Compiling Effectful Terms to Transducers

Prototype Implementation of
Memoryful Geometry of Interaction

Koko Muroya
Toshiki Kataoka
Ichiro Hasuo
(Dept. CS, Univ. Tokyo)

Naohiko Hoshino
(RIMS, Kyoto Univ.)

LOLA (Vienna), July 13, 2014

Compiling Effectful Terms to Transdu

TtT
("Terms to Transducers")

Prototype Implementation of
Memoryful Geometry of Interaction

Koko Muroya
Toshiki Kataoka
Ichiro Hasuo
(Dept. CS, Univ. Tokyo)

Naohiko Hoshino
(RIMS, Kyoto Univ.)

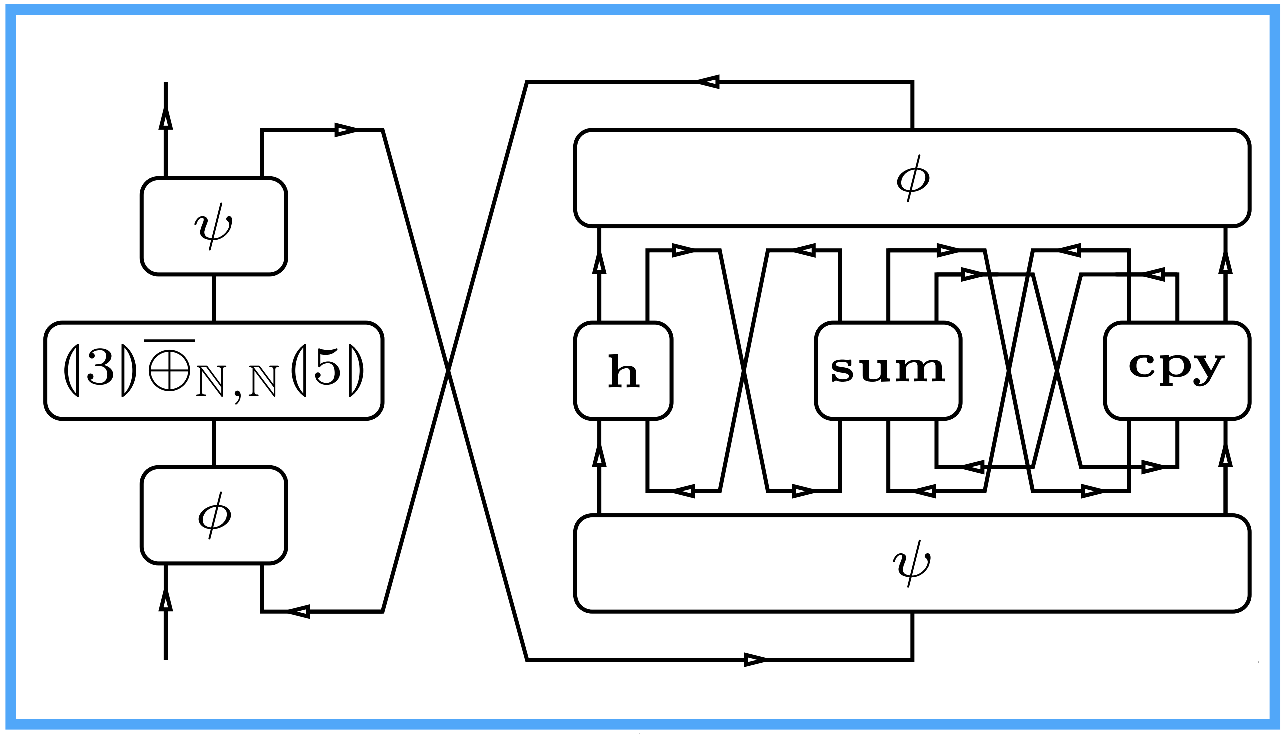
LOLA (Vienna), July 13, 2014

Our Tool *TtT*

"Terms to Transducers"

$(\lambda \mathtt{x} : \mathtt{nat}.\, \mathtt{x} + \mathtt{x})\,(3 \sqcup 5) \quad : \mathtt{nat}$

TtT Compiler



TtT Simulator

terms

transducers

Overview

terms

← λ-terms with algebraic effects

TtT Compiler

← **memoryful GoI**
[Hoshino, —, Hasuo CSL-LICS '14]

transducers

← stream transducers

TtT Simulator

simulation result

Geometry of Interaction (GoI)

- semantics of linear logic proof [Girard ’89],

  functional programming

- token machine presentation [Mackie ’95]

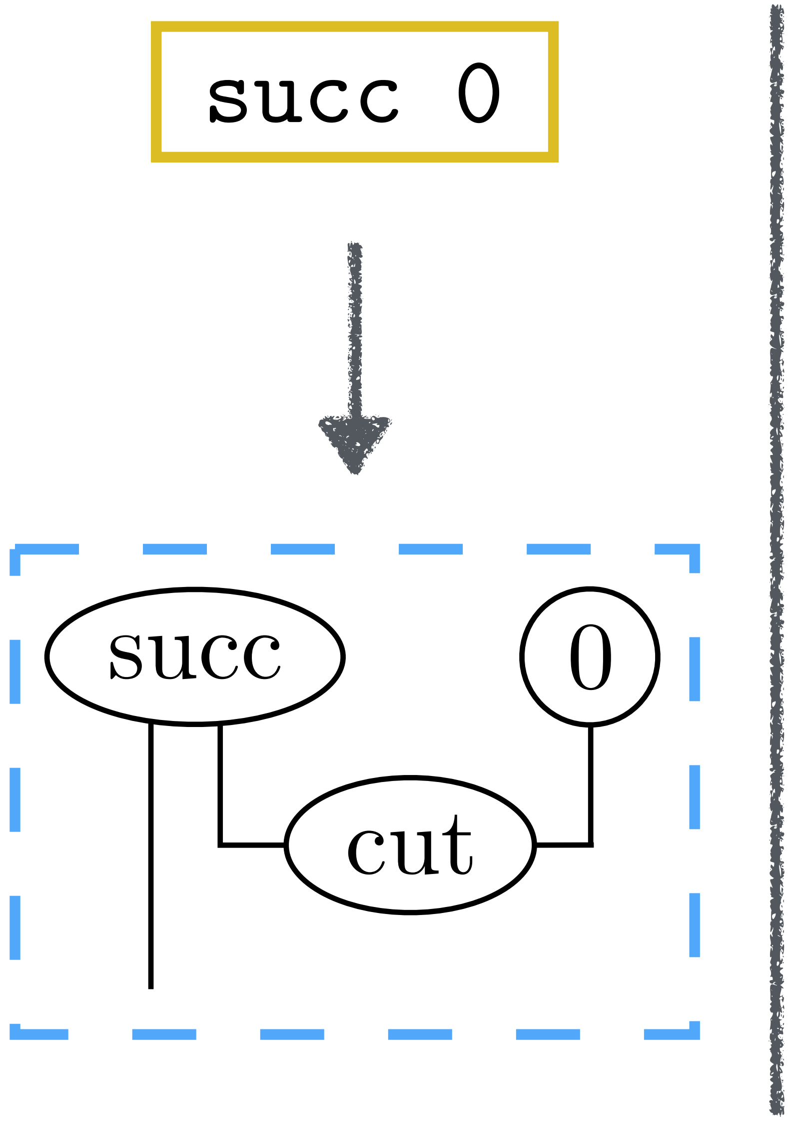
“GoI implementation”

compilation techniques and implementations

[Mackie ’95] [Pinto ’01] [Ghica ’07]

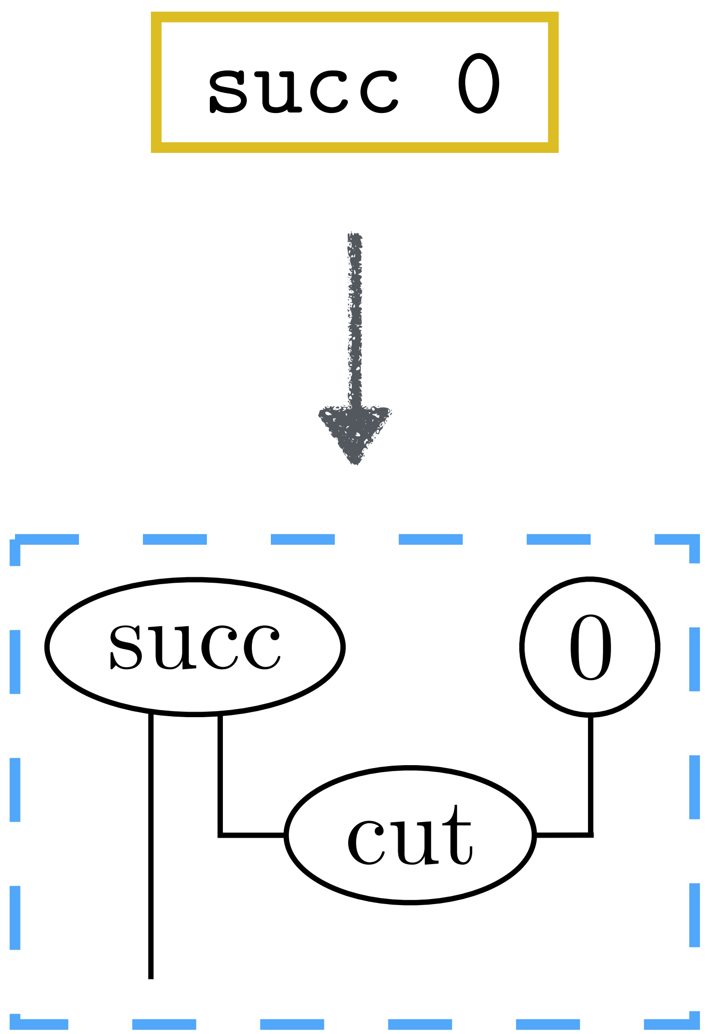
Geometry of Interaction (GoI)

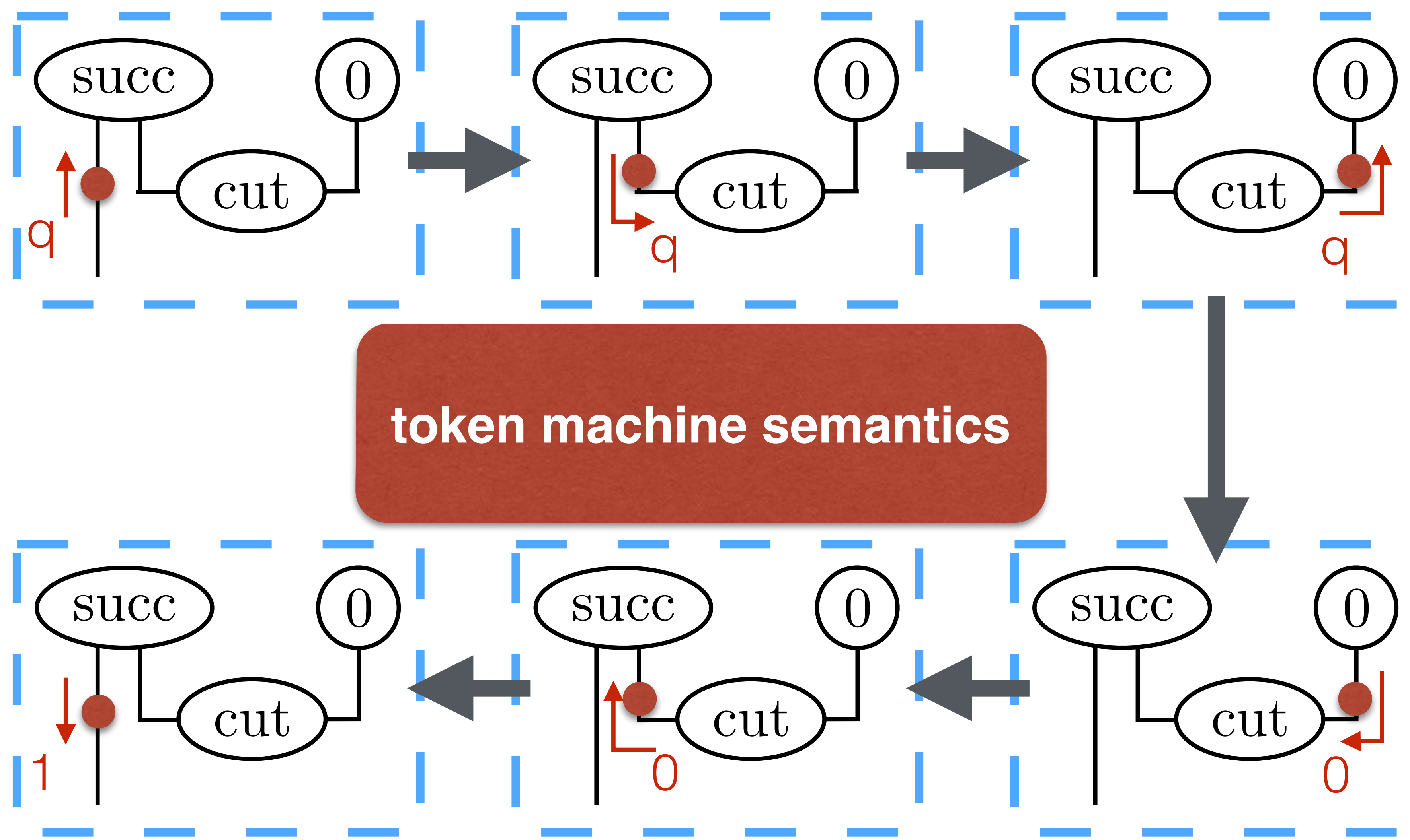
- token machine presentation [Mackie '95]



