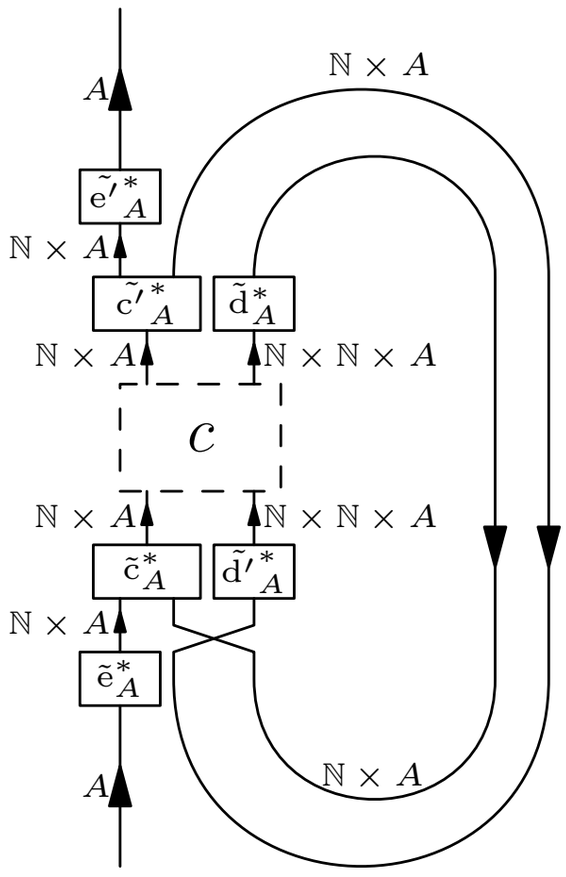
Component Calculus for Recursion

Girard style fixed point operator



- fixed point wrt. “cross connection”
- supremum of finite approximations

Mackie style fixed point operator

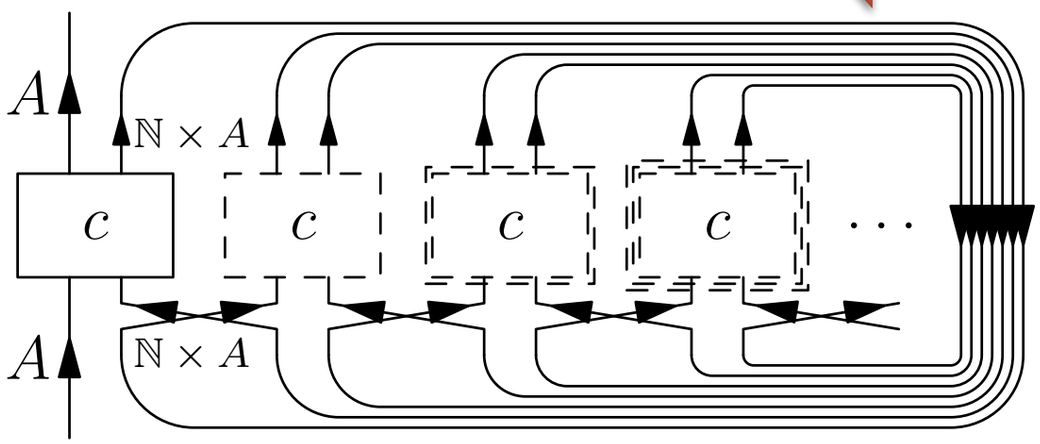


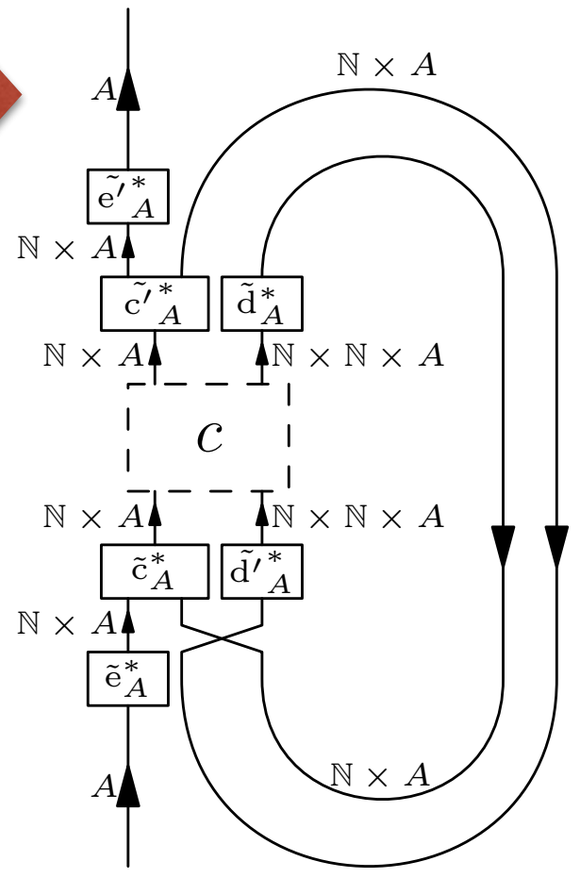
Component Calculus for Recursion

Girard style
fixed point operator

coincidence

Mackie style
fixed point operator

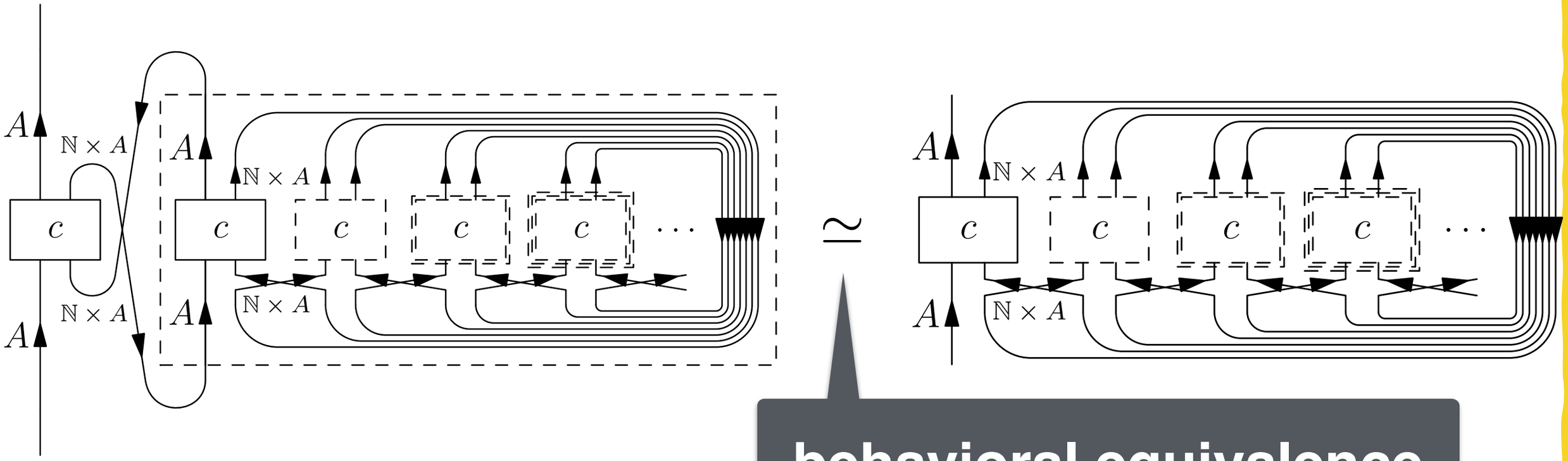




- fixed point wrt. “cross connection”
- supremum of finite approximations

Component Calculus for Recursion

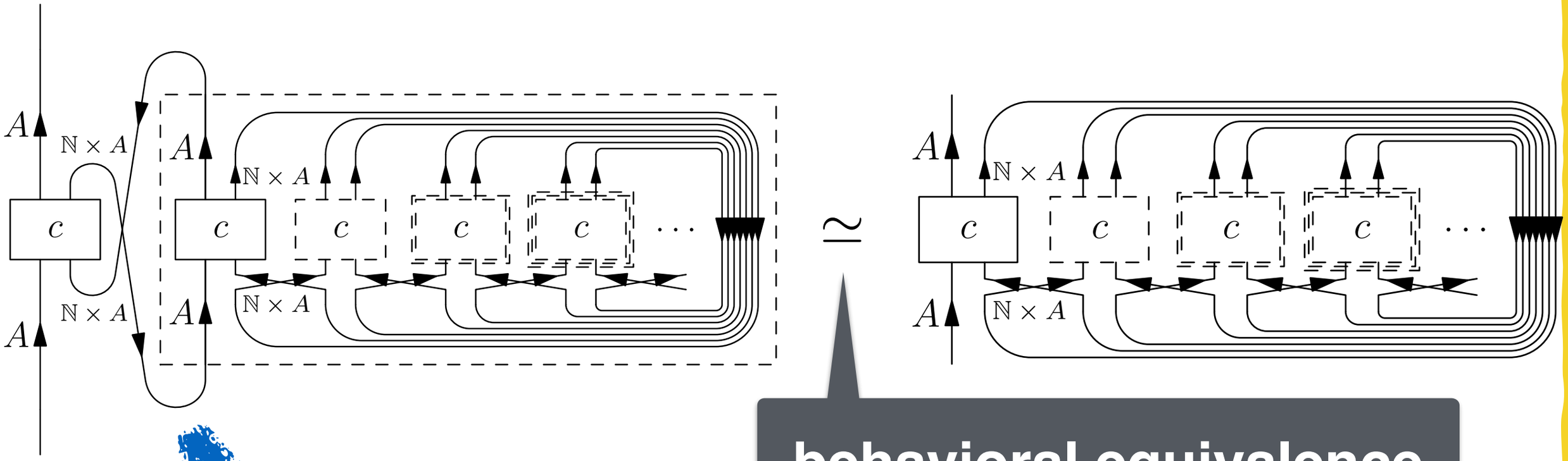
Prop. (Girard style as **fixed point** wrt. “cross connection”)



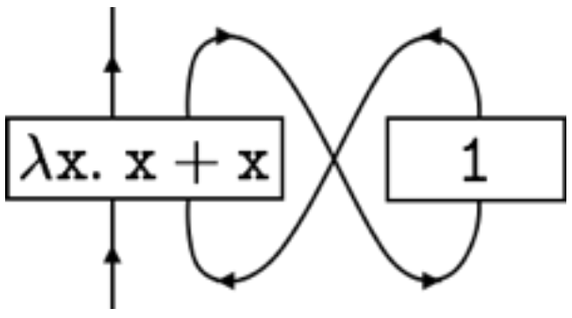
behavioral equivalence

Component Calculus for Recursion

Prop. (Girard style as **fixed point** wrt. “cross connection”)

