Abstract interpretation of concurrent, higher-order programs

Matt Might
University of Utah
matt.might.net
@mattmight

David Van Horn
Northeastern University
lambda-calcul.us
@lambda_calculus

Goal: MHP & CFA for concurrent, higher-order programs

tl;dr

tl;dr

- Concrete semantics: CESK => P(CEK*)S

tl;dr

- Concrete semantics: CESK => P(CEK*)S
- MHP = abstract semantics + thread shape

tl;dr

- Concrete semantics: CESK => P(CEK*)S
- MHP = abstract semantics + thread shape
- CFA = re-abstraction of MHP semantics

Theme: Semantic design

More in paper

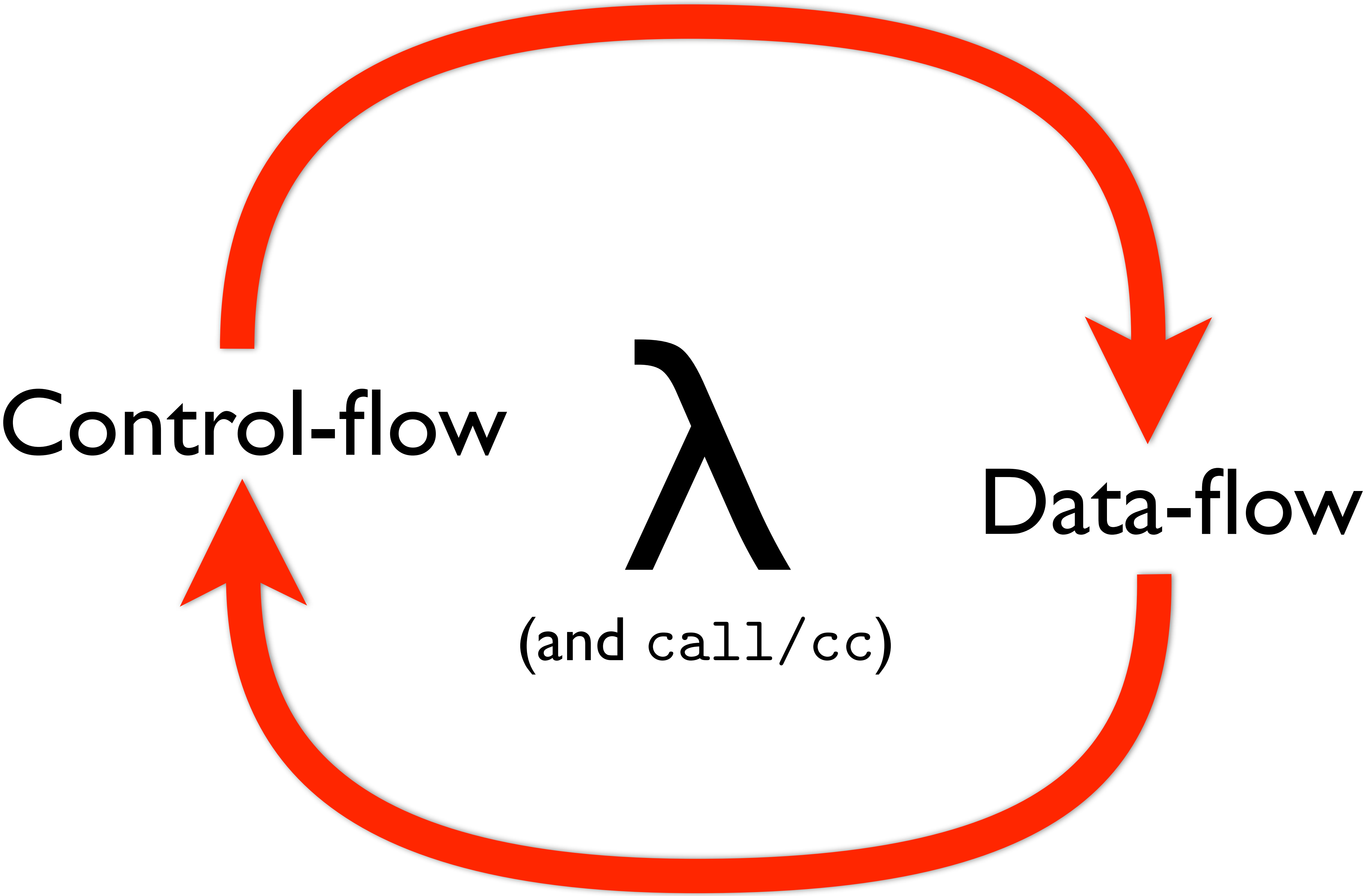
- A desugaring of `future`
- Thread polyvariance
- More formal semantics
- More on parallel `call/cc`