Geometry of Interaction (GoI)

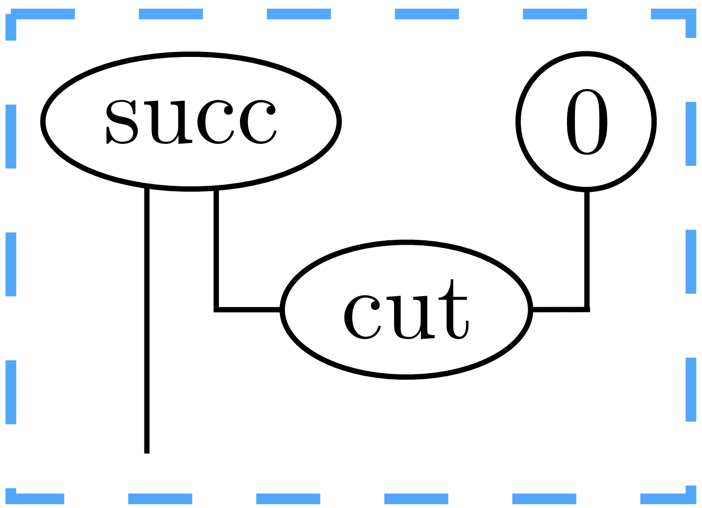
- token machine presentation [Mackie '95]



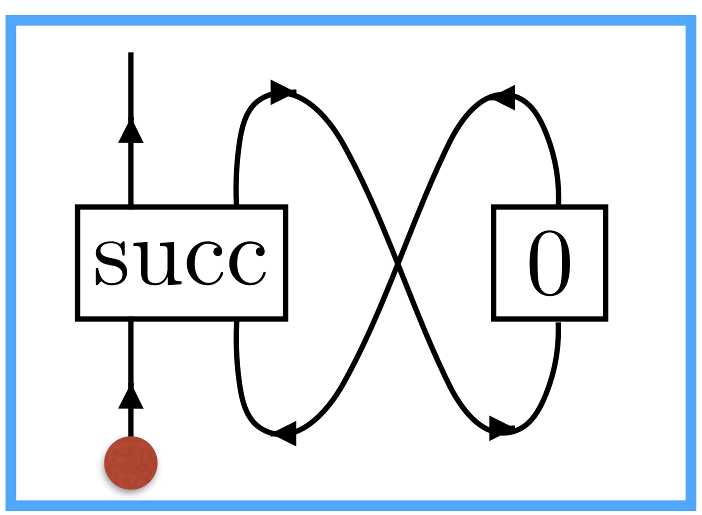


Geometry of Interaction (GoI)

- token machine presentation [Mackie ’95]



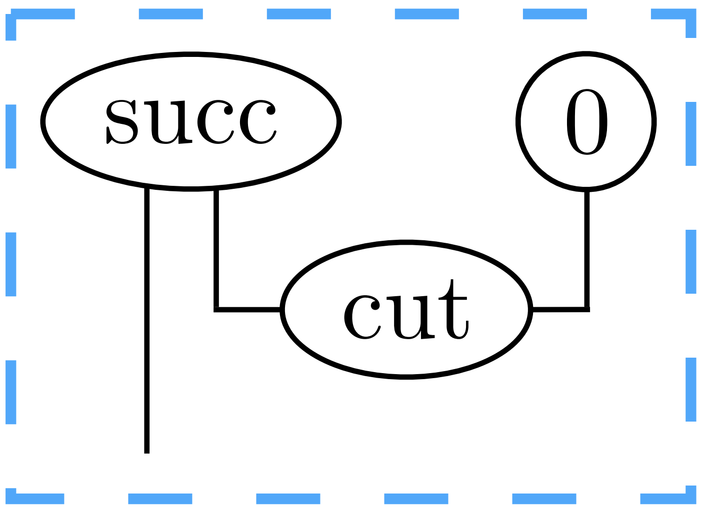
proof net style



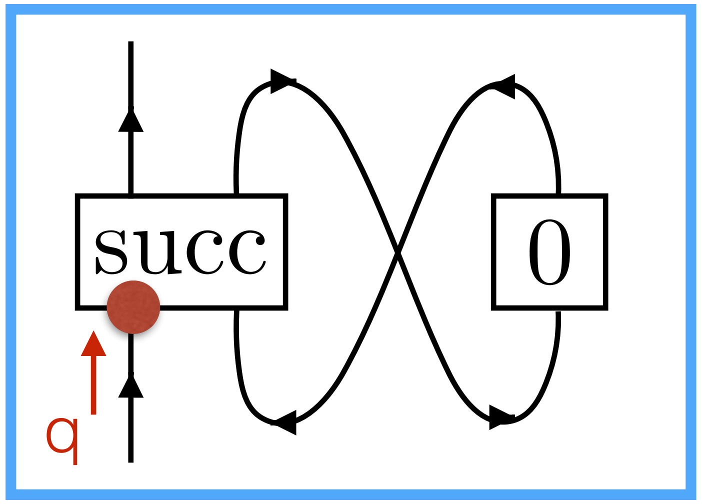
string diagram style
in traced monoidal category

Geometry of Interaction (GoI)

- token machine presentation [Mackie '95]



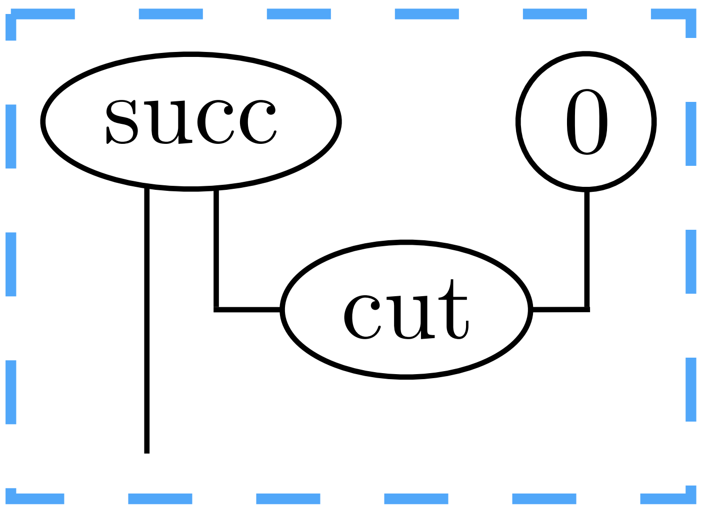
proof net style



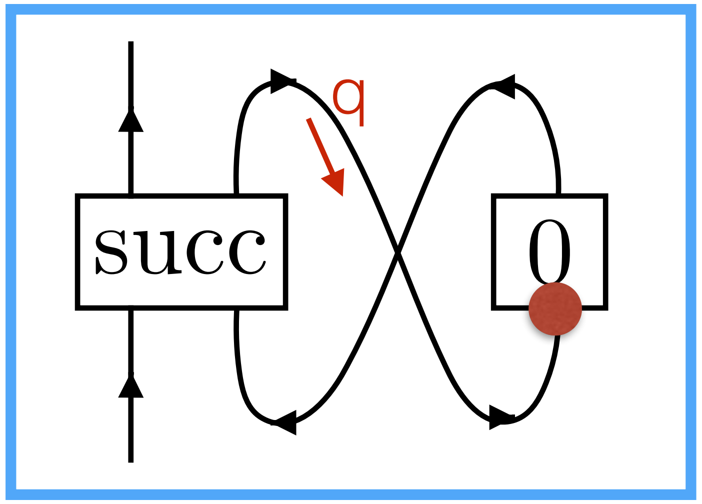
string diagram style
in traced monoidal category

Geometry of Interaction (GoI)

- token machine presentation [Mackie '95]



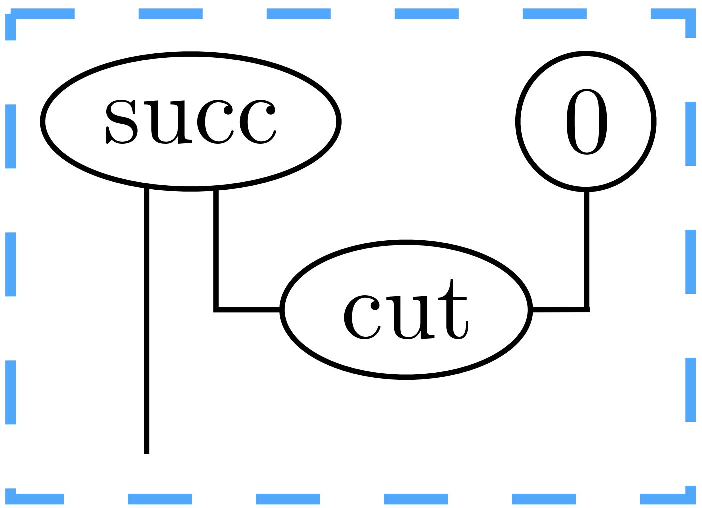
proof net style



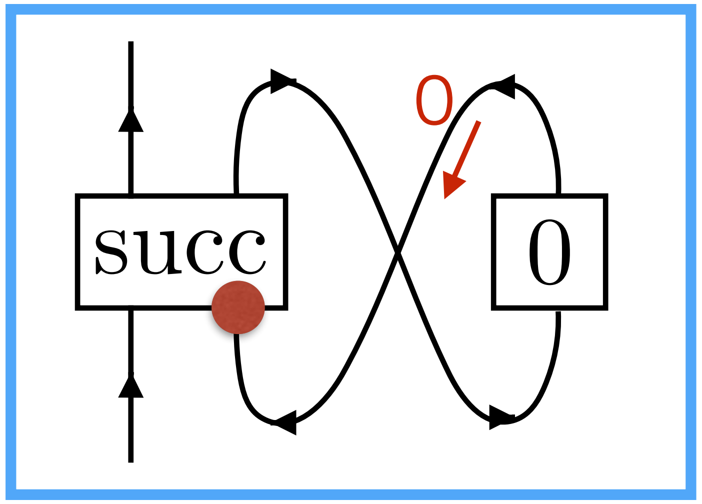
string diagram style
in traced monoidal category

Geometry of Interaction (GoI)

- token machine presentation [Mackie '95]



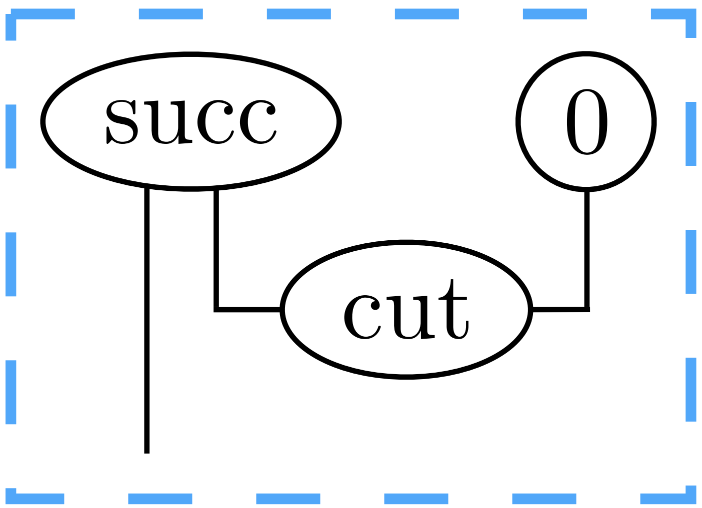
proof net style



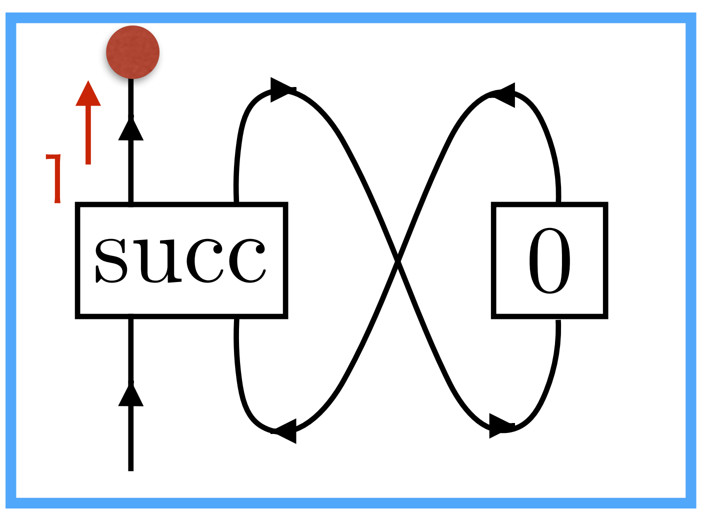
string diagram style
in traced monoidal category

Geometry of Interaction (GoI)

- token machine presentation [Mackie '95]

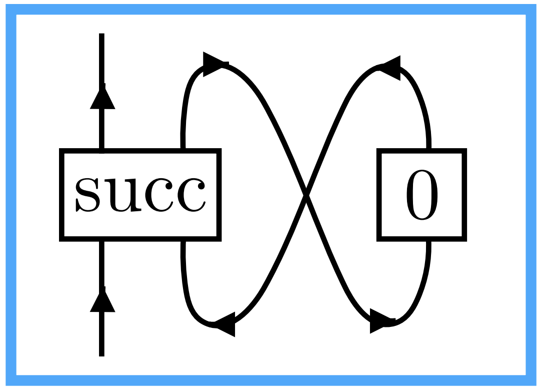


proof net style



string diagram style
in traced monoidal category

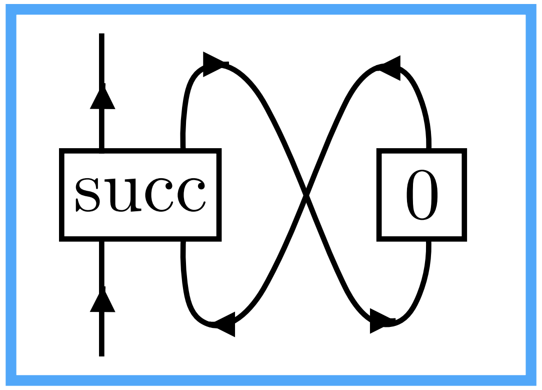
GoI is “memoryless”