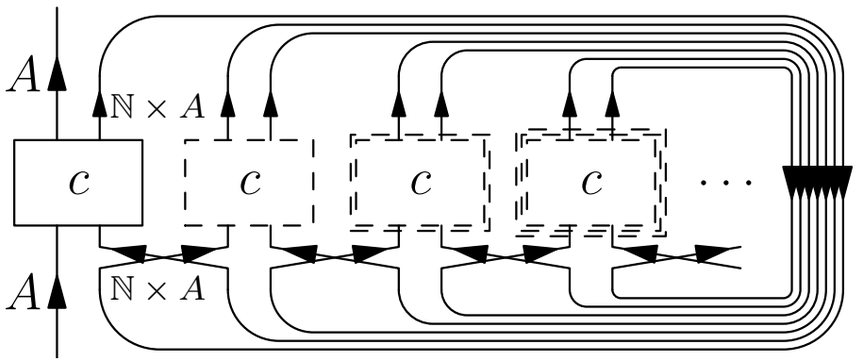


behavioral equivalence

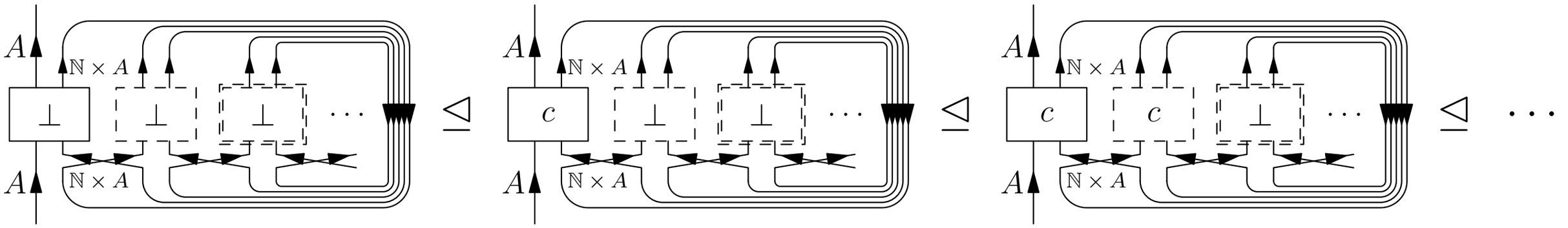


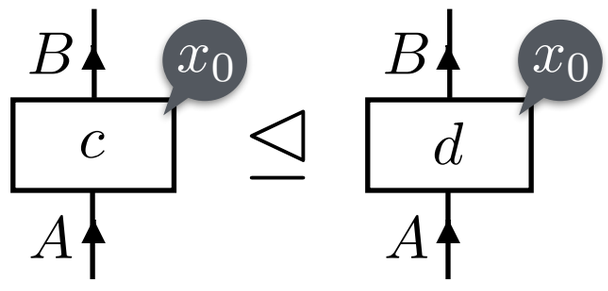
Component Calculus for Recursion

Prop. (Girard style as **supremum** of finite approximations)



is the supremum of the ω-chain

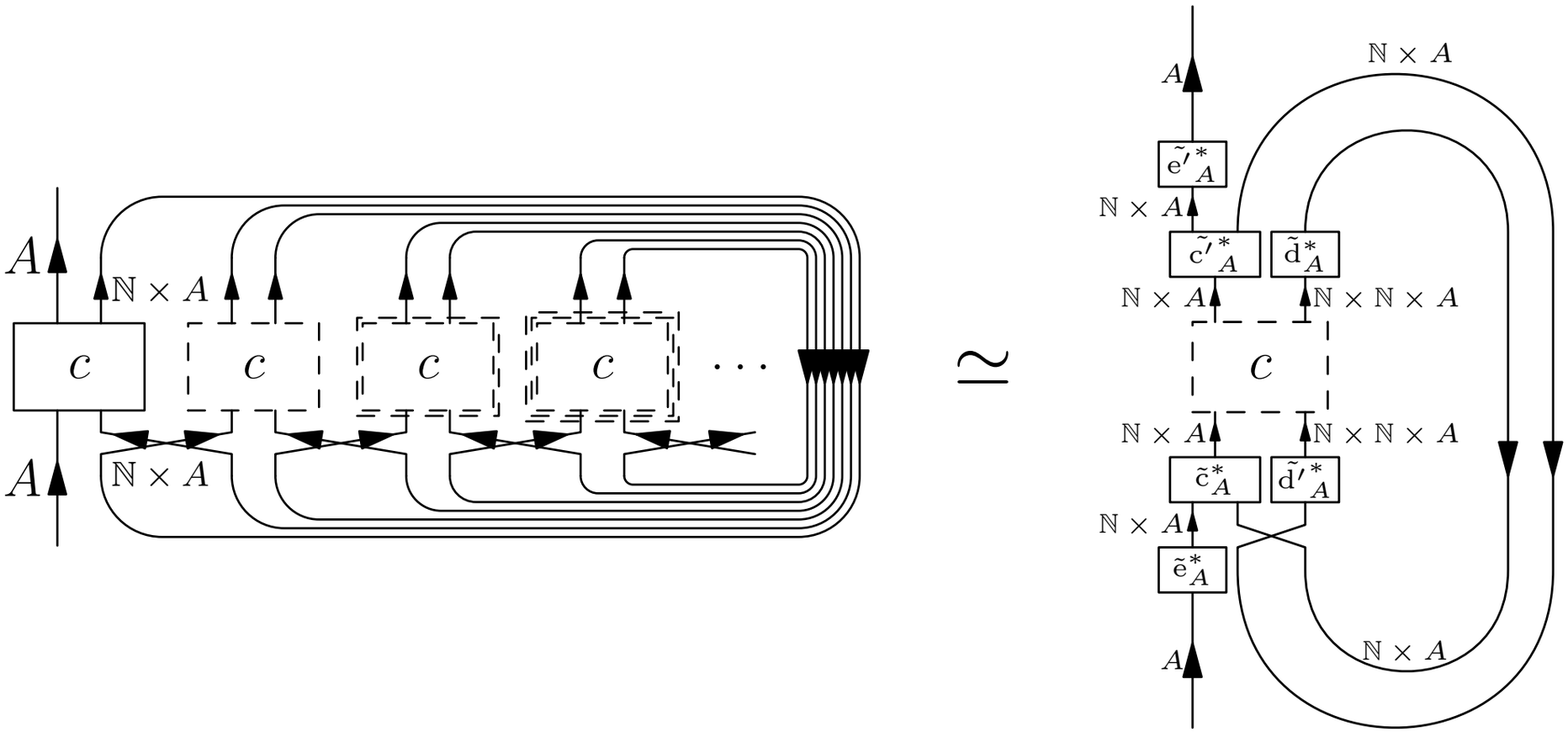


where 

$\overset{\text{def.}}{\Longleftrightarrow}$ $T(X \times B)$ with $c \sqsubseteq d$ over $X \times A$

Component Calculus for Recursion

Thm. (**coincidence** of Girard style & Mackie style)



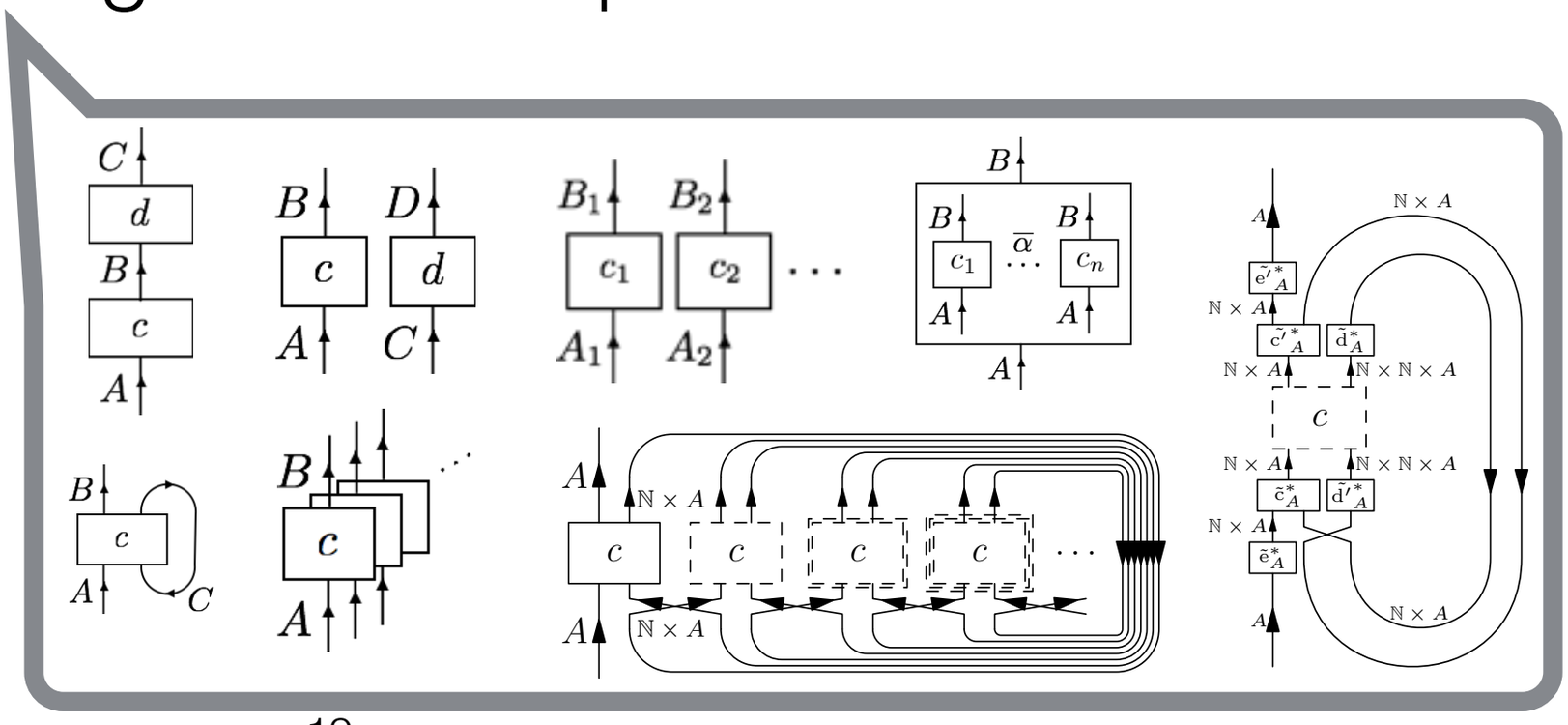
Memoryful GoI

effectful PCF terms

compositional translation

transducers

via coalgebraic component calculus



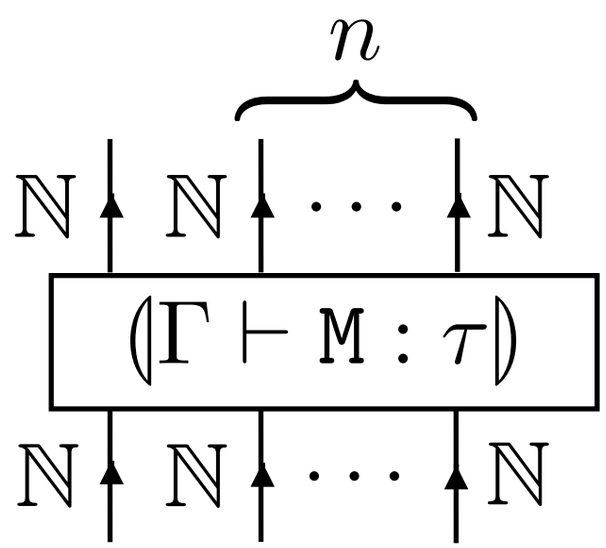
Translation

Def. (translation $(\!|\Gamma \vdash \mathtt{M} : \tau|\!)$)