Challenges

Higher-order is hard.

λ

(and `call/cc`)


Control-flow
λ
(and call/cc)
Data-flow

But, we can do it.

(Jones, 1981)
(Shivers, 1988)

Parallelism is hard.

Thread 1

```
a := 3
b := x + y
c := z * x
```

Thread 2

```
x := 10
y := a + b
z := c + x
```

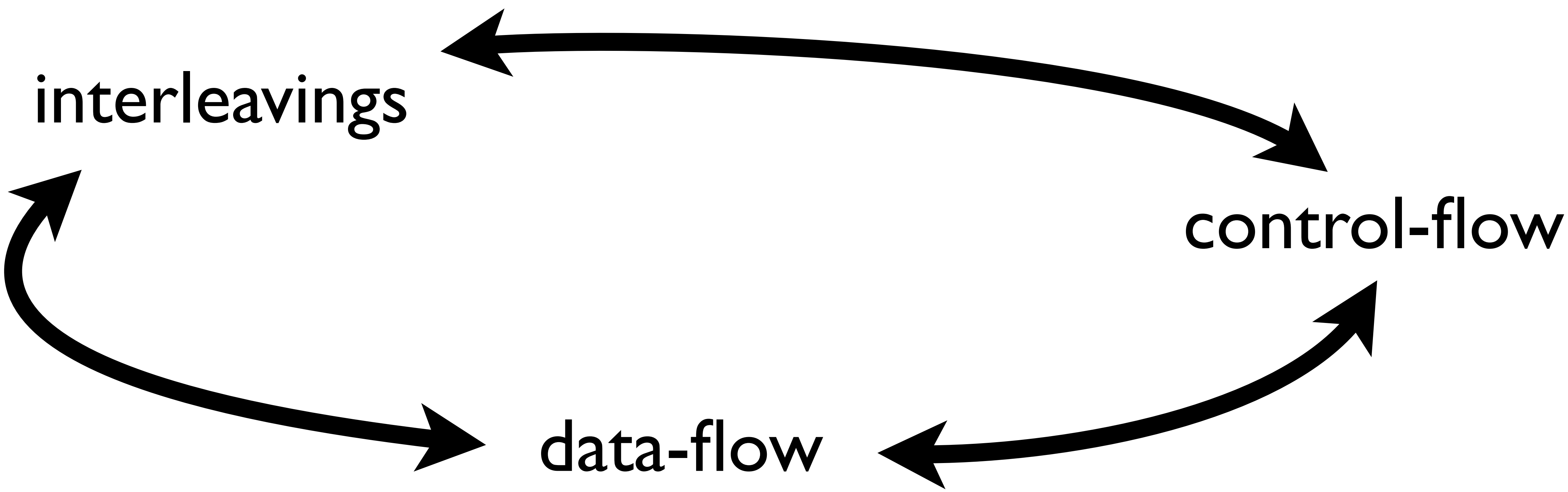
Thread 1 | Thread 2

```
a := 3
x := 10
b := x + y
y := a + b
c := z * x
z := c + x
```

But, we can do that too.