- advantage: simplicity
- challenges
  - additive connectives $\&, \oplus$
  - computational effects

GoI is "memoryless"

- advantage: simplicity



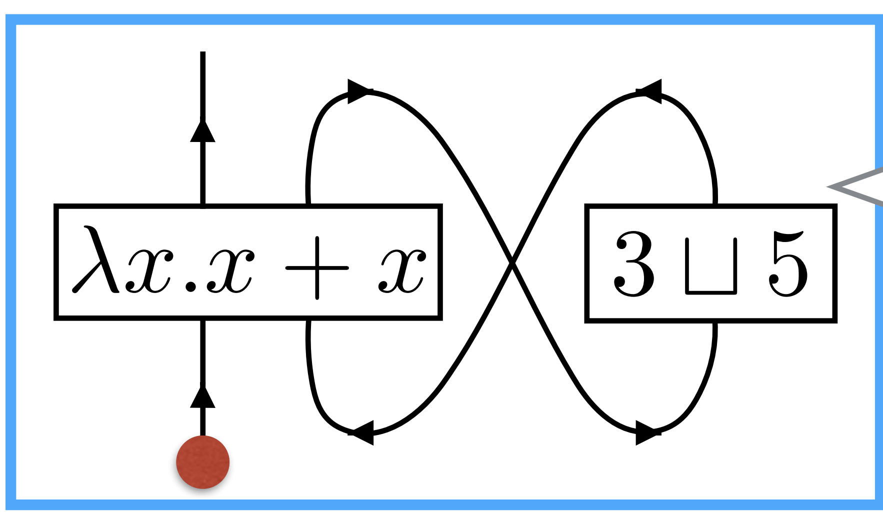
- challenges
  - additive connectives $\&, \oplus$
  - computational effects

additive slices
[Laurent '01]

GoI is "memoryless"

- challenge: computational effects

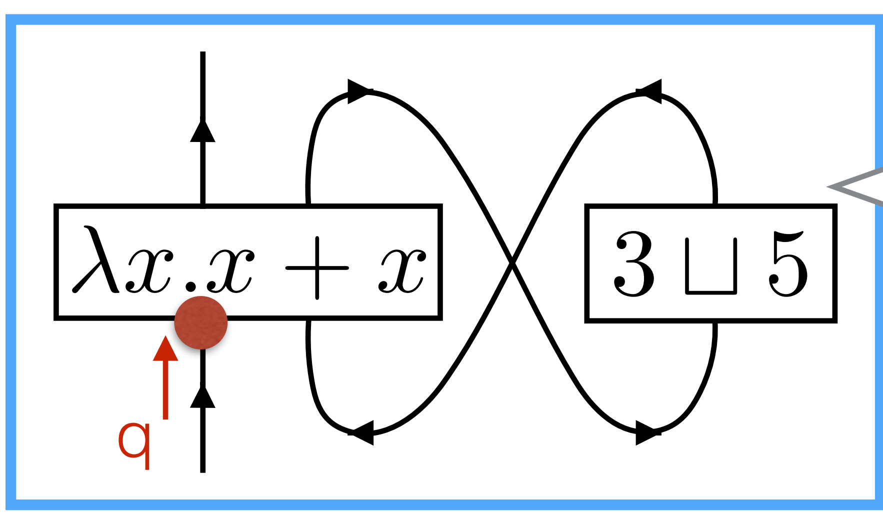
$(\lambda \mathtt{x} : \mathtt{nat}.\, \mathtt{x} + \mathtt{x})\,(3 \sqcup 5) \quad : \mathtt{nat}$



GoI is "memoryless"

- challenge: computational effects

$$(\lambda \mathtt{x} : \mathtt{nat}.\, \mathtt{x} + \mathtt{x})\,(3 \sqcup 5) \quad : \mathtt{nat}$$

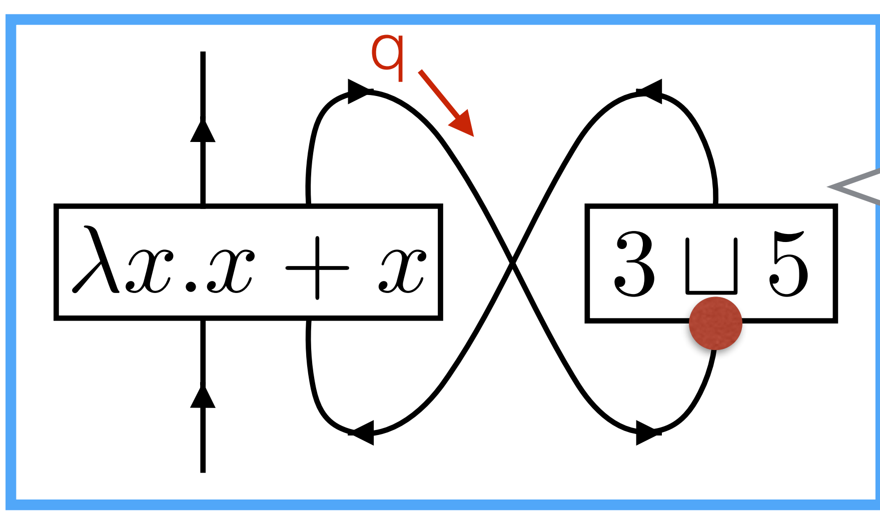


$\lambda x.x + x$	$3 \sqcup 5$

GoI is “memoryless”

- challenge: computational effects

$$(\lambda \mathtt{x} : \mathtt{nat}.\, \mathtt{x} + \mathtt{x})\,(3 \sqcup 5) \quad : \mathtt{nat}$$

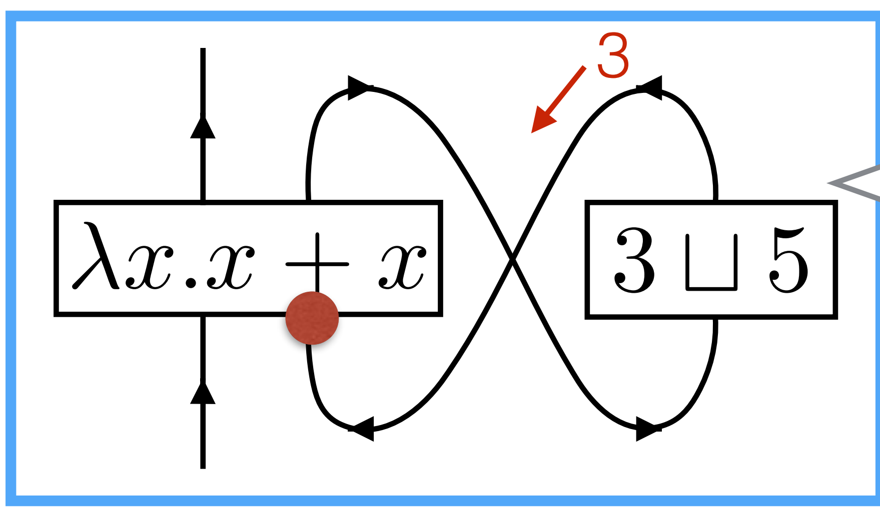


$\lambda x.x + x$	$3 \sqcup 5$
ask (left) x	

GoI is “memoryless”

- challenge: computational effects

$\boxed{(\lambda \mathtt{x} : \mathtt{nat}.\, \mathtt{x} + \mathtt{x})\,(3 \sqcup 5) \quad : \mathtt{nat}}$

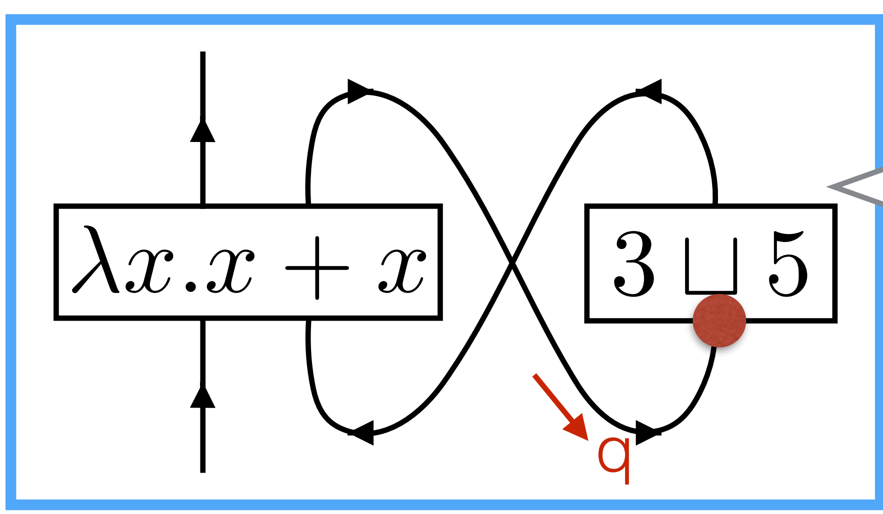


$\lambda x.x + x$	$3 \sqcup 5$
ask (left) x	
	answer 3

GoI is “memoryless”

- challenge: computational effects

$$(\lambda \mathtt{x} : \mathtt{nat}.\, \mathtt{x} + \mathtt{x})\,(3 \sqcup 5) \qquad : \mathtt{nat}$$

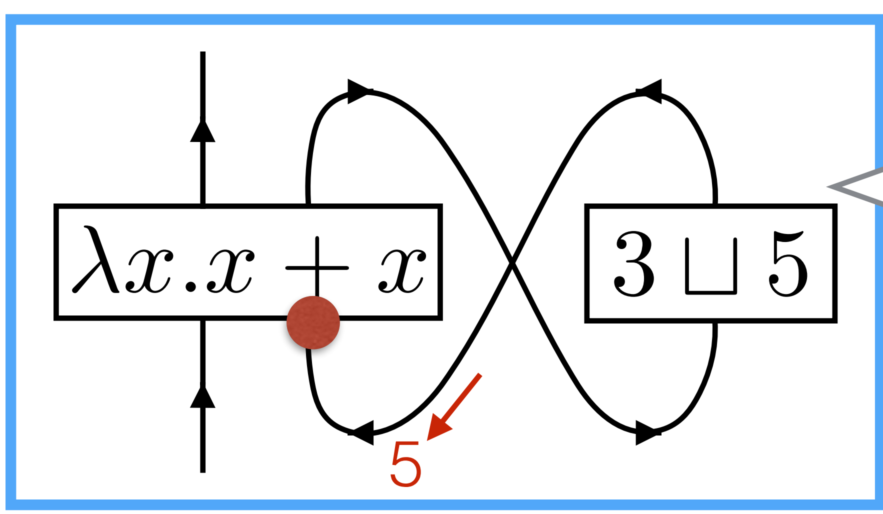


$\lambda x.x + x$	$3 \sqcup 5$
ask (left) x	
	answer 3
ask (right) x	

GoI is "memoryless"

- challenge: computational effects

$(\lambda \mathtt{x} : \mathtt{nat}.\, \mathtt{x} + \mathtt{x})\,(\mathtt{3} \sqcup \mathtt{5}) \quad : \mathtt{nat}$

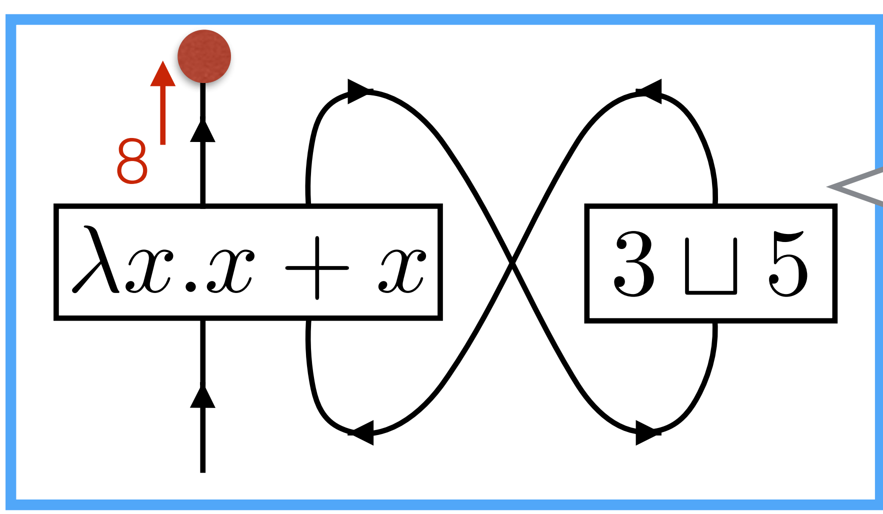


$\lambda x.x + x$	$3 \sqcup 5$
ask (left) x	
	answer 3
ask (right) x	
	answer 5

GoI is “memoryless”

- challenge: computational effects

$$(\lambda \mathtt{x} : \mathtt{nat}.\, \mathtt{x} + \mathtt{x})\,(3 \sqcup 5) \quad : \mathtt{nat}$$