For a type judgement $(\!|\Gamma \vdash \mathtt{M} : \tau|\!)$ $(\Gamma = \mathtt{x}_1 : \tau_1, \ldots, \mathtt{x}_n : \tau_n)$

we inductively define

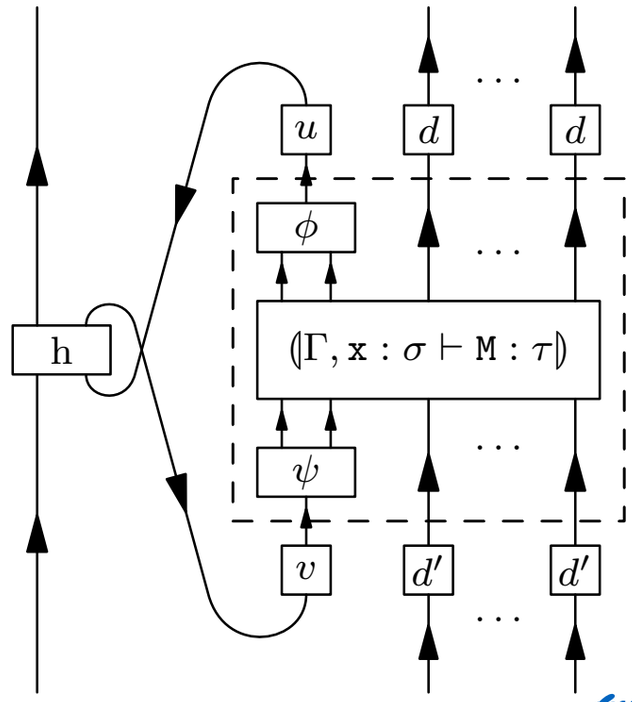
$$(\!|\Gamma \vdash \mathtt{M} : \tau|\!) \;=$$



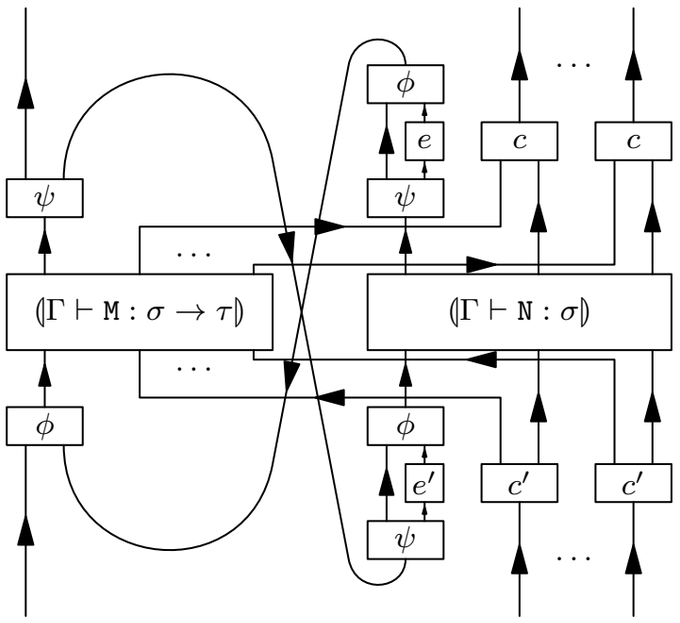
Translation

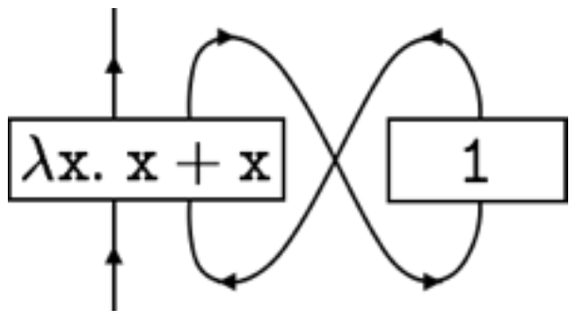
Def. (translation $(\!|\Gamma \vdash \mathtt{M} : \tau|\!)$)

$(\!|\Gamma \vdash \lambda \mathtt{x} : \sigma . \mathtt{M} : \sigma \to \tau|\!) =$



$(\!|\Gamma \vdash \mathtt{M}\ \mathtt{N} : \tau|\!) =$

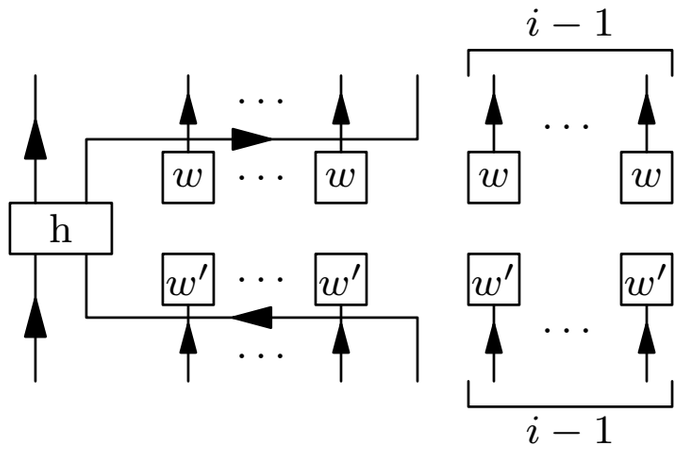




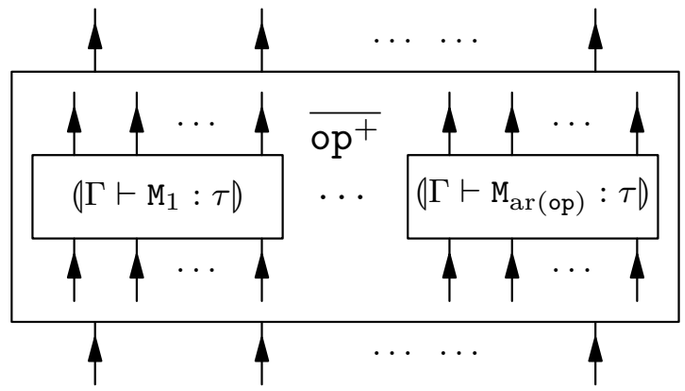
Translation

Def. (translation $(\!|\Gamma \vdash \mathtt{M} : \tau|\!)$)

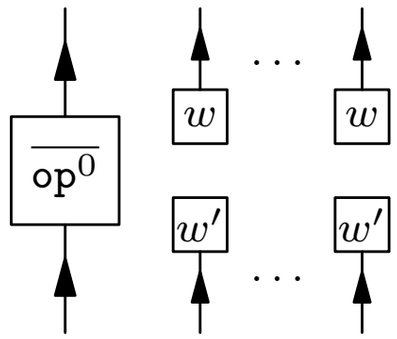
$$(\!|\Gamma \vdash \mathbf{x}_i : \tau_i|\!) =$$



$$(\!|\Gamma \vdash \mathtt{op}^{+}(\mathtt{M}_1, \ldots, \mathtt{M}_{\mathrm{ar}(\mathtt{op})}) : \tau|\!) =$$



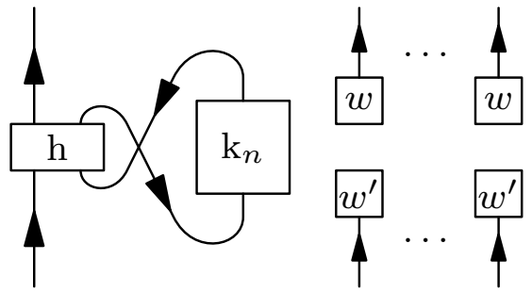
$$(\!|\Gamma \vdash \mathtt{op}^{0}() : \tau|\!) =$$



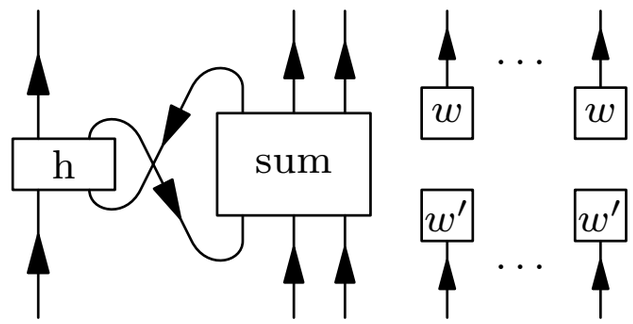
Translation

Def. (translation $(\!|\Gamma \vdash \mathtt{M} : \tau|\!)$)

$(\!|\Gamma \vdash \underline{\mathtt{n}} : \mathtt{nat}|\!) =$

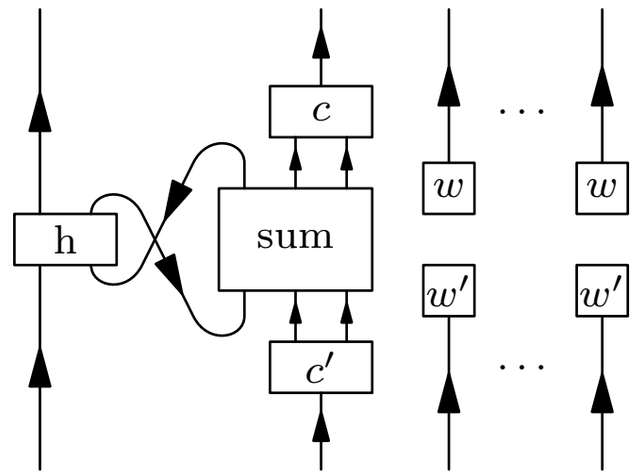


$(\!|\Gamma, \mathtt{x} : \mathtt{nat}, \mathtt{y} : \mathtt{nat} \vdash \mathtt{x} + \mathtt{y} : \mathtt{nat}|\!) =$



(if $\mathtt{x} \not\equiv \mathtt{y}$)

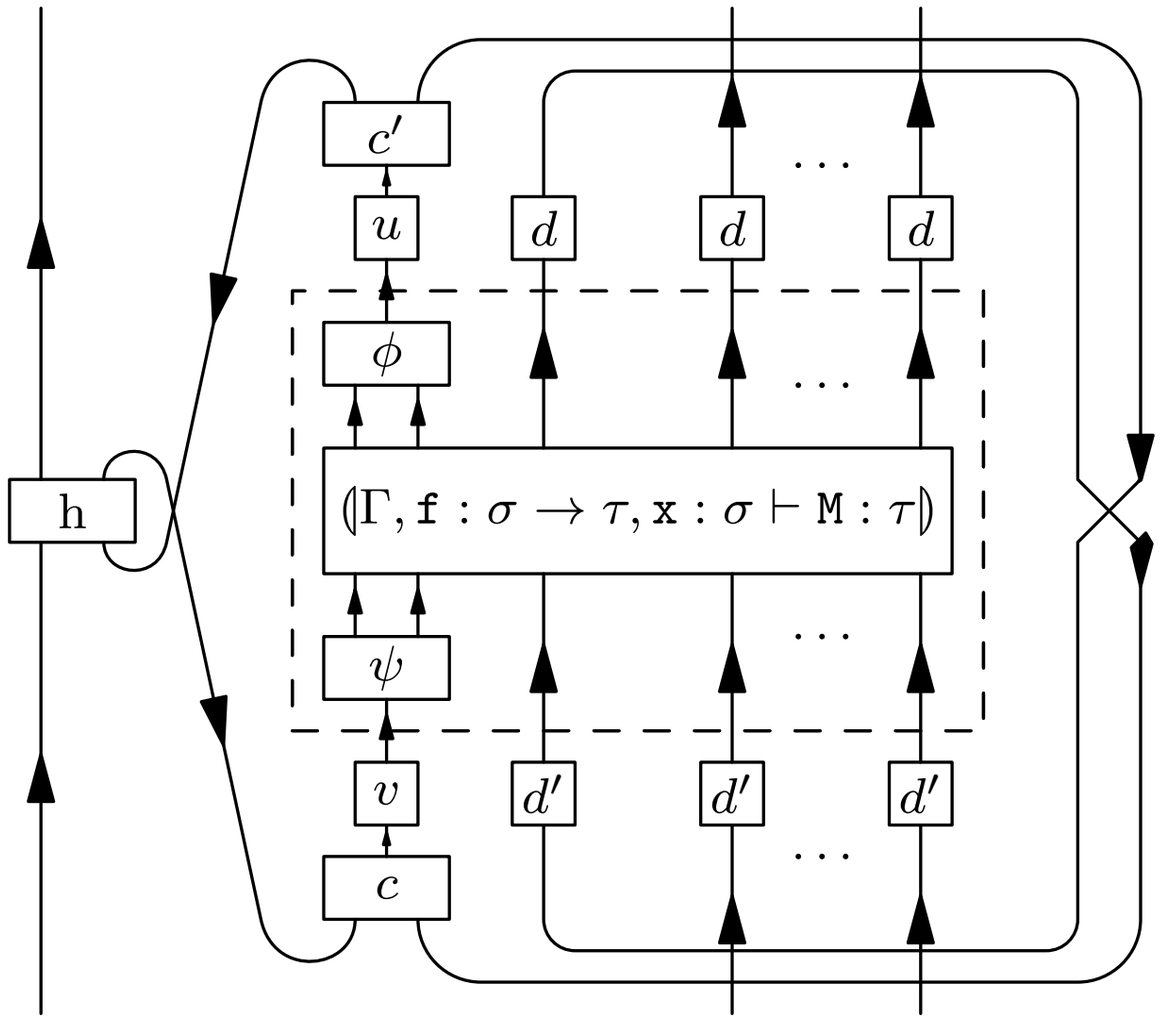
$(\!|\Gamma, \mathtt{x} : \mathtt{nat} \vdash \mathtt{x} + \mathtt{x} : \mathtt{nat}|\!) =$

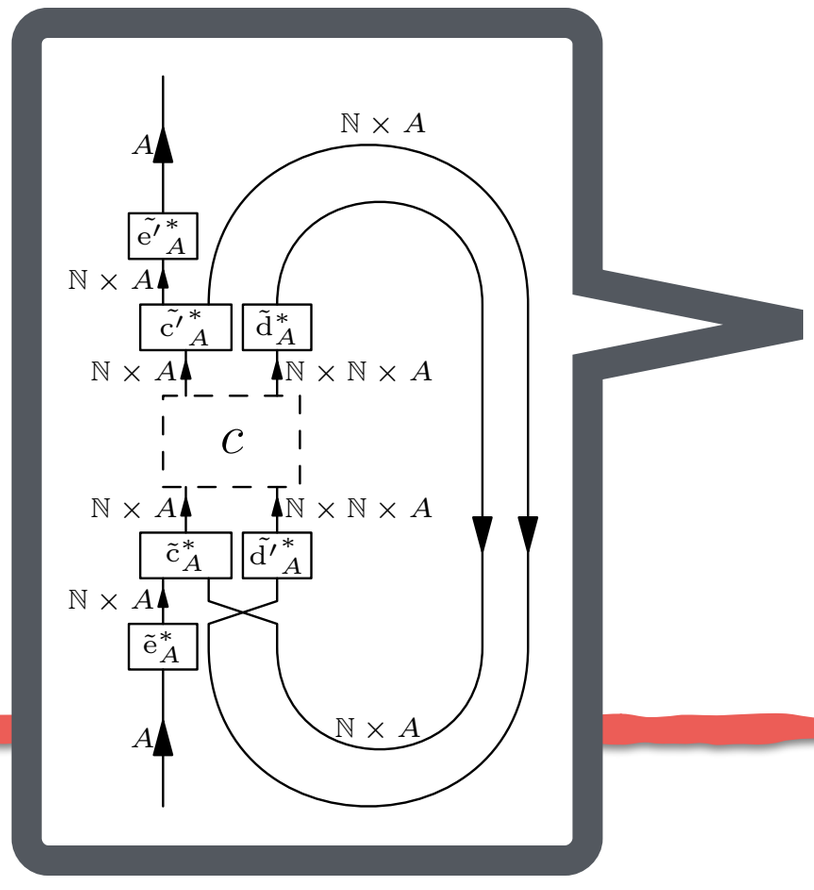


Translation

Def. (translation $(\!|\Gamma \vdash \mathtt{M} : \tau|\!)$)

$$(\!|\Gamma \vdash \mathtt{rec}(\mathtt{f} : \sigma \to \tau, \mathtt{x} : \sigma).\mathtt{M} : \sigma \to \tau|\!) =$$



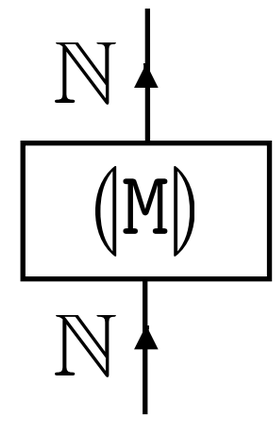


Translation

Theorem 6.4 (adequacy)**.** Any closed term $\mathtt{M}$ of base type $\mathtt{nat}$ satisfies $[\![|\mathtt{M}|]\!] = (\!|\mathtt{M}|\!)^{\dagger}$.

evaluation result of term $\mathtt{M}$

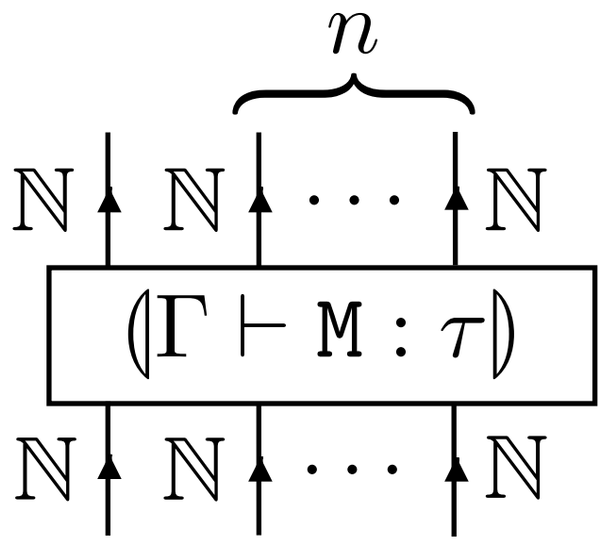
output of transducer



Memoryful GoI

effectful PCF terms

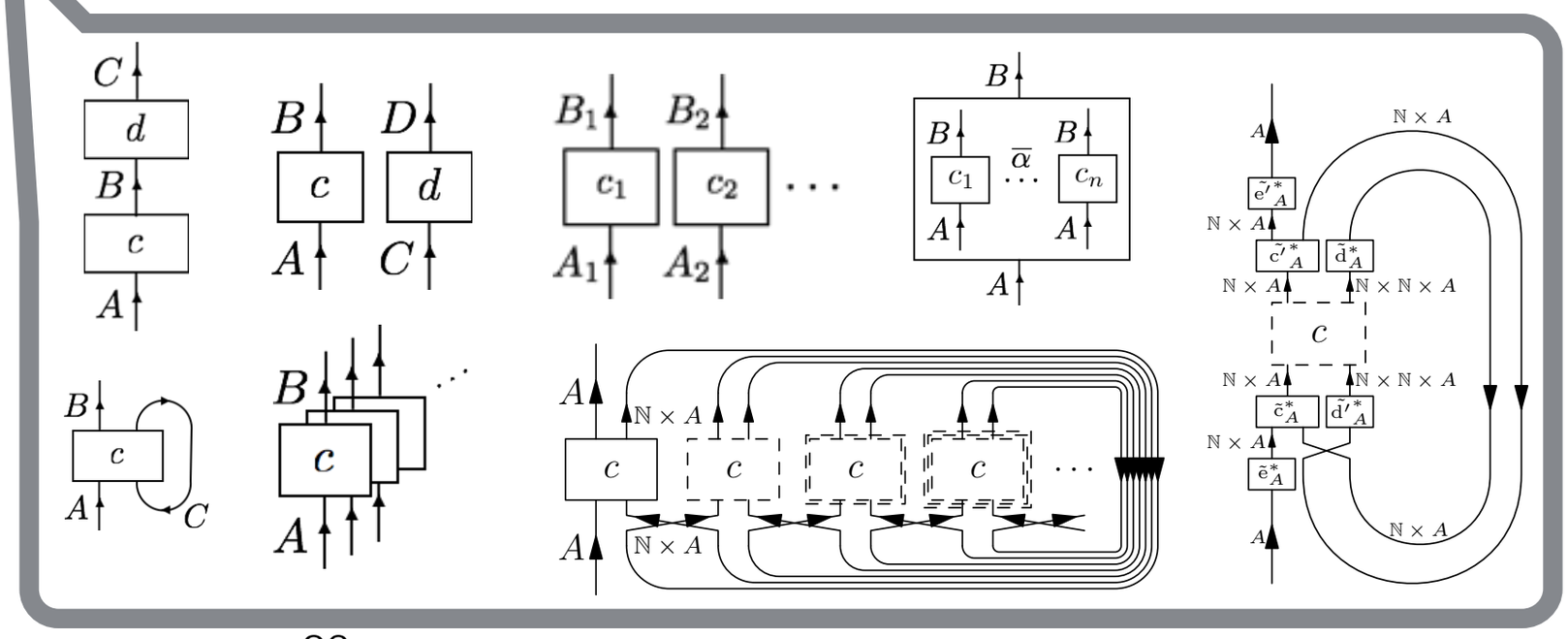
$$(\!|\Gamma \vdash \mathtt{M} : \tau|\!) \;=$$



adequate translation

transducers

via coalgebraic component calculus



Example: Recursive Probabilistic Program

$$(\texttt{rec}(\texttt{flipLoop}, \texttt{x}).\, \texttt{choose}_{0.4}(\texttt{x},\ \texttt{flipLoop}\,(\texttt{x}+1)))\ \ 0$$

coin flips	return value	probability
H	0	0.4
TH	1	$0.4 * 0.6^2$
TTH	2	$0.4 * 0.6^3$
TTTH	3	$0.4 * 0.6^4$

Example: Recursive Probabilistic Program

$$(\texttt{rec}(\texttt{flipLoop}, \texttt{x}).\, \texttt{choose}_{0.4}(\texttt{x},\ \texttt{flipLoop}\,(\texttt{x}+1)))\ \ \texttt{0}$$

coin flips		probability
H	0	0.4
TH	1	$0.4 * 0.6^2$
TTH	2	$0.4 * 0.6^3$
TTTH	3	$0.4 * 0.6^4$

Example: Recursive Probabilistic Program

$$(\texttt{rec}(\texttt{flipLoop}, \texttt{x}).\, \texttt{choose}_{0.4}(\texttt{x},\ \texttt{flipLoop}\,(\texttt{x}+1)))\ \ \texttt{0}$$

$$\text{evaluation result of term } \texttt{M} = \begin{bmatrix} 0 & \mapsto & 0.4 \\ 1 & \mapsto & 0.4 \times 0.6 \\ 2 & \mapsto & 0.4 \times 0.6^2 \\ 3 & \mapsto & 0.4 \times 0.6^3 \\ & \vdots & \end{bmatrix} \in \mathcal{D}_{\leq 1}(\mathbb{N})$$

Example: Recursive Probabilistic Program

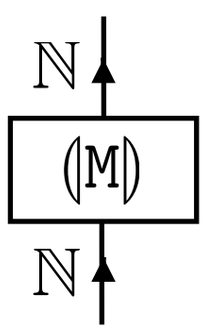
$$(\mathtt{rec}(\mathtt{flipLoop}, \mathtt{x}).\, \mathtt{choose}_{0.4}(\mathtt{x},\ \mathtt{flipLoop}\,(\mathtt{x}+\mathtt{1})))\ \mathtt{0}$$

evaluation result of term $\mathtt{M}$

$$= \begin{bmatrix} 0 & \mapsto & 0.4 \\ 1 & \mapsto & 0.4 \times 0.6 \\ 2 & \mapsto & 0.4 \times 0.6^2 \\ 3 & \mapsto & 0.4 \times 0.6^3 \\ & \vdots & \end{bmatrix} \in \mathcal{D}_{\leq 1}(\mathbb{N})$$