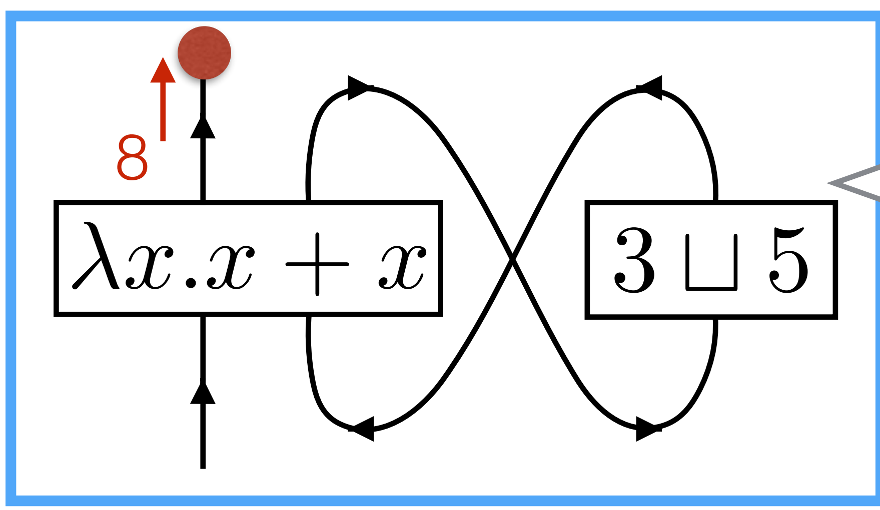
$\lambda x.x + x$	$3 \sqcup 5$
ask (left) x	
	answer 3
ask (right) x	
	answer 5
answer 8	

GoI is “memoryless”

memoryful GoI
[Hoshino, —, Hasuo CSL-LICS ’14]

- challenge: computational effects

$(\lambda \mathtt{x} : \mathtt{nat}.\, \mathtt{x} + \mathtt{x})\,(3 \sqcup 5) \quad : \mathtt{nat}$



$\lambda x.x + x$	$3 \sqcup 5$
ask (left) x	
	answer 3
ask (right) x	
	answer 5
answer 8	

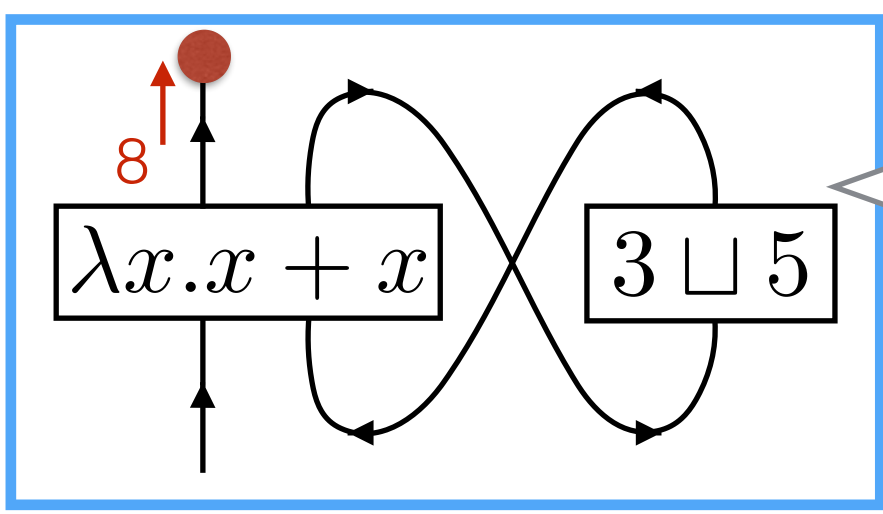
GoI is "memoryless"

memoryful GoI
[Hoshino, —, Hasuo CSL-LICS '14]

Thu., July 17
11:45 -

- challenge: computational effects

$$(\lambda \mathtt{x} : \mathtt{nat}.\, \mathtt{x} + \mathtt{x})\,(3 \sqcup 5) \quad : \mathtt{nat}$$



$\lambda x.x+x$	$3 \sqcup 5$
ask (left) x	
	answer 3
ask (right) x	
	answer 5
answer 8	

GoI is “memoryless”

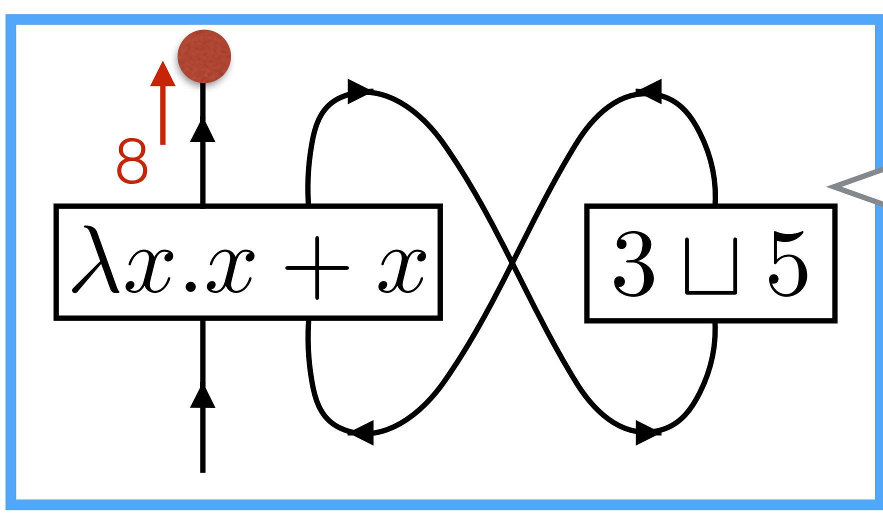
- challenge: computational effects

memoryful GoI
[Hoshino, —, Hasuo CSL-LICS ’14]

Thu., July 17
11:45 -

$(\lambda \mathtt{x} : \mathtt{nat}.\, \mathtt{x} + \mathtt{x})\,(3 \sqcup$

idea: equip each node with “memory”



$\lambda x.x + x$	$3 \sqcup 5$
ask (left) x	
	answer 3
ask (right) x	
	answer 5
answer 8	

Memoryful GoI — Input

terms

transducers

λ-terms with <u>algebraic effects</u>

algebraic operations [Plotkin, Power '03]

- nondeterministic choice
- probabilistic choice
- action on global state

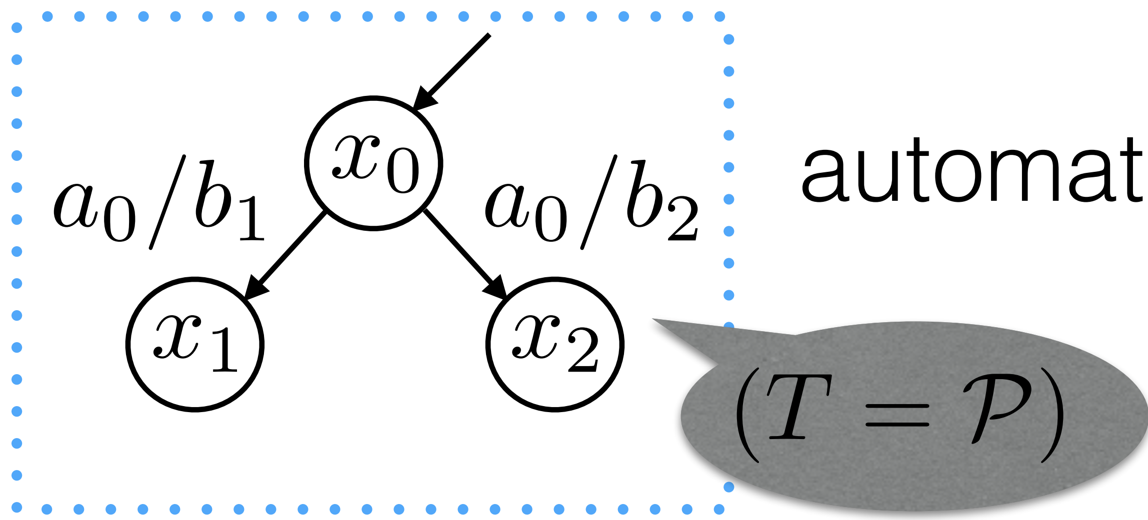
Memoryful GoI — Output

terms

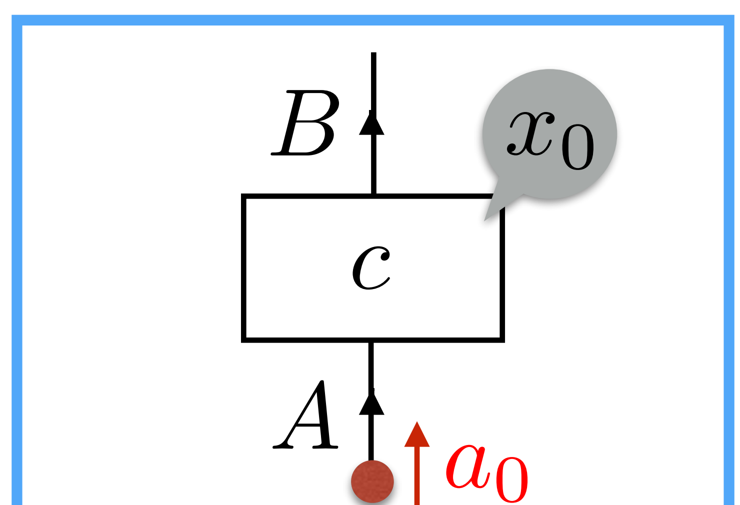
transducers

stream transducers (Mealy machines)

$$\mathcal{C} = (X, X \times A \xrightarrow{c} T(X \times B), x_0 \in X)$$



automaton style



string diagram style

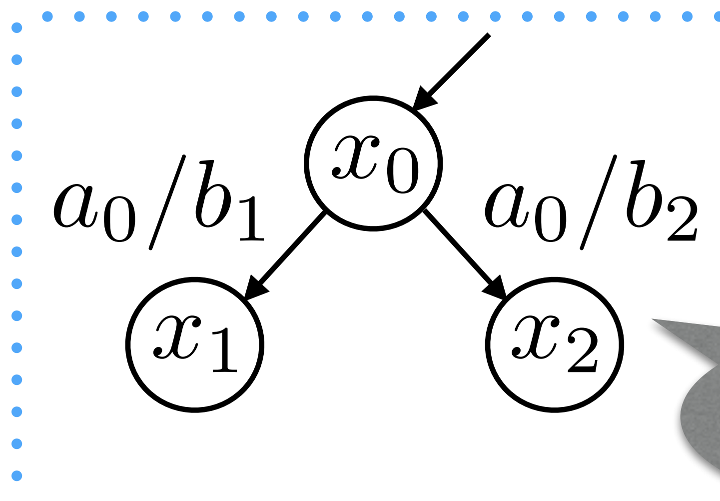
Memoryful GoI — Output

terms

transducers

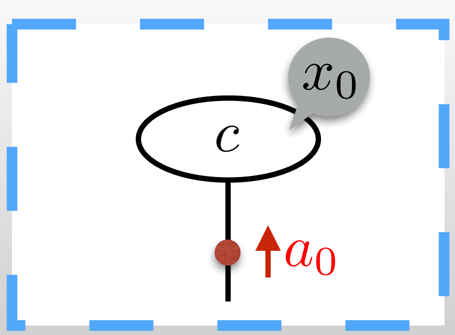
stream transducers (Mealy machines)

$$\mathcal{C} = (X, X \times A \xrightarrow{c} T(X \times B), x_0 \in X)$$

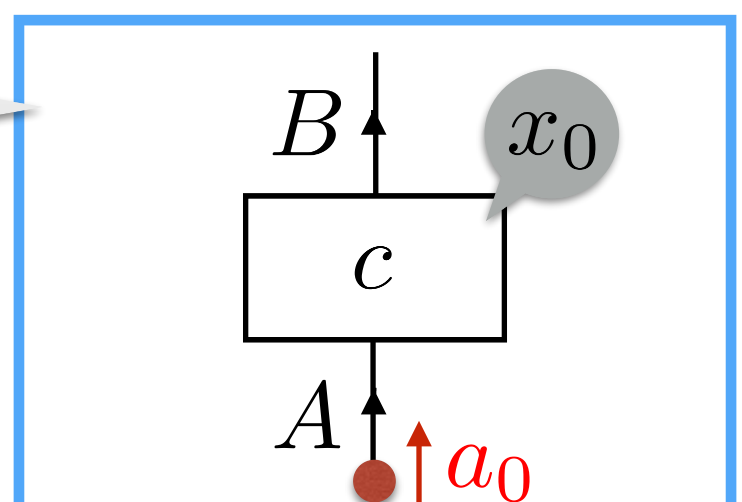


$(T = \mathcal{P})$

automaton style



proof net style



string diagram style

Memoryful GoI — Output

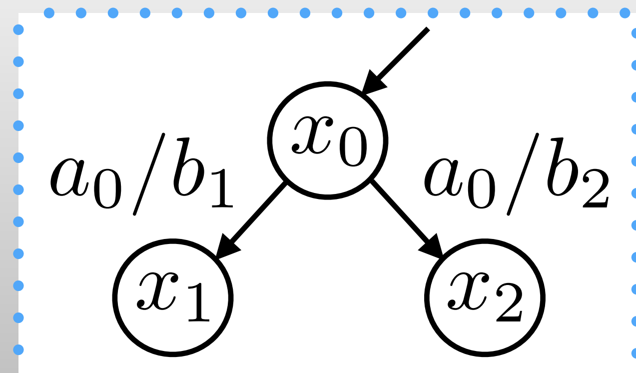
terms

transducers

stream transducers (Mealy machines)