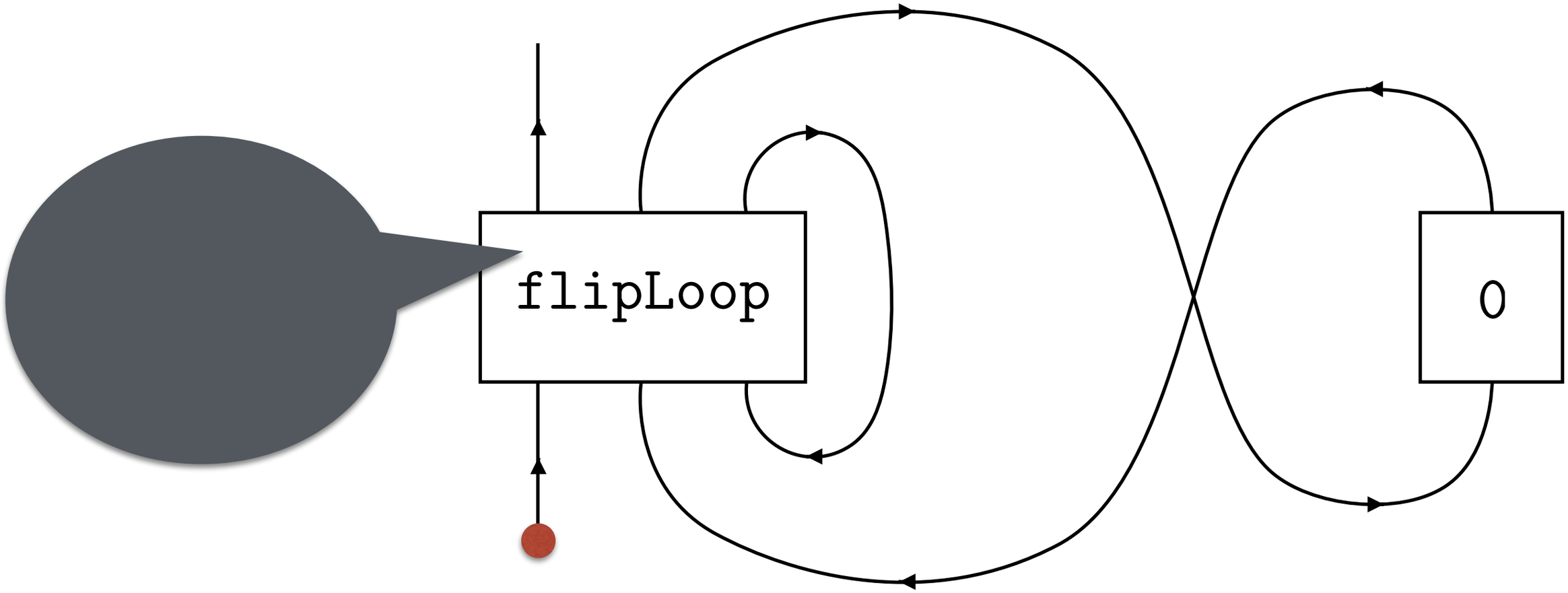
||

output of transducer



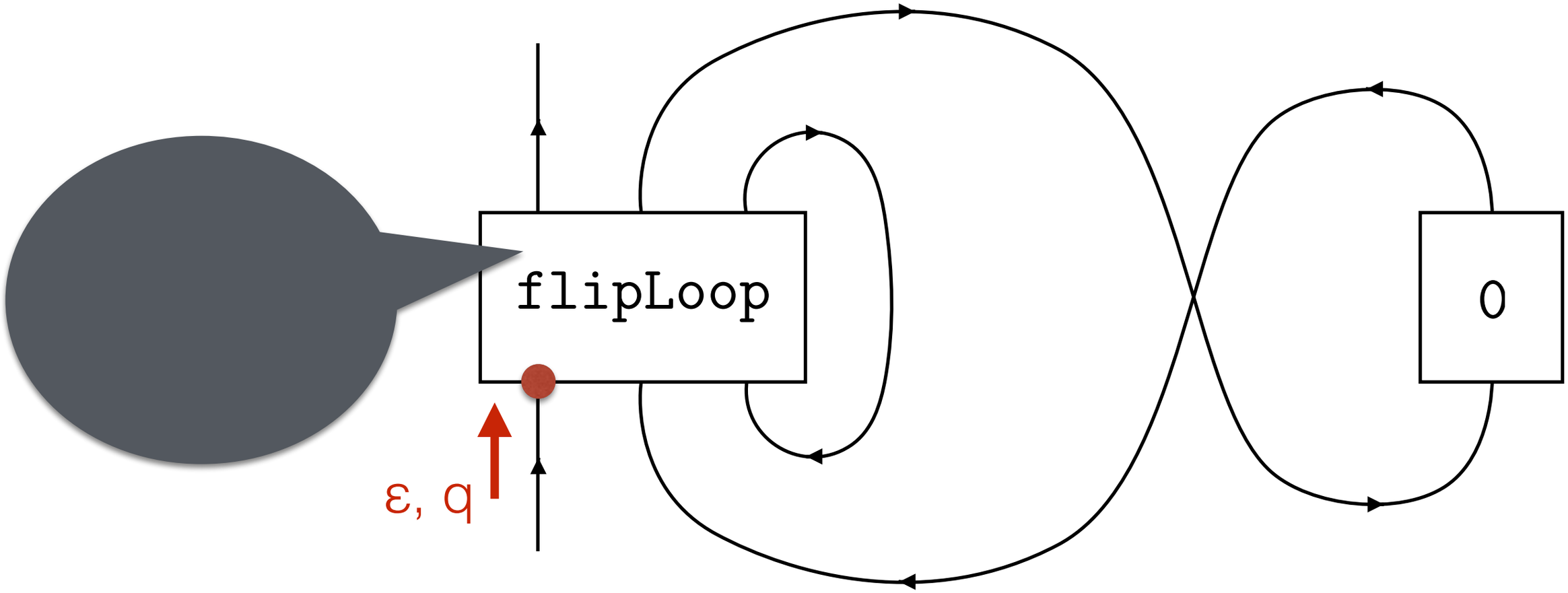
Example: Recursive Probabilistic Program

$$(\texttt{rec}(\texttt{flipLoop}, \texttt{x}).\, \texttt{choose}_{0.4}(\texttt{x},\ \texttt{flipLoop}\,(\texttt{x}+1)))\ \ \texttt{0}$$



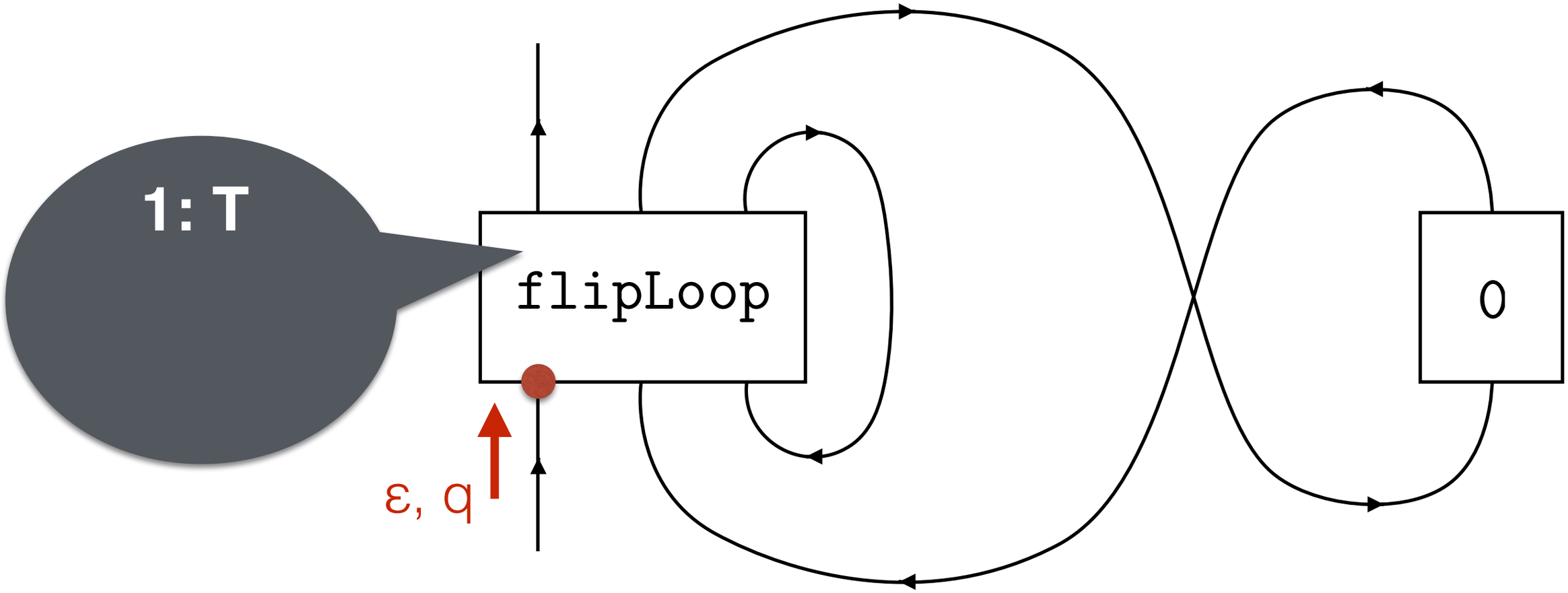
Example: Recursive Probabilistic Program

$$(\texttt{rec}(\texttt{flipLoop}, \texttt{x}).\, \texttt{choose}_{0.4}(\texttt{x},\ \texttt{flipLoop}\,(\texttt{x}+1)))\ \ \texttt{0}$$



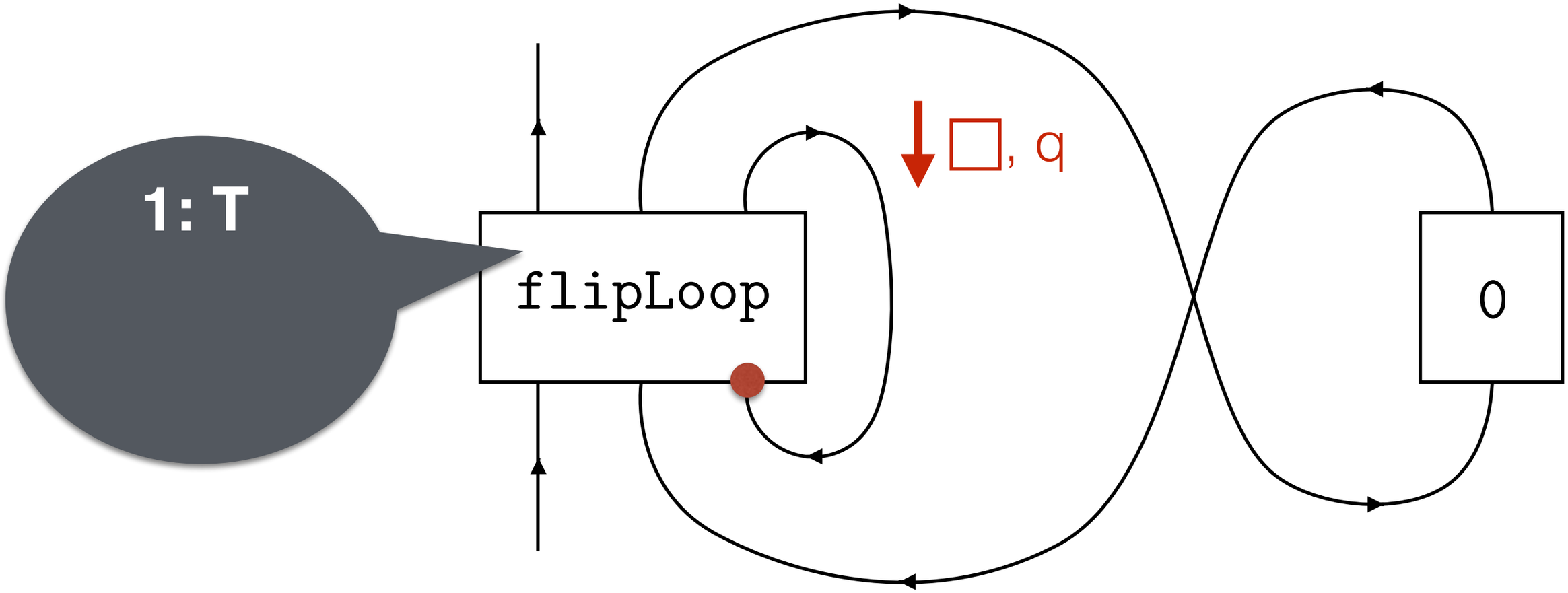
Example: Recursive Probabilistic Program

$$(\texttt{rec}(\texttt{flipLoop}, \texttt{x}).\, \texttt{choose}_{0.4}(\texttt{x},\ \texttt{flipLoop}\,(\texttt{x}+1)))\ \ 0$$



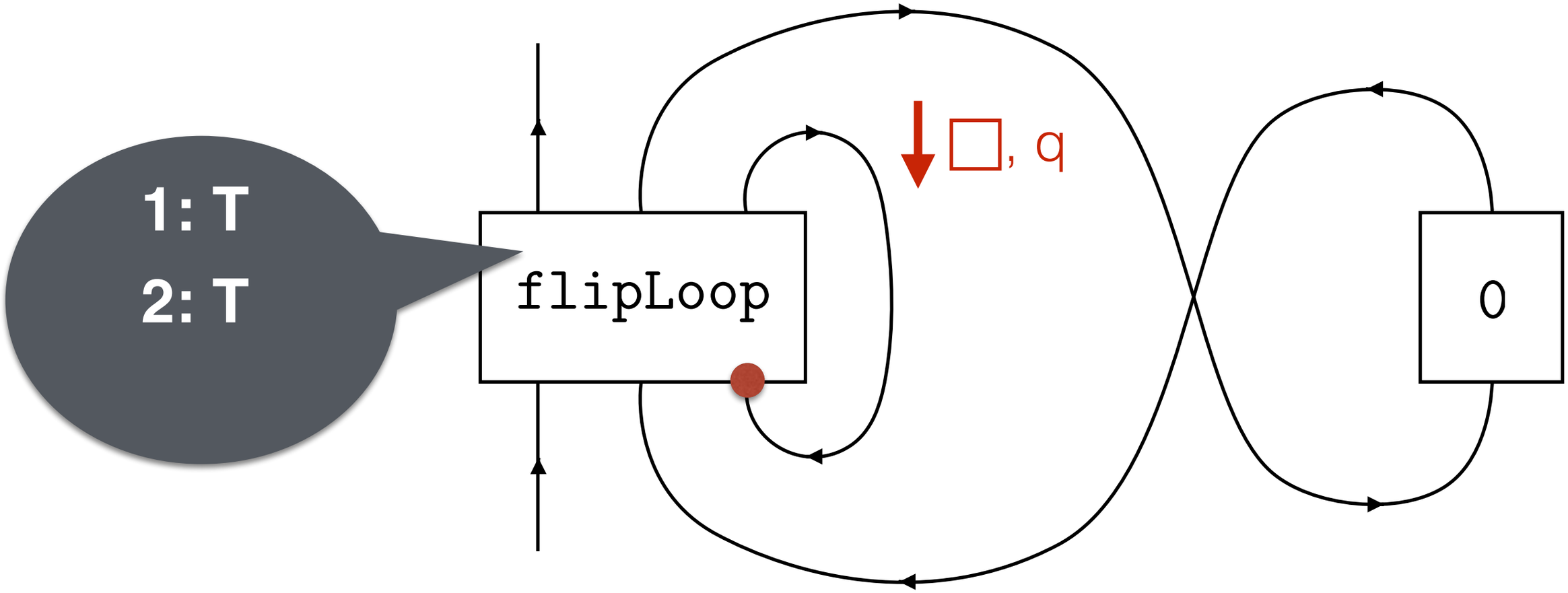
Example: Recursive Probabilistic Program

$$(\texttt{rec}(\texttt{flipLoop}, \texttt{x}).\, \texttt{choose}_{0.4}(\texttt{x},\ \texttt{flipLoop}\,(\texttt{x}+1)))\ \ 0$$



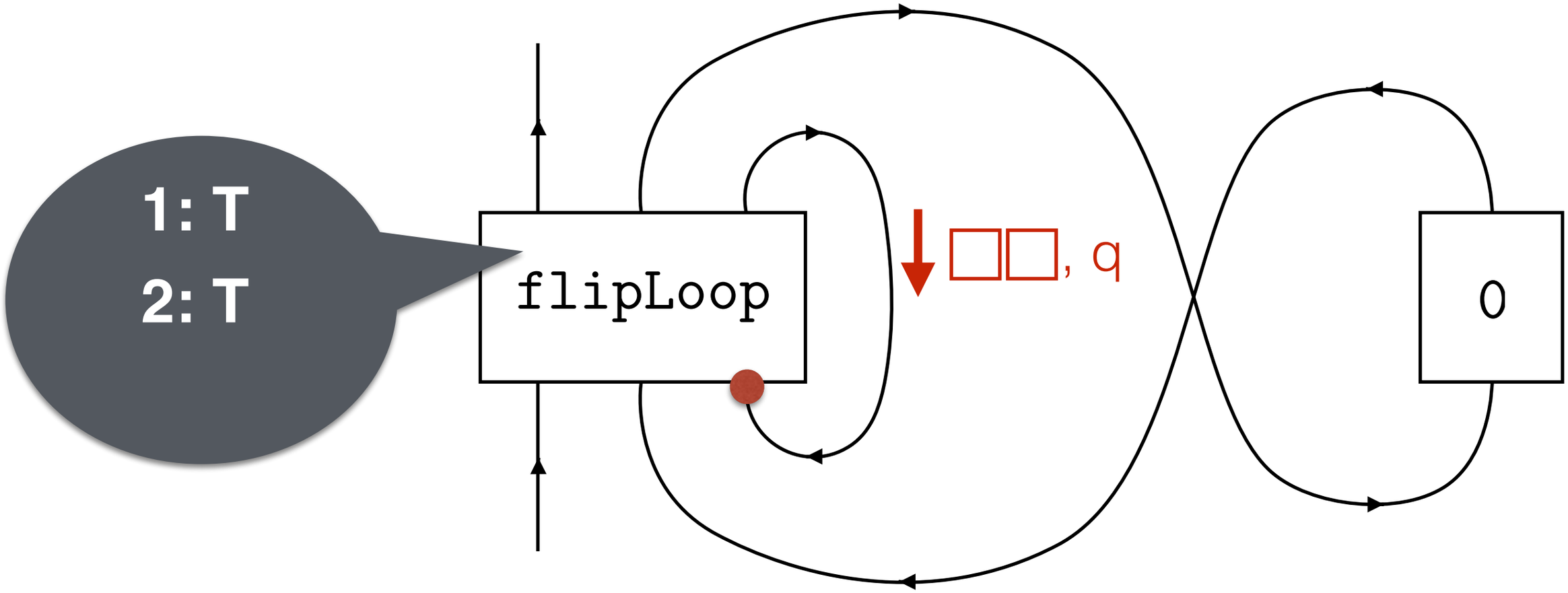
Example: Recursive Probabilistic Program

$$(\texttt{rec}(\texttt{flipLoop}, \texttt{x}).\, \texttt{choose}_{0.4}(\texttt{x},\ \texttt{flipLoop}\,(\texttt{x}+1)))\ \ 0$$



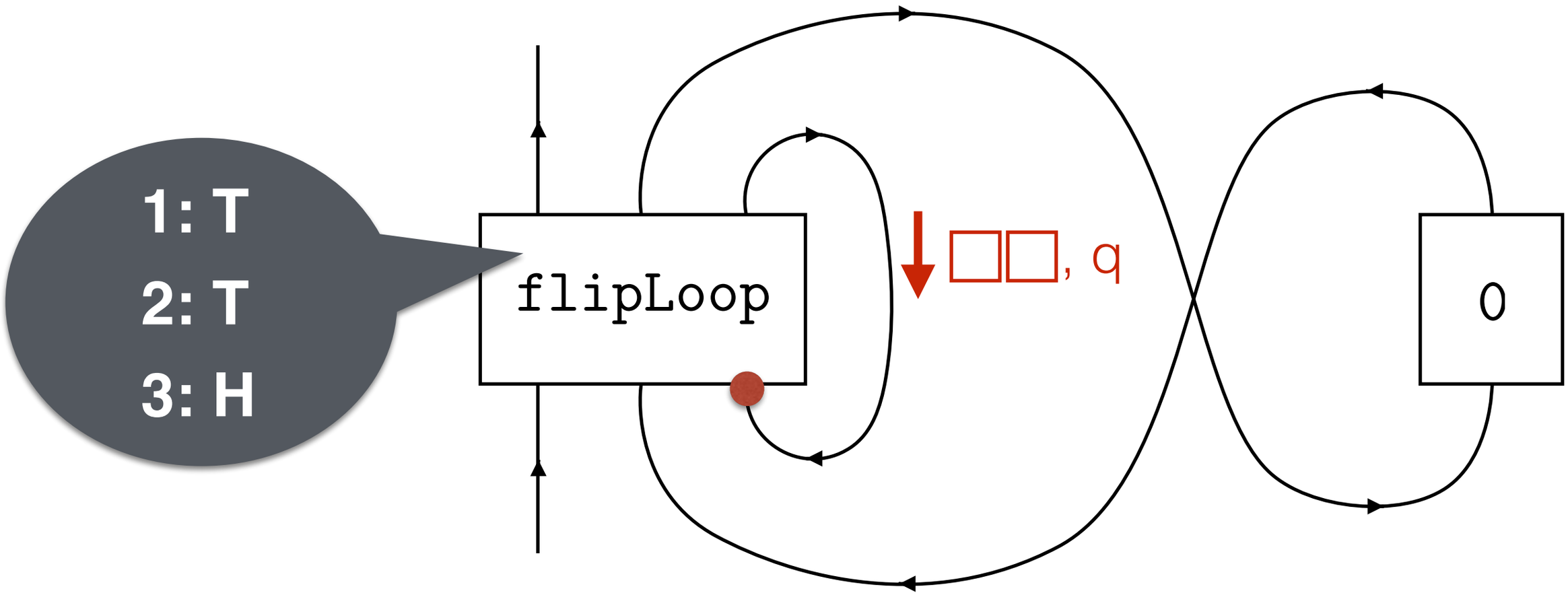
Example: Recursive Probabilistic Program

$$(\texttt{rec}(\texttt{flipLoop}, \texttt{x}).\, \texttt{choose}_{0.4}(\texttt{x},\ \texttt{flipLoop}\,(\texttt{x}+1)))\ \ 0$$