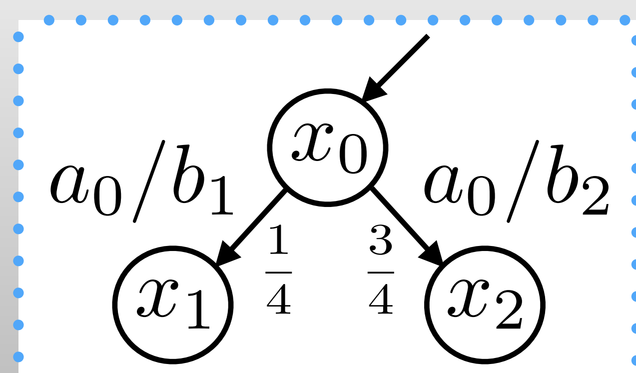
$$\mathcal{C} = (X,\ X \times A \xrightarrow{c} T(X \times B),\ x_0 \in X)$$

$T = \mathcal{P}$ $\quad (x_0, a_0) \longmapsto \{(x_1, b_1), (x_2, b_2)\}$



nondeterministic computation

$T = \mathcal{D}$ $\quad (x_0, a_0) \longmapsto \begin{bmatrix} (x_1, b_1) \mapsto 1/4, \\ (x_2, b_2) \mapsto 3/4 \end{bmatrix}$



probabilistic computation

Memoryful GoI — Translation

terms

↓

transducers

- idea: resumptions + categorical GoI

  [Abramsky, Haghverdi, Scott '02]

- use **coalgebraic component calculus**

  [Barbosa '03] [Hasuo, Jacobs '11]

  - **composition operations for software components**
  - **(many-sorted) process calculus**

Memoryful GoI — Translation

1. introduce component calculus over transducers

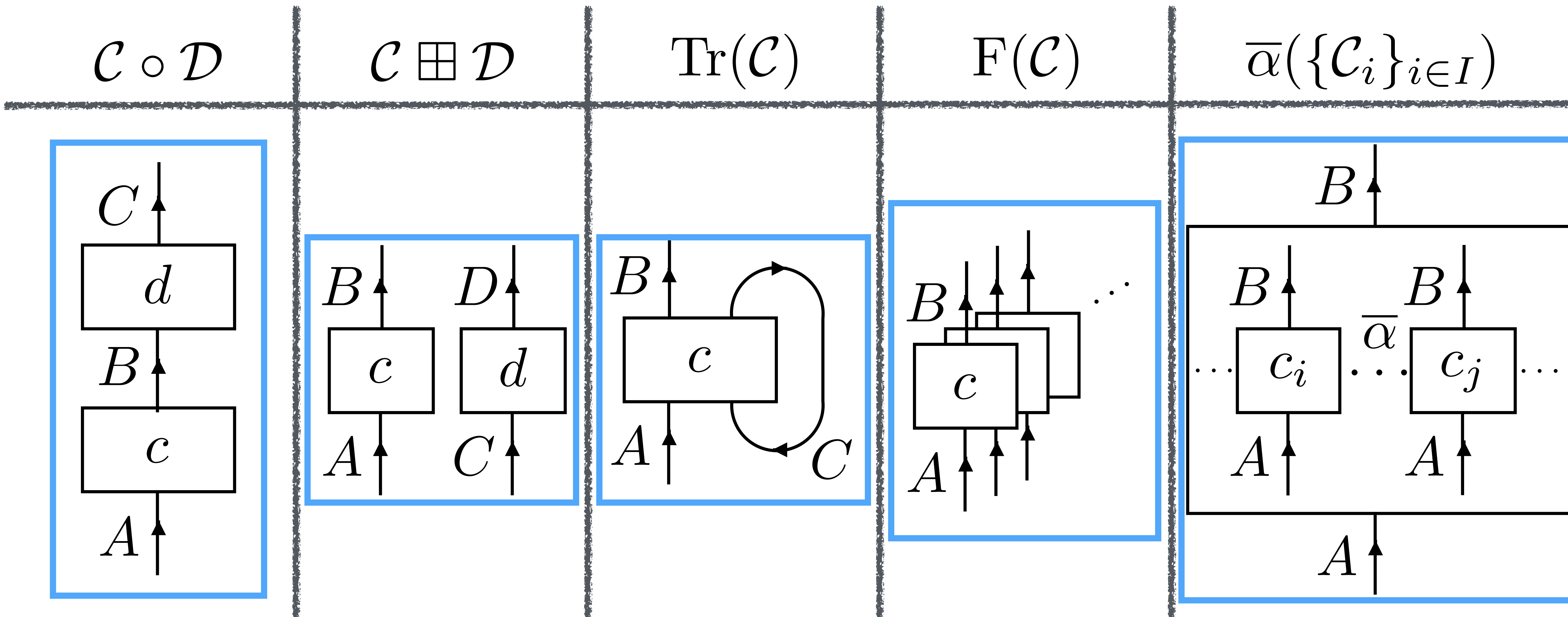
2. define interpretation inductively $(\!|\Gamma \vdash t\colon \tau|\!)$

$$(\!|\Gamma \vdash t\ \ s\colon \tau|\!)$$
$$= (\!|\Gamma \vdash t\colon \sigma \Rightarrow \tau|\!) \bullet (\!|\Gamma \vdash s\colon \sigma|\!)$$

3. prove soundness of interpretation $(\!|\Gamma \vdash t\colon \tau|\!)$

Memoryful GoI — Translation

Def. (component calculus)



(α: I-ary algebraic operation)

Memoryful GoI — Translation

Def. (component calculus)

sequential composition

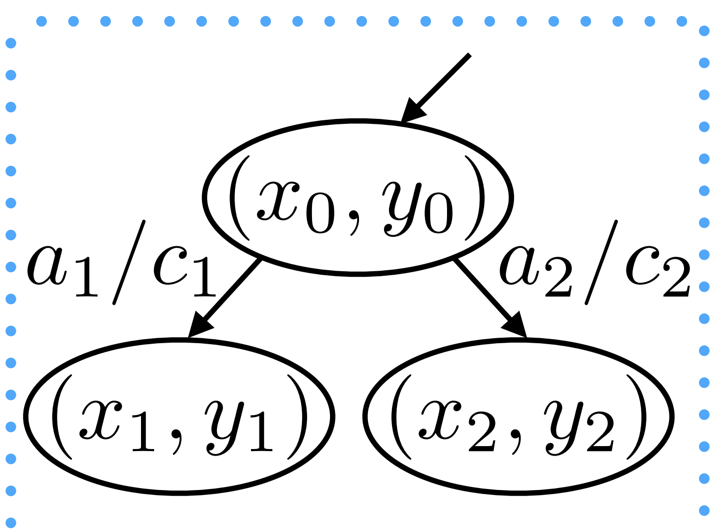
$\mathcal{C} \circ \mathcal{D}$

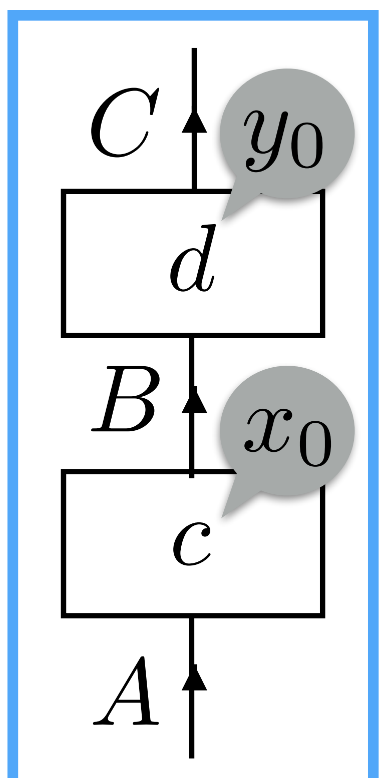
parallel composition

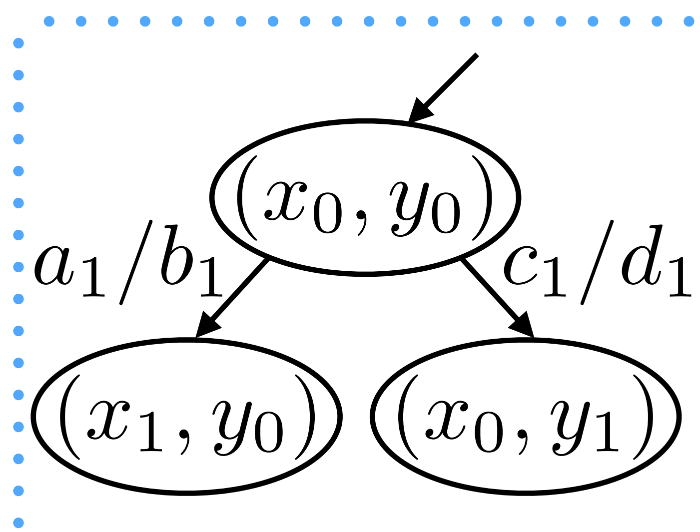
$\mathcal{C} \boxplus \mathcal{D}$

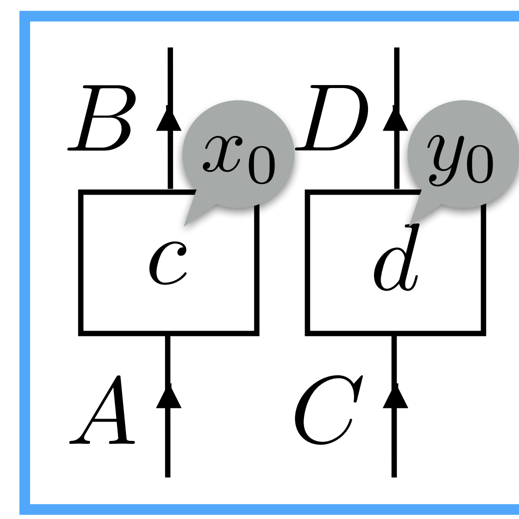
$$\begin{pmatrix} Y, \\ Y \times B \xrightarrow{d} T(Y \times C), \\ y_0 \in Y \end{pmatrix} \circ \begin{pmatrix} X, \\ X \times A \xrightarrow{c} T(X \times B), \\ x_0 \in X \end{pmatrix} = \begin{pmatrix} X \times Y, \\ \cdots \\ (x_0, y_0) \in X \times Y \end{pmatrix}$$

$$\begin{pmatrix} X, \\ X \times A \xrightarrow{c} T(X \times B), \\ x_0 \in X \end{pmatrix} \boxplus \begin{pmatrix} Y, \\ Y \times C \xrightarrow{d} T(Y \times D), \\ y_0 \in Y \end{pmatrix} = \begin{pmatrix} X \times Y, \\ \cdots \\ (x_0, y_0) \in X \times Y \end{pmatrix}$$









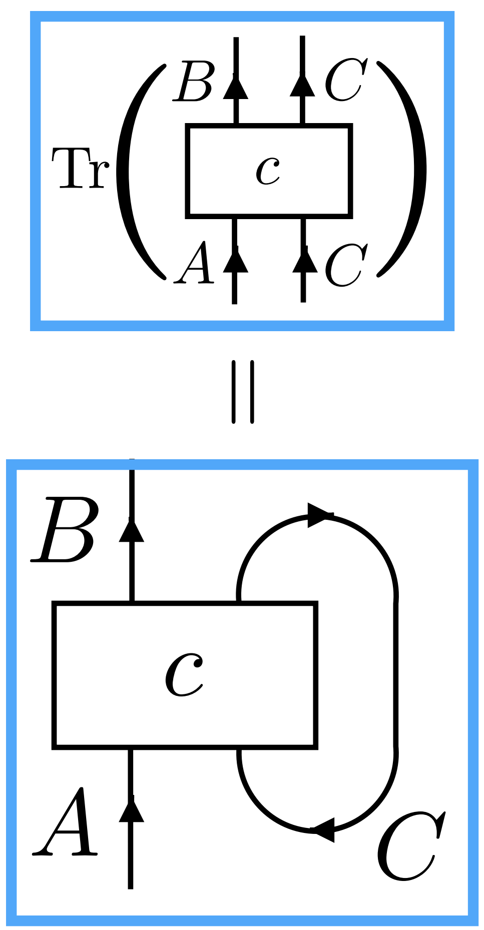
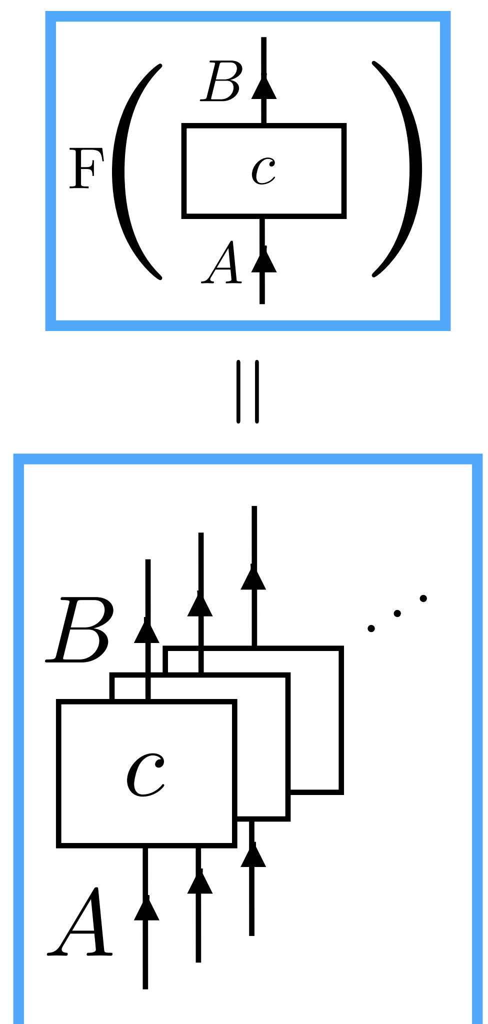
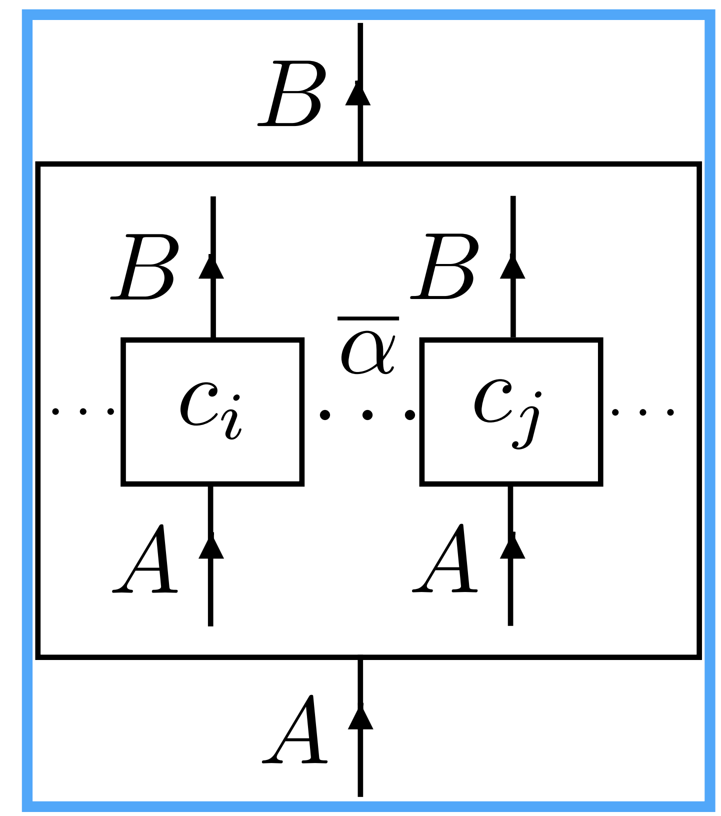
Memoryful GoI — Translation