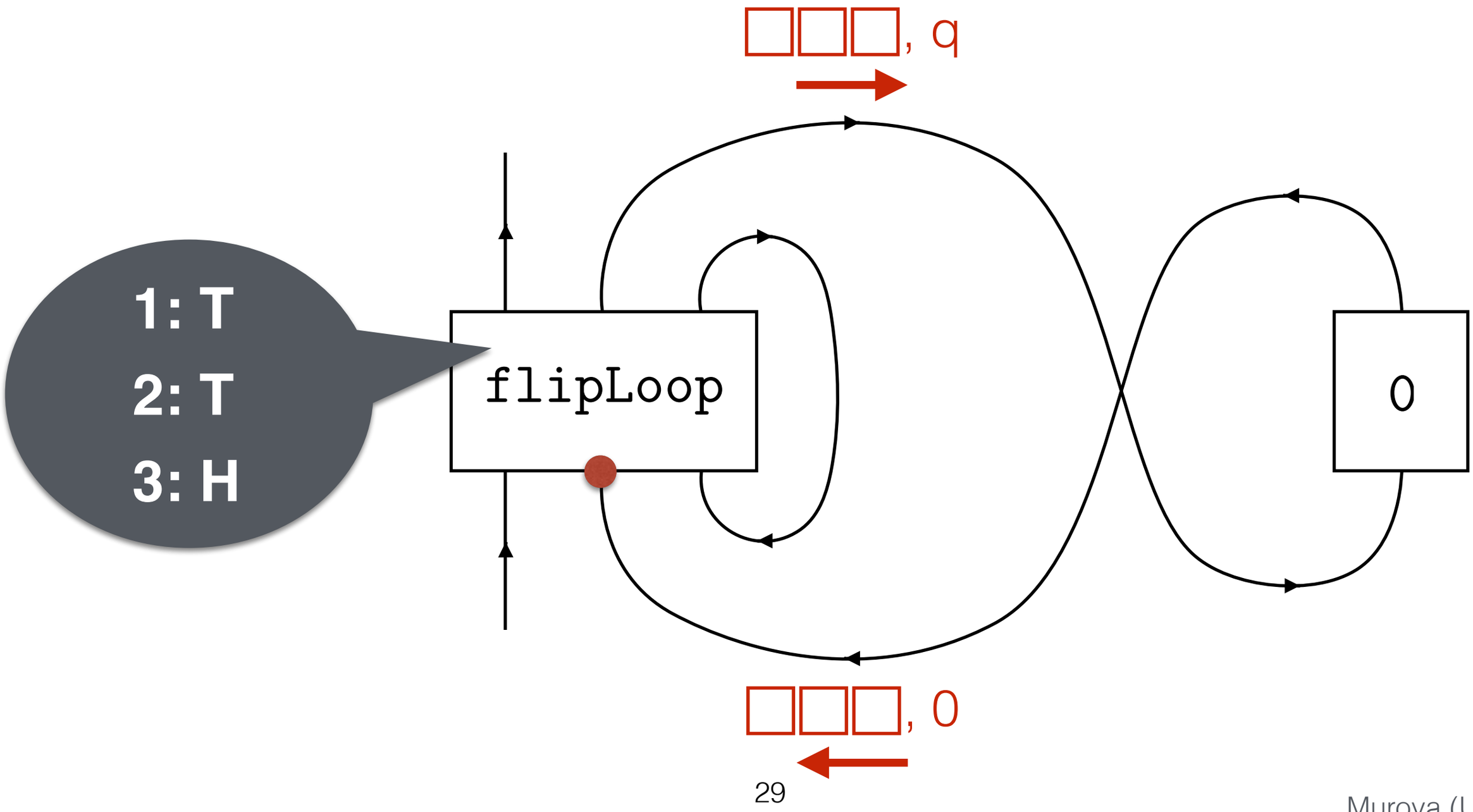
Example: Recursive Probabilistic Program

$$(\texttt{rec}(\texttt{flipLoop}, \texttt{x}).\, \texttt{choose}_{0.4}(\texttt{x},\ \texttt{flipLoop}\,(\texttt{x}+1)))\ \ 0$$



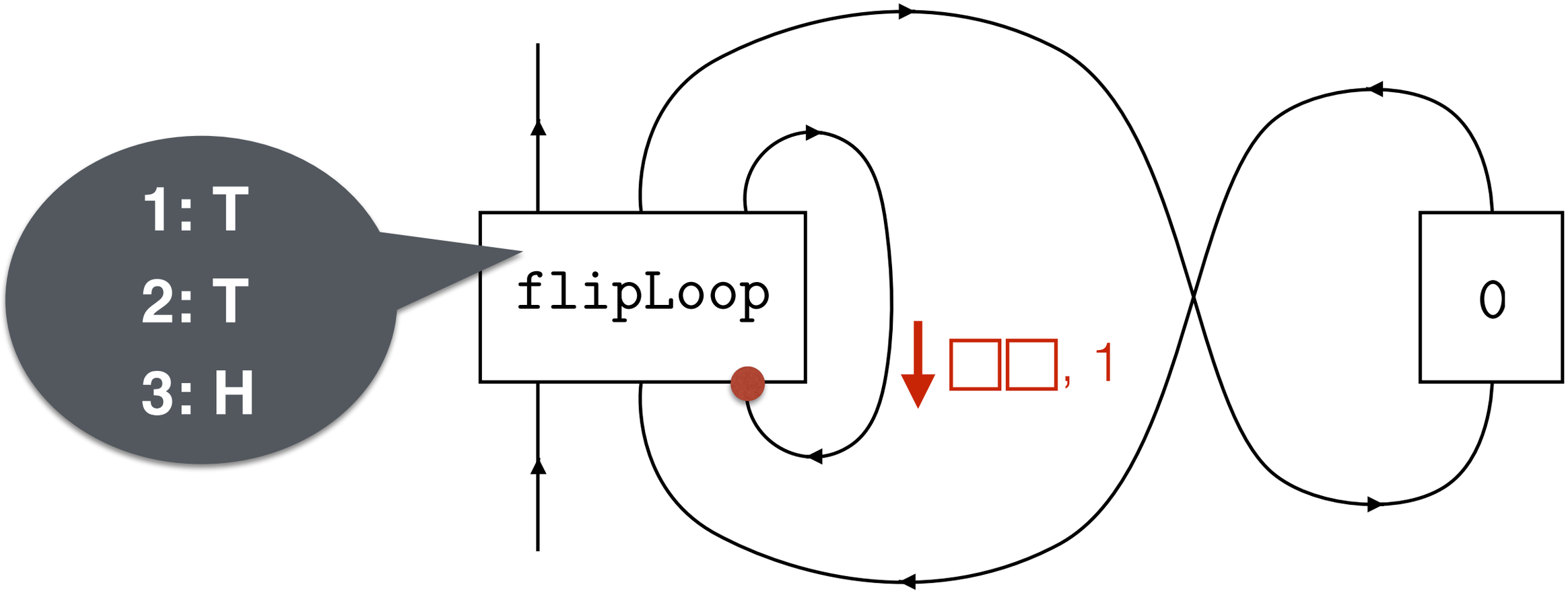
Example: Recursive Probabilistic Program

$$(\texttt{rec}(\texttt{flipLoop}, \texttt{x}).\, \texttt{choose}_{0.4}(\texttt{x},\ \texttt{flipLoop}\,(\texttt{x}+1)))\ \ \texttt{0}$$



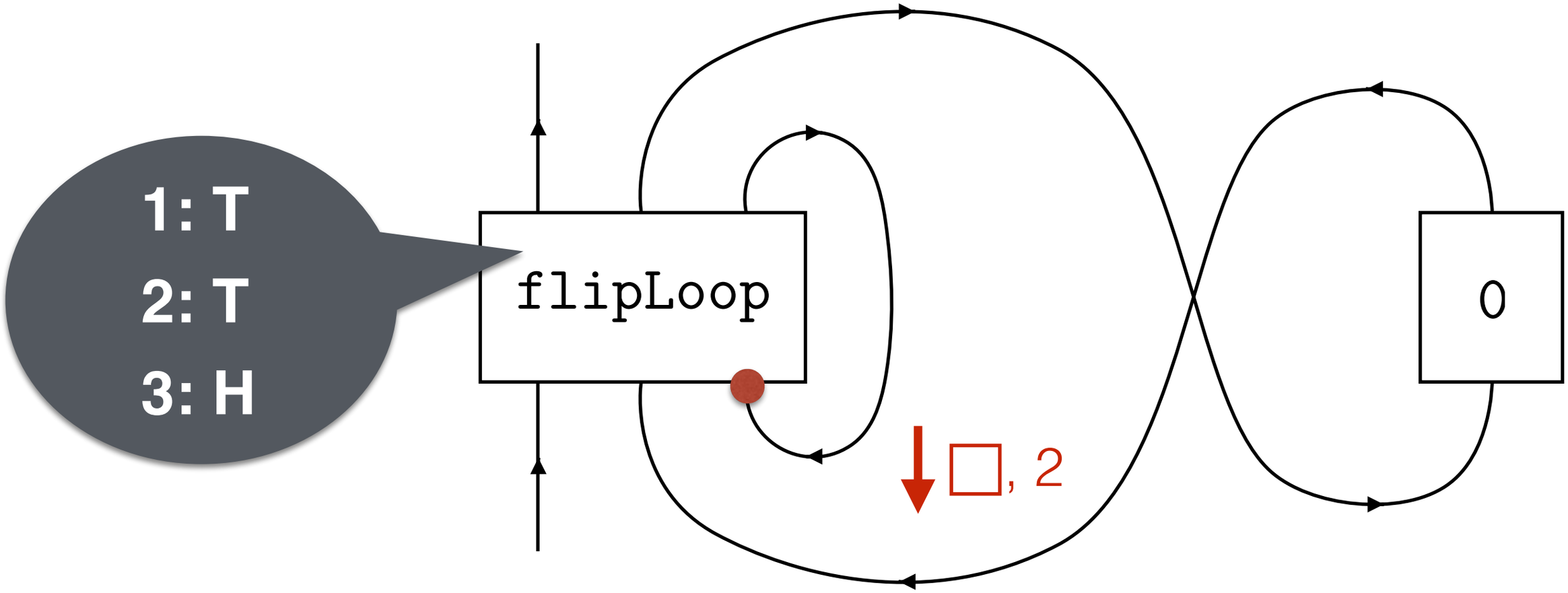
Example: Recursive Probabilistic Program

$$(\texttt{rec}(\texttt{flipLoop}, \texttt{x}).\, \texttt{choose}_{0.4}(\texttt{x},\ \texttt{flipLoop}\,(\texttt{x}+1)))\ \ \texttt{0}$$