Def. (component calculus)

application

!-modality

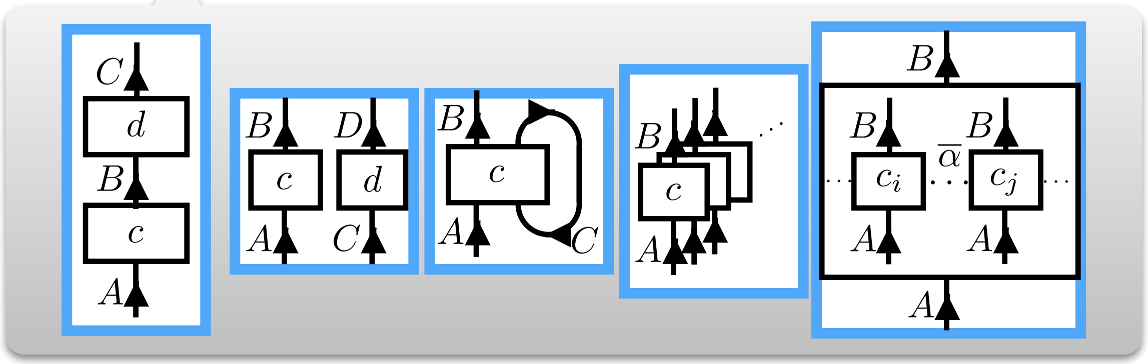
algebraic effect

$\mathrm{Tr}(\mathcal{C})$	$\mathrm{F}(\mathcal{C})$	$\overline{\alpha}(\{\mathcal{C}_i\}_{i\in I})$
		

(α: I-ary algebraic operation)

Memoryful GoI — Translation

1. introduce component calculus over transducers



2. define interpretation inductively $(\!|\Gamma \vdash t \colon \tau|\!)$

$$(\!|\Gamma \vdash t\ \ s \colon \tau|\!) = (\!|\Gamma \vdash t \colon \sigma \Rightarrow \tau|\!) \bullet (\!|\Gamma \vdash s \colon \sigma|\!)$$

3. prove soundness of interpretation $(\!|\Gamma \vdash t \colon \tau|\!)$

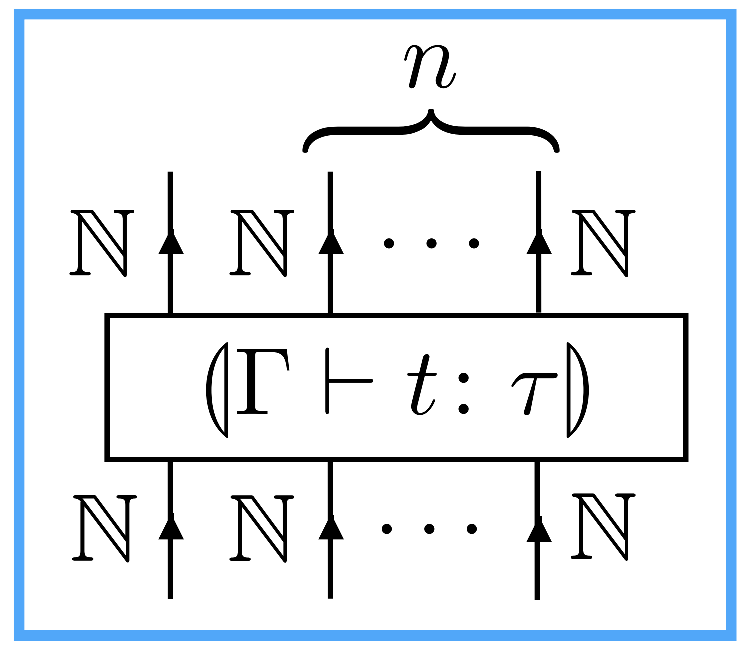
Memoryful GoI — Translation

Def. (interpretation $(\!|\Gamma \vdash t \colon \tau|\!)$)

For a type judgement $\Gamma \vdash t \colon \tau \;\; (\Gamma = x_1 \colon \tau_1, \ldots, x_n \colon \tau_n)$,

we inductively define

$$(\!|\Gamma \vdash t \colon \tau|\!) =$$

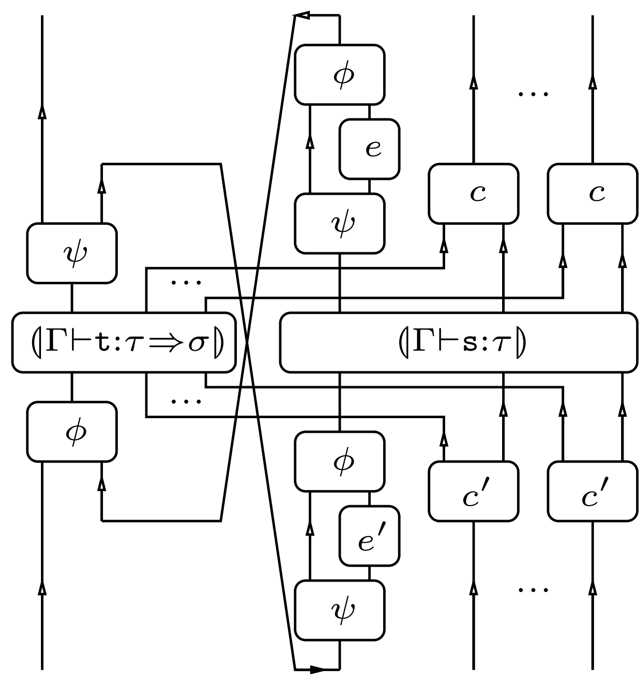


.

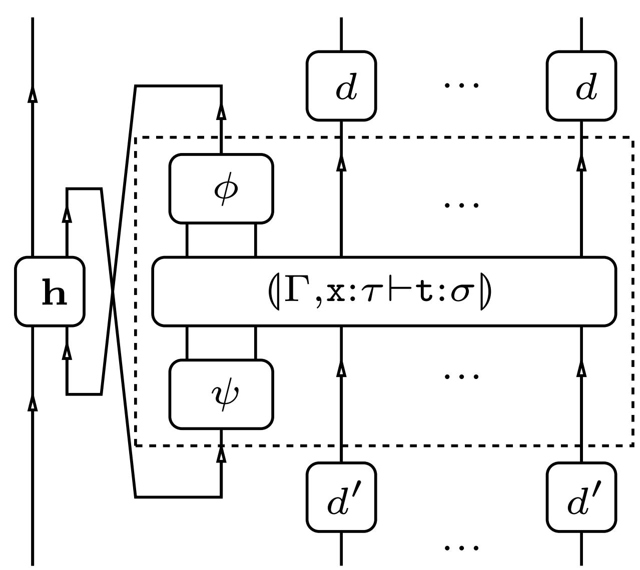
Memoryful GoI — Translation

Def. (interpretation $(\!|\Gamma \vdash t : \tau|\!)$)

$(\!|\Gamma \vdash \mathtt{t}\ \mathtt{s} : \sigma|\!) =$



$(\!|\Gamma \vdash \lambda\mathtt{x} : \tau.\, \mathtt{t} : \tau \Rightarrow \sigma|\!) =$



$$\frac{A \Rightarrow B}{!A \multimap B}$$

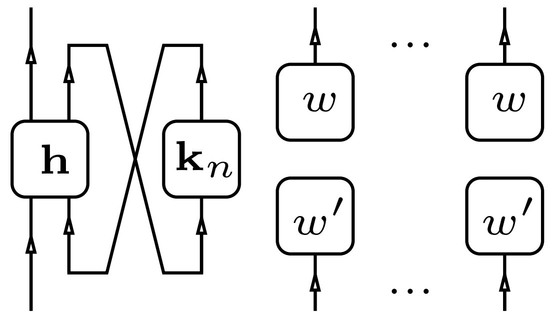
intuitionistic logic

linear logic

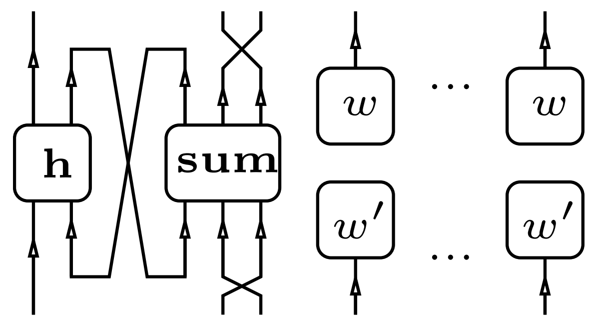
Memoryful GoI — Translation

<u>Def. (interpretation $(\!|\Gamma \vdash t : \tau|\!)$)</u>

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

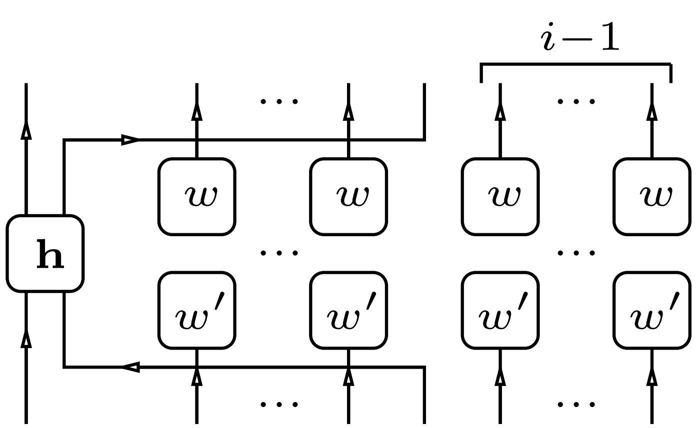


$$(\!|\Gamma, \mathtt{x} : \mathbf{nat}, \mathtt{y} : \mathbf{nat} \vdash \mathtt{x} + \mathtt{y} : \mathbf{nat}|\!) =$$



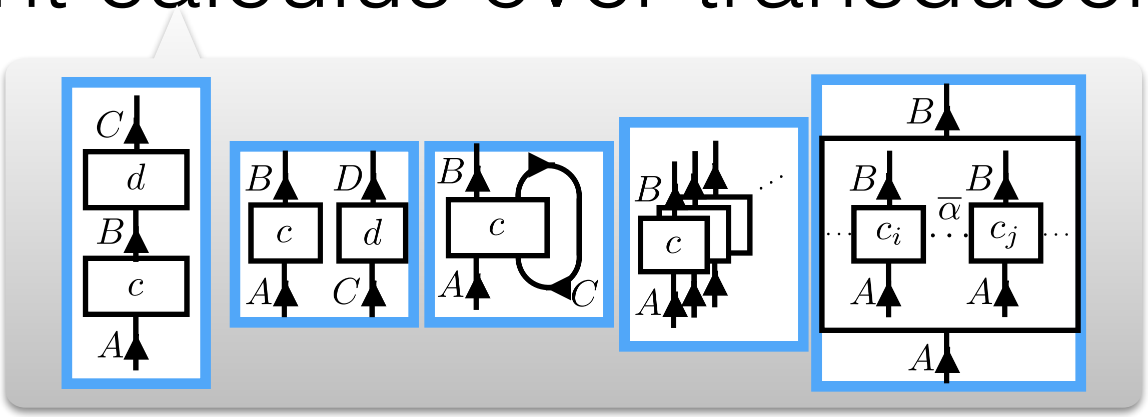
$$(\!|\Gamma \vdash \mathtt{t} + \mathtt{s} : \mathbf{nat}|\!) = (\!|\Gamma \vdash (\lambda \mathtt{xy} : \mathbf{nat}.\, \mathtt{x} + \mathtt{y})\, \mathtt{t}\, \mathtt{s} : \mathbf{nat}|\!)$$

$$(\!|\mathtt{x}_1 : \tau_1, \cdots, \mathtt{x}_n : \tau_n \vdash \mathtt{x}_i : \tau_i|\!) =$$

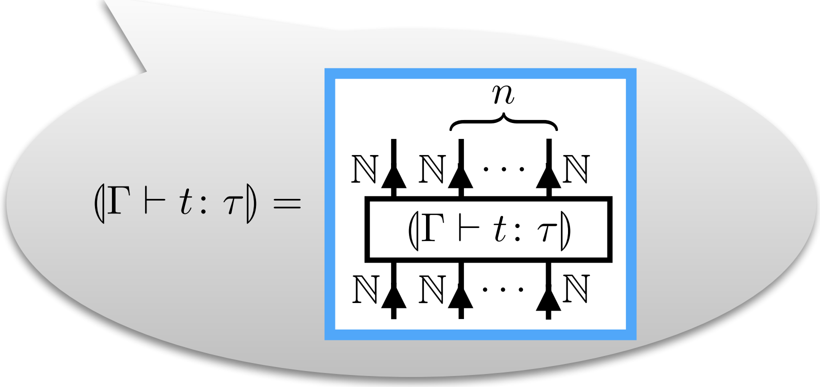


Memoryful GoI — Translation

1. introduce component calculus over transducers



2. define interpretation inductively $(\!|\Gamma \vdash t\colon \tau|\!)$



3. prove soundness of interpretation $(\!|\Gamma \vdash t\colon \tau|\!)$

Memoryful GoI — Translation

Thm. (soundness)

Theorem 6.2 (Soundness). *For closed terms* $\mathtt{t}$ *and* $\mathtt{s}$ *of type* τ*,*

- *If* $\mathtt{t} \approx \mathtt{s}$*, then* $([\llparenthesis \mathtt{t} \rrparenthesis], [\llparenthesis \mathtt{s} \rrparenthesis]) \in \Phi[\![\tau]\!]$.
- *If* $\mathtt{t} \approx \mathtt{s}$ *and* τ *is the base type* $\mathtt{nat}$*, then* $\llparenthesis \mathtt{t} \rrparenthesis \simeq^{T}_{\mathbb{N},\mathbb{N}} \llparenthesis \mathtt{s} \rrparenthesis$.

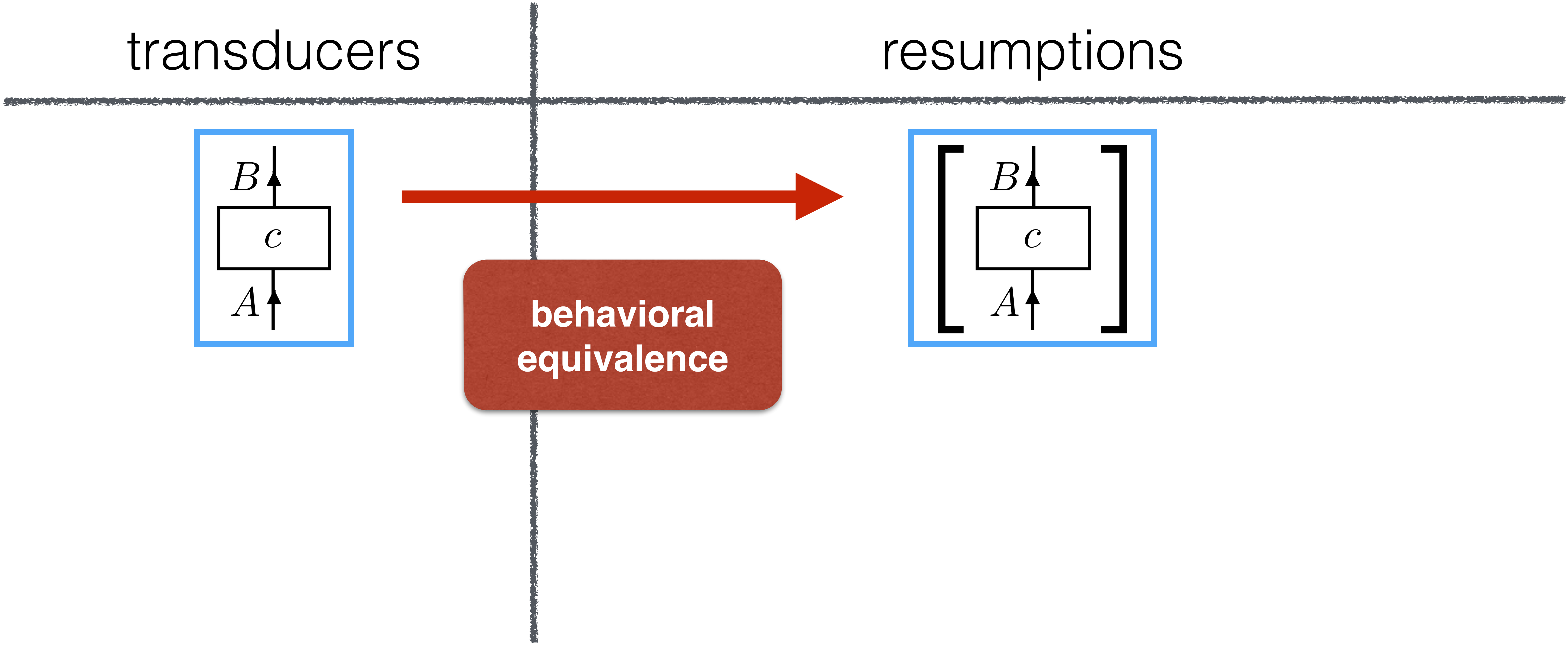
where $[\llparenthesis \mathtt{t} \rrparenthesis]$ *is the* $\mathbf{Res}(T)$*-morphism represented by* $\llparenthesis \mathtt{t} \rrparenthesis$*, and we write* $\mathtt{t} \approx \mathtt{s}$ *when the equation holds in the extension of the computational lambda calculus. For example, we have*

$$\mathtt{v}\,(3 \sqcup 5) \approx \mathtt{v}\,3 \sqcup \mathtt{v}\,5, \qquad 3 \sqcup 5 \sqcup 3 \approx 3 \sqcup 5 \approx 5 \sqcup 3$$

for any value $\mathtt{v}$ *when the extension of the computational lambda calculus has nondeterminism.*

Memoryful GoI — Translation

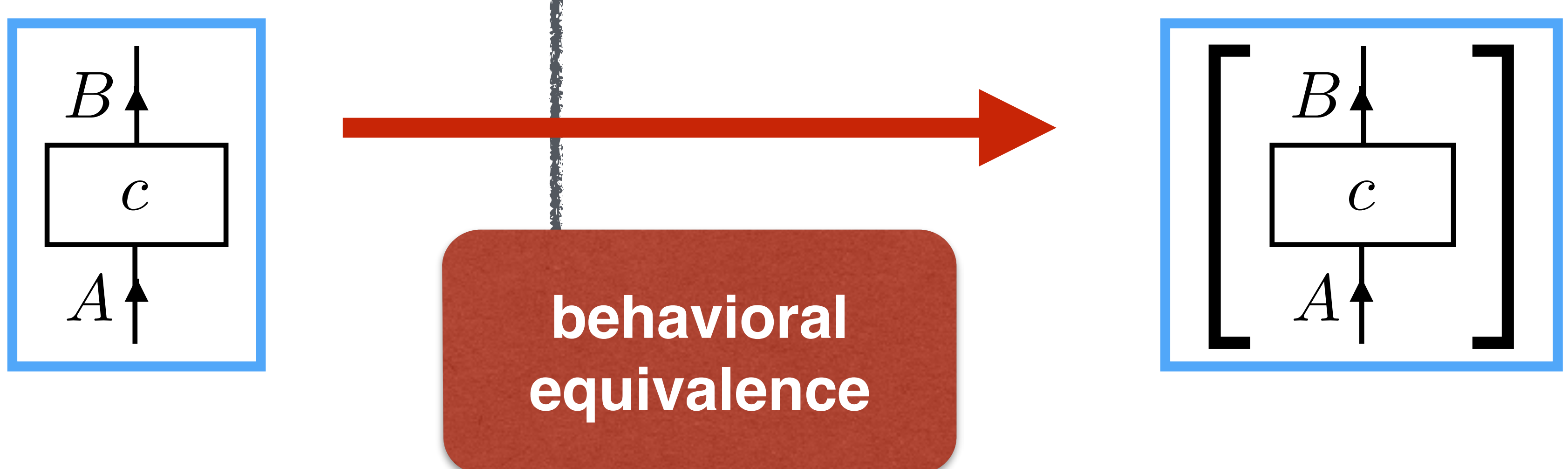
proof (soundness)



Memoryful GoI — Translation

<u>proof (soundness)</u>

transducers | resumptions

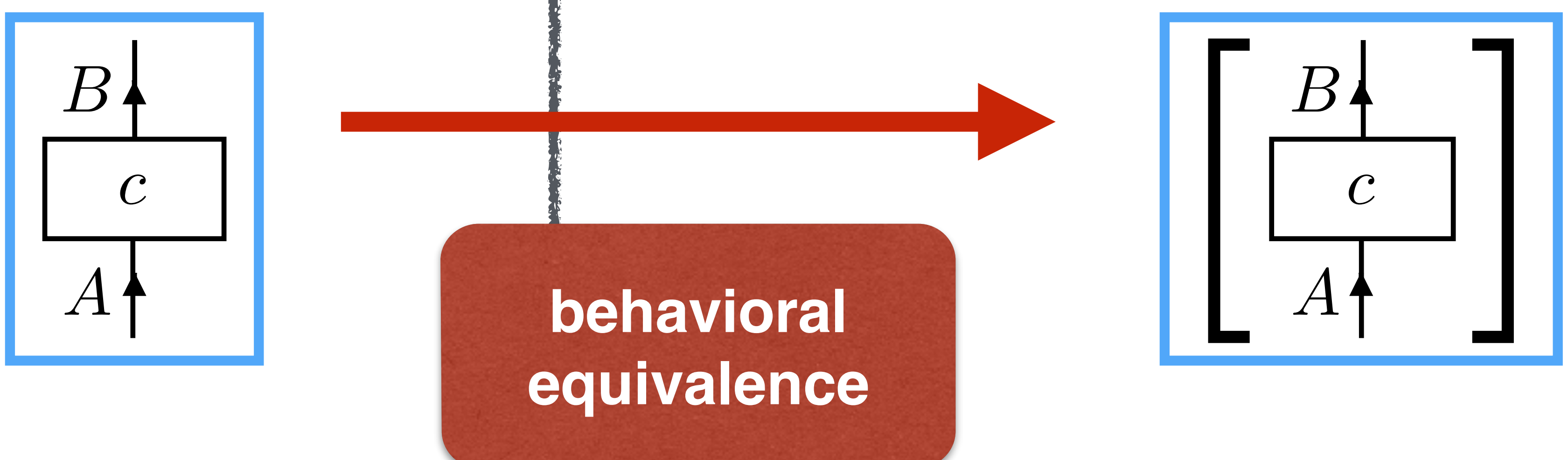


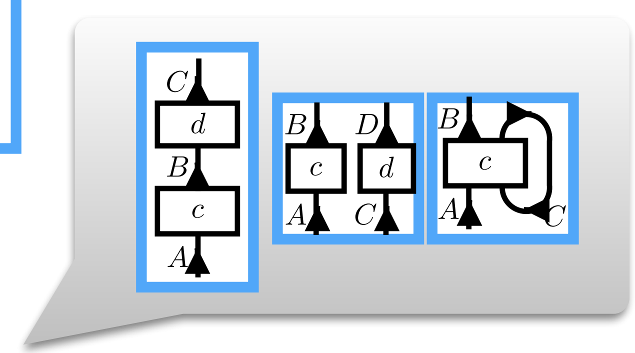
GoI situation $\left(\begin{array}{c}(\mathbf{Res}(T), \emptyset, \boxplus, \mathrm{Tr}), \\ F, J_0\phi, J_0\psi, J_0u, J_0v\end{array}\right)$

Memoryful GoI — Translation

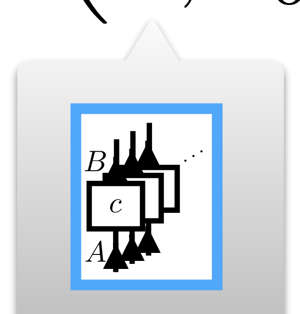
<u>proof (soundness)</u>

transducers	resumptions





GoI situation $\left(\begin{array}{c}(\mathbf{Res}(T), \emptyset, \boxplus, \mathrm{Tr}), \\ F, J_0\phi, J_0\psi, J_0u, J_0v\end{array}\right)$



$\phi : \mathbb{N} + \mathbb{N} \cong \mathbb{N} : \psi$

$u : \mathbb{N} \times \mathbb{N} \cong \mathbb{N} : v$

Memoryful GoI — Translation

proof (soundness)

resumptions	partial equivalence relations (per's) on resumptions
GoI situation $\begin{pmatrix} (\mathbf{Res}(T), \emptyset, \boxplus, \mathrm{Tr}), \\ F, J_0\phi, J_0\psi, J_0u, J_0v \end{pmatrix}$ →	cartesian closed category $\mathbf{Per}(T)$

categorical GoI [Abramsky, Haghverdi, Scott '02]
realizability

Memoryful GoI — Translation

proof (soundness)

resumptions	partial equivalence relations (per's) on resumptions
GoI situation $\left(\begin{array}{c}(\mathbf{Res}(T), \emptyset, \boxplus, \mathrm{Tr}), \\ F, J_0\phi, J_0\psi, J_0 u, J_0 v\end{array}\right)$ →	cartesian closed category $\mathbf{Per}(T)$ monad Φ on $\mathbf{Per}(T)$

categorical GoI [Abramsky, Haghverdi, Scott '02]
realizability

Memoryful GoI — Translation

<u>proof (soundness)</u>

transducers	resumptions	partial equivalence relations (per's) on resumptions
		cartesian closed category $\mathbf{Per}(T)$
		monad Φ on $\mathbf{Per}(T)$
		$\downarrow$ denotational semantics
		$[\![\vdash t\colon \tau]\!] =$ equivalence class of $\Phi[\![\tau]\!] \in \mathbf{Per}(T)$