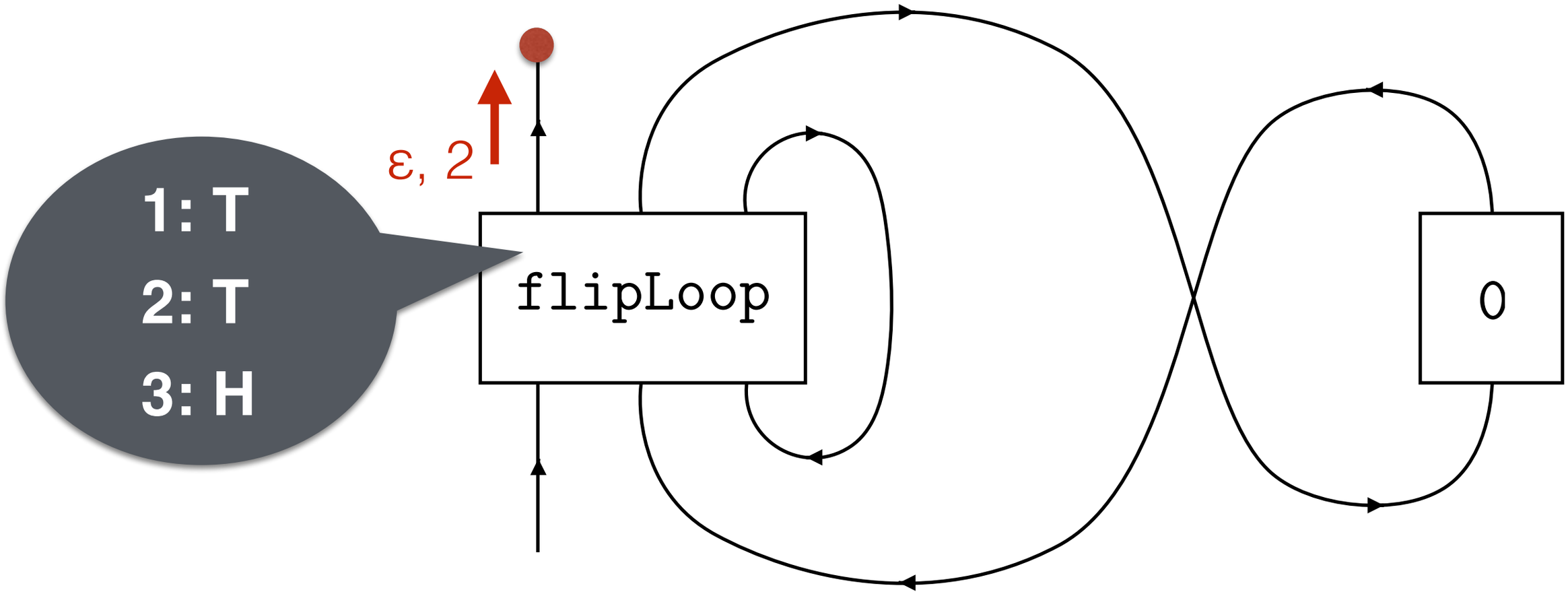
Example: Recursive Probabilistic Program

$$(\texttt{rec}(\texttt{flipLoop}, \texttt{x}).\, \texttt{choose}_{0.4}(\texttt{x},\ \texttt{flipLoop}\,(\texttt{x}+1)))\ \ \texttt{0}$$



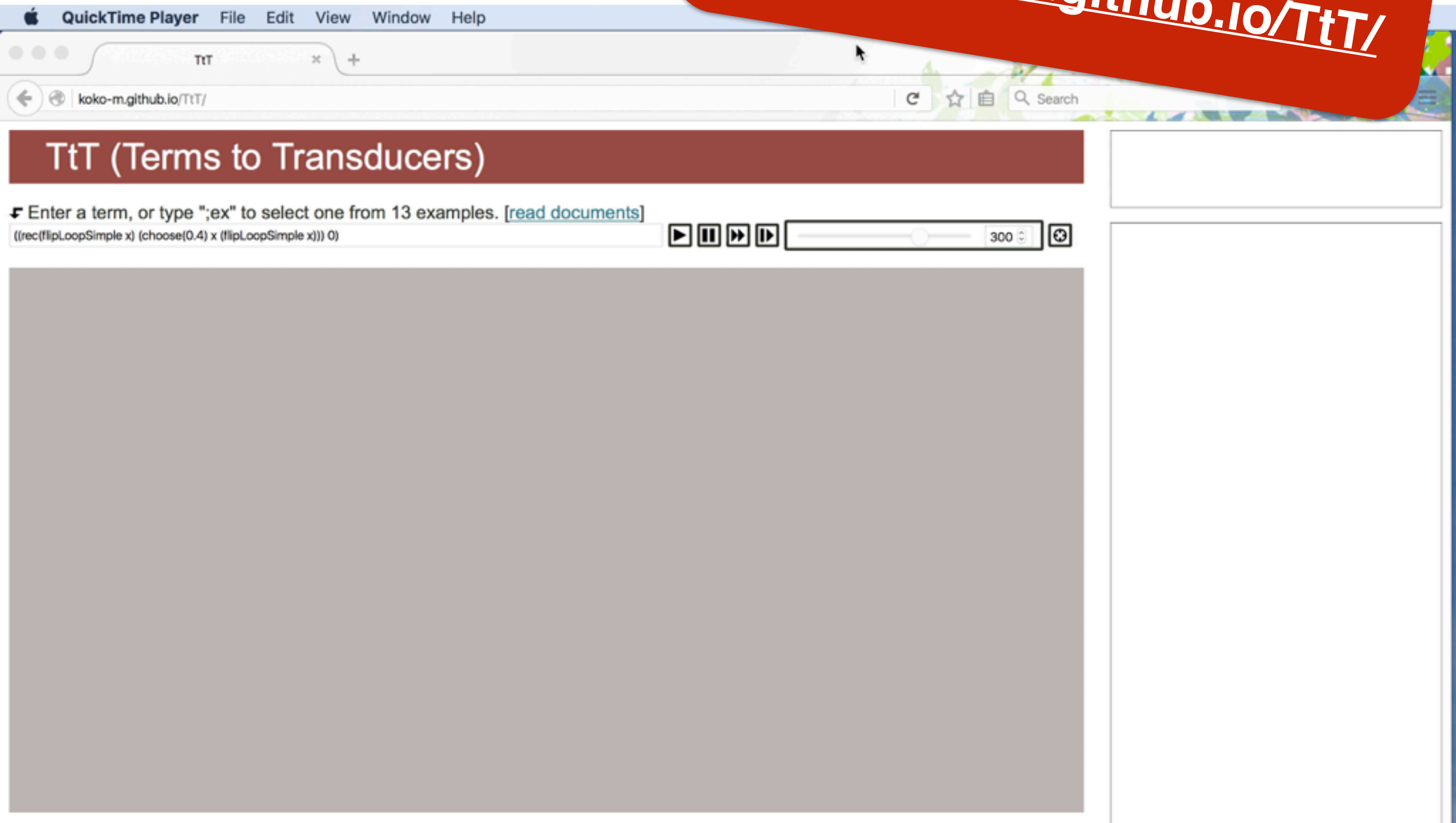
Example: Recursive Probabilistic Program

$$(\texttt{rec}(\texttt{flipLoop}, \texttt{x}).\, \texttt{choose}_{0.4}(\texttt{x},\ \texttt{flipLoop}\,(\texttt{x}+1)))\ \ 0$$



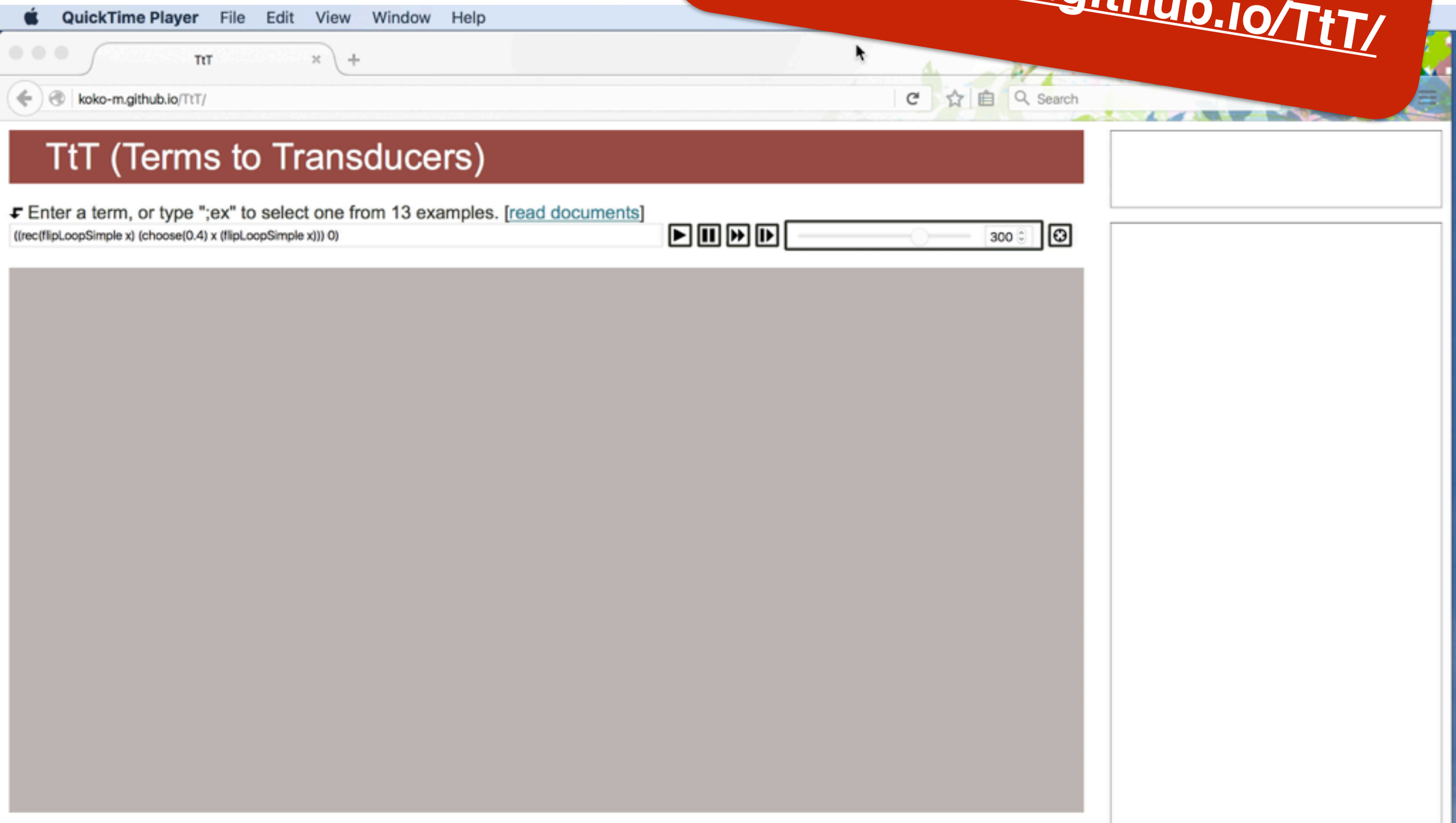
Our Tool ***TtT***

http://koko-m.github.io/TtT/



Our Tool ***TtT***

http://koko-m.github.io/TtT/

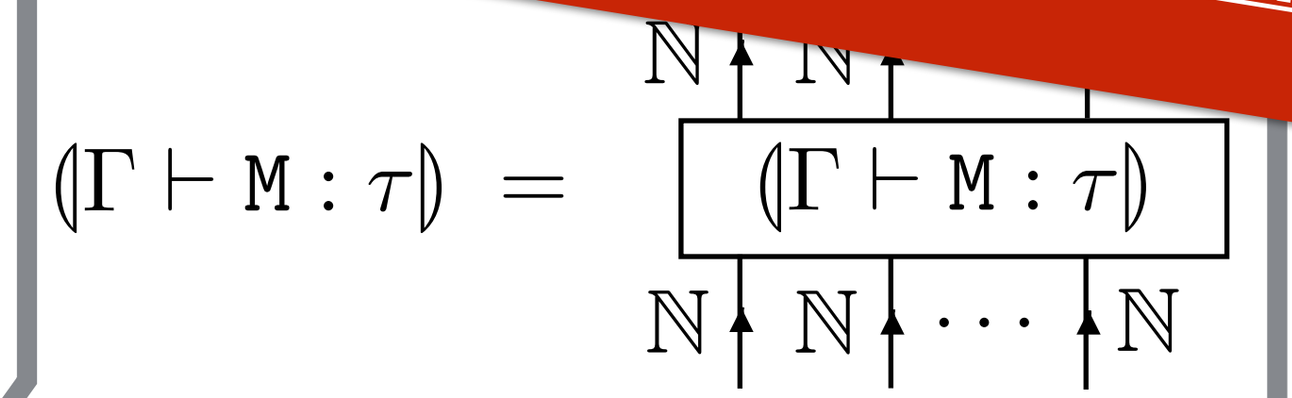


Memoryful GoI

effectful PCF terms

transducers

http://koko-m.github.io/TtT/



adequate translation

via coalgebraic component calculus