Memoryful GoI — Translation

<u>proof (soundness)</u>

transducers	resumptions	partial equivalence relations (per's) on resumptions
		cartesian closed category $\mathbf{Per}(T)$
		monad Φ on $\mathbf{Per}(T)$
		denotational semantics
$(\!\vdash t \colon \tau\!)$		$[\![\vdash t \colon \tau]\!] =$ equivalence class of $\Phi[\![\tau]\!] \in \mathbf{Per}(T)$

Memoryful GoI — Translation

proof (soundness)

transducers	resumptions	partial equivalence relations (per's) on resumptions
		cartesian closed category $\mathbf{Per}(T)$
		monad Φ on $\mathbf{Per}(T)$
		denotational semantics
$(\!\vert \vdash t : \tau \vert\!)$ →		$[\![\vdash t : \tau]\!] =$ equivalence class of $\Phi[\![\tau]\!] \in \mathbf{Per}(T)$

Memoryful GoI — Translation

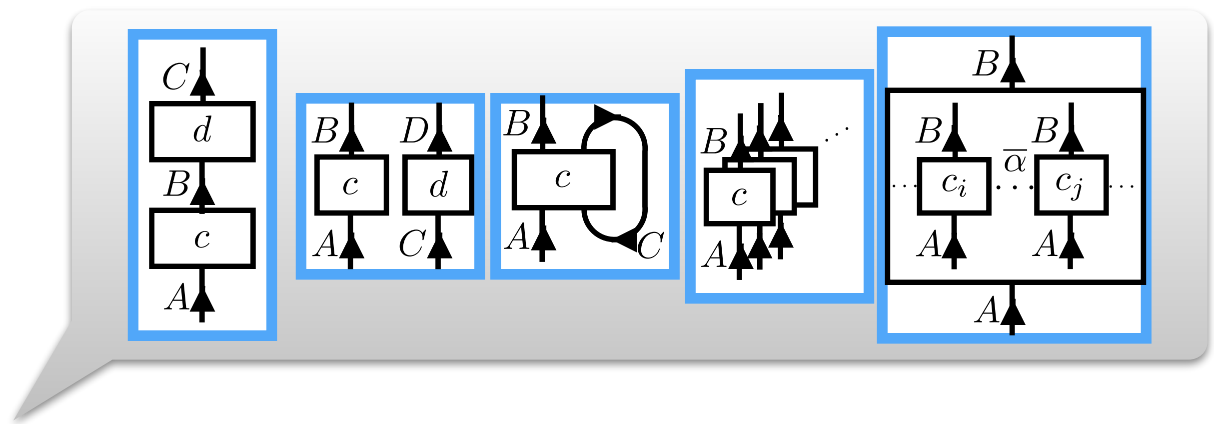
<u>proof (soundness)</u>

transducers	resumptions	partial equivalence relations (per's) on resumptions
		cartesian closed category $\mathbf{Per}(T)$
		monad Φ on $\mathbf{Per}(T)$
		denotational semantics
$(\!\vert \vdash t \colon \tau \vert\!)$		$[\![\vdash t \colon \tau]\!] =$ equivalence class of $\Phi[\![\tau]\!] \in \mathbf{Per}(T)$

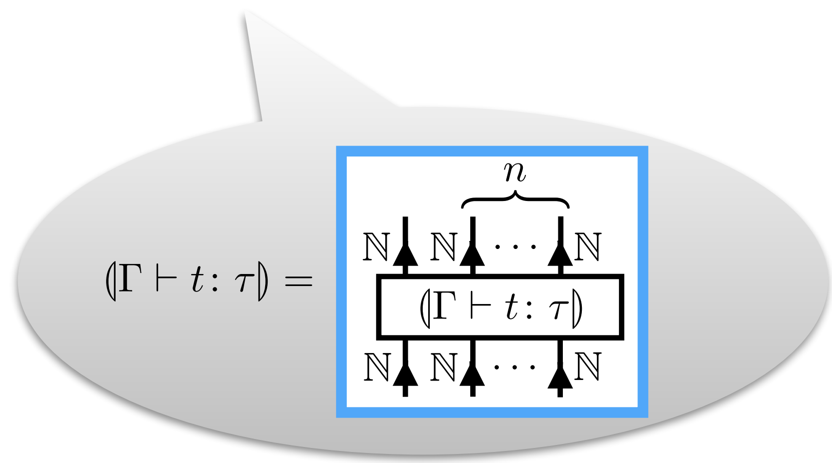
Memoryful GoI — Summary

terms

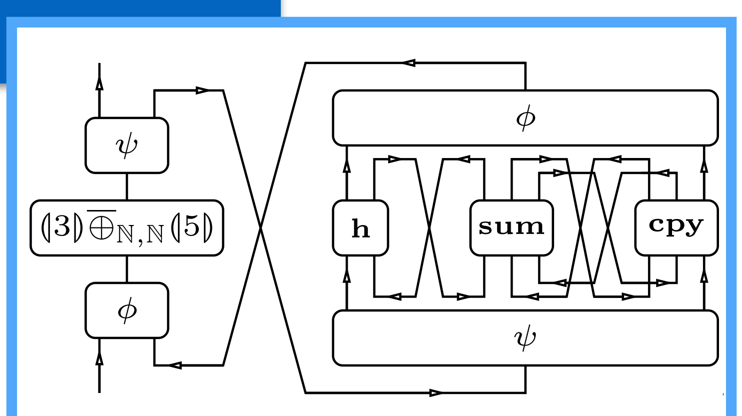
$(\lambda \mathtt{x} : \mathtt{nat}.\, \mathtt{x} + \mathtt{x})\,(3 \sqcup 5) \quad : \mathtt{nat}$



use **coalgebraic component calculus**



transducers



Our Tool *TtT*

“Terms to Transducers”

terms

memoryful GoI

transducers

λ-terms with effects

TtT Compiler

Haskell program

```
type  Td m x a b  =  (x, a) -> m (x, b)
```

TtT Simulator

simulation result

Our Tool *TtT* — Demonstration

$\boxed{3 \sqcup 5}$

```
threeOrFive =
  Oplus (Const 3) (Const 5)
```

$\boxed{(\lambda x.\, x)\; 1}$

```
idOne = Apply
  (Abst "x" $ Variable "x")
  (Const 1)
```

$\boxed{(\lambda f.\, f\, 0 + f\, 1)\; (\lambda x.\, 3 \sqcup 5)}$

```
secondNondetExample = Apply
  (Abst "f" $ sumLambda
    (Apply (Variable "f") (Const 0))
    (Apply (Variable "f") (Const 1))
  )
  (Abst "x" $ Oplus (Const 3) (Const 5))
```

Our Tool *TtT* — Demonstration

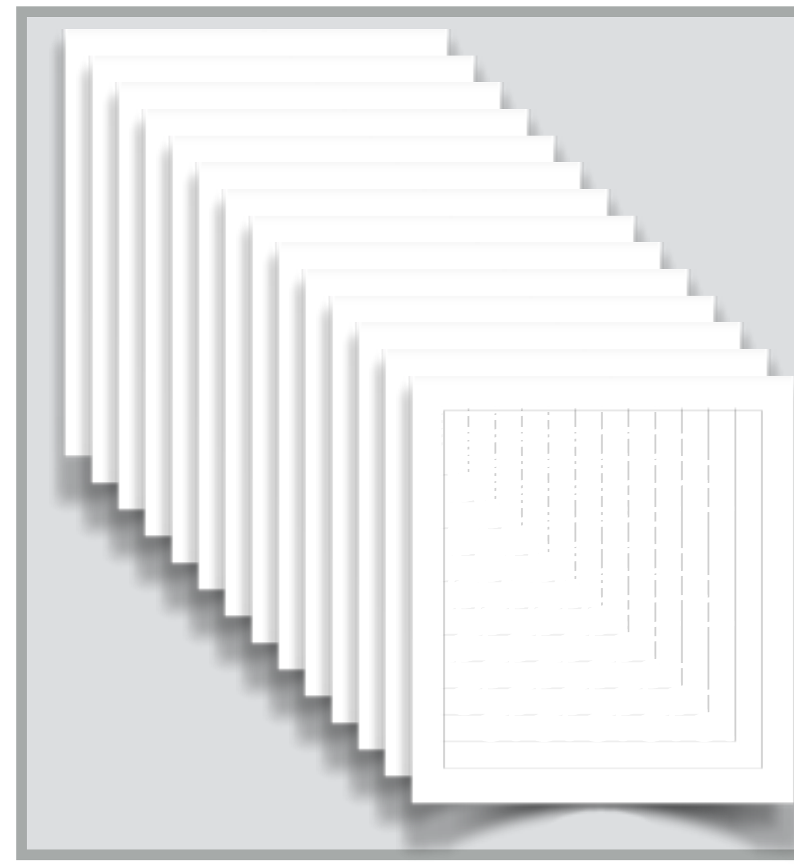
$3 \sqcup 5$

```
.    ---- dd<42,137> --->
|    Query:[ [3|_|5] @ Nothing ]
+-.    ---- dd<42,137> --->
| |    Query:[ [3]|_|5 @ * ]
| |      h; k_3; h
| |    [ [3]|_|5 @ * ]:Answer
| |    ---- dd<42,3> --->
| |  [ [3|_|5] @ Just (Left (*)) ]:Answer
| |  ---- dd<42,3> --->
| Result: 3 / State: Just (Left (*))
`-.    ---- dd<42,137> --->
  |    Query:[ 3|_|[5] @ * ]
  |      h; k_5; h
  |    [ 3|_|[5] @ * ]:Answer
  |    ---- dd<42,5> --->
  |  [ [3|_|5] @ Just (Right (*)) ]:Answer
  |  ---- dd<42,5> --->
  Result: 5 / State: Just (Right (*))
```

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

```
---- dd<42,137> --->
Query:[ [(\x.x) 1] @ ({_: *}, *) ]
  phi
  ---- gdd<42,137> --->
  Query:[ [\x.x] 1 @ {_: *} ]
    h
  [ [\x.x] 1 @ {_: *} ]:Answer
  ---- dgdd<42,137> --->
  psi; psi; phi
  ---- gdd<42,137> --->
  Query:[ (\x.x) [1] @ * ]
    h
  [ (\x.x) [1] @ * ]:Answer
  ---- dgdd<42,137> --->
  psi; e; phi; phi
  ---- dd<0,gdd<42,137>> --->
  Query:[ [\x.x] 1 @ {_: *} ]
    h; v
    0 {
      psi
      ---- dd<42,137> --->
      Query:[ (\x.[x]) 1 @ * ]
        h
      [ (\x.[x]) 1 @ * ]:Query x
      ---- <42,137> --->
      phi
    } 0
    u; h
  [ [\x.x] 1 @ {_: *} ]:Answer
  ---- dd<0,d<42,137>> --->
  psi; psi; e'; phi
  ---- dd<42,137> --->
  Query:[ (\x.x) [1] @ * ]
    h; k_1; h
  [ (\x.x) [1] @ * ]:Answer
  ---- dd<42,1> --->
  psi; e; phi; phi
  ---- dd<0,d<42,1>> --->
  Query:[ [\x.x] 1 @ {_: *} ]
    h; v
    0 {
      psi
      ---- <42,1> --->
      Answer x:[ (\x.[x]) 1 @ * ]
        h
      [ (\x.[x]) 1 @ * ]:Answer
      ---- dd<42,1> --->
      phi
    } 0
    u; h
  [ [\x.x] 1 @ {_: *} ]:Answer
  ---- dd<0,gdd<42,1>> --->
  psi; psi; e'; phi
  ---- dgdd<42,1> --->
  Query:[ (\x.x) [1] @ * ]
    h
  [ (\x.x) [1] @ * ]:Answer
  ---- gdd<42,1> --->
  psi; phi; phi
  ---- dgdd<42,1> --->
  Query:[ [\x.x] 1 @ {_: *} ]
    h
  [ [\x.x] 1 @ {_: *} ]:Answer
  ---- gdd<42,1> --->
  psi
[ [(\x.x) 1] @ ({_: *}, *) ]:Answer
---- dd<42,1> --->
Result: 1 / State: ({_: *}, *)
```

$(\lambda f.\, f\ 0 + f\ 1)\ (\lambda x.\, 3 \sqcup 5)$



(4,526 lines)

Our Tool *TtT*

Our Tool *TtT*

- currently no practical use

Our Tool *TtT*

- currently no practical use
- nevertheless worthwhile
  - helpful for studying higher-order effectful computations
    - showing dynamics of token
  - (speculative) basis of compiler for effectful computations
    - following [Mackie '95] [Pinto '01] [Ghica '07]

Our Tool *TtT*

- currently no practical use
- nevertheless worthwhile
  - helpful for studying higher-order effectful computations
    - showing dynamics of token
  - (speculative) basis of compiler for effectful computations
    - following [Mackie '95] [Pinto '01] [Ghica '07]
- fun to see GoI at work!