(Yahav, 2001)

Parallel & higher-order?


interleavings
control-flow
data-flow

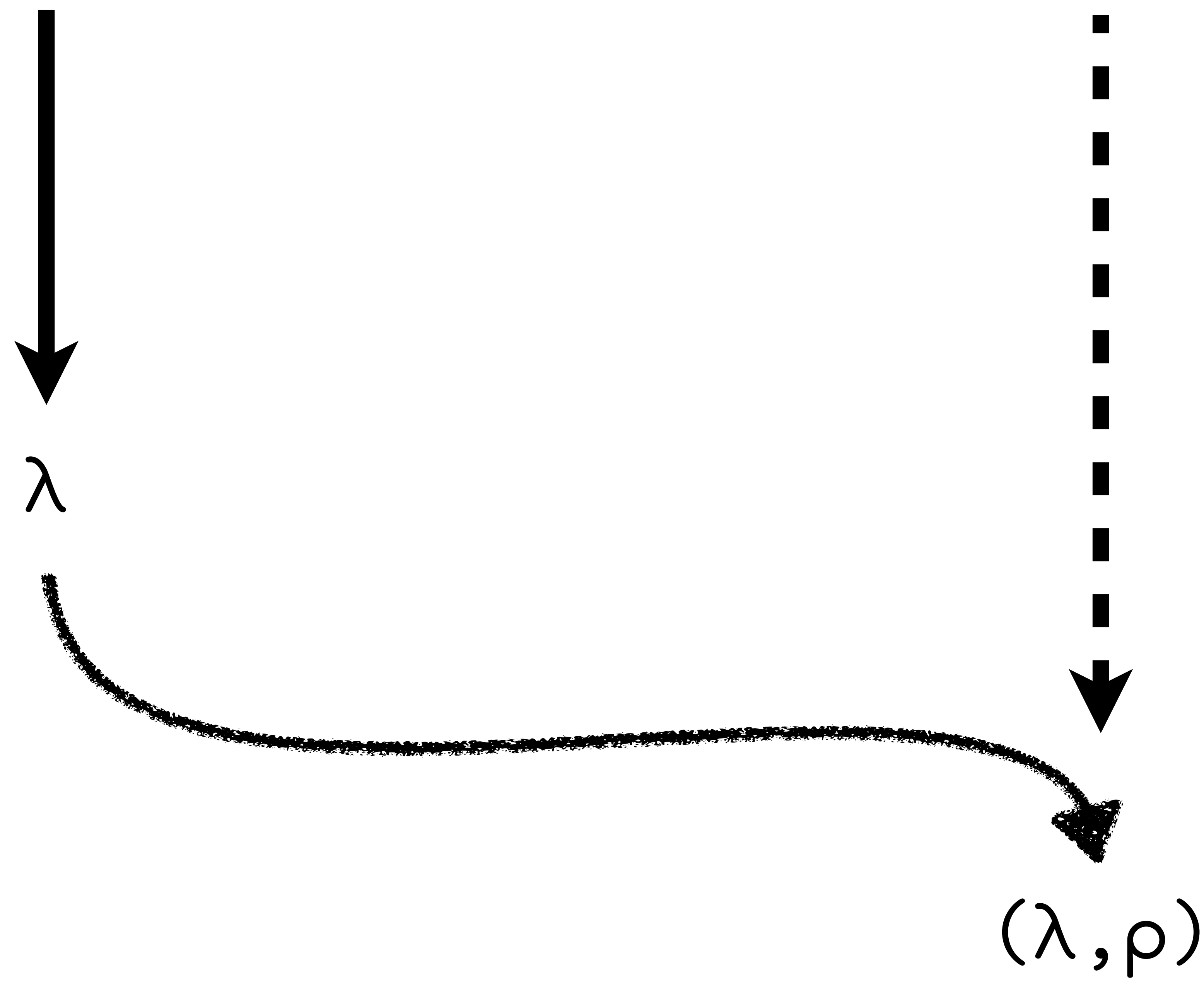
Close in one; call in another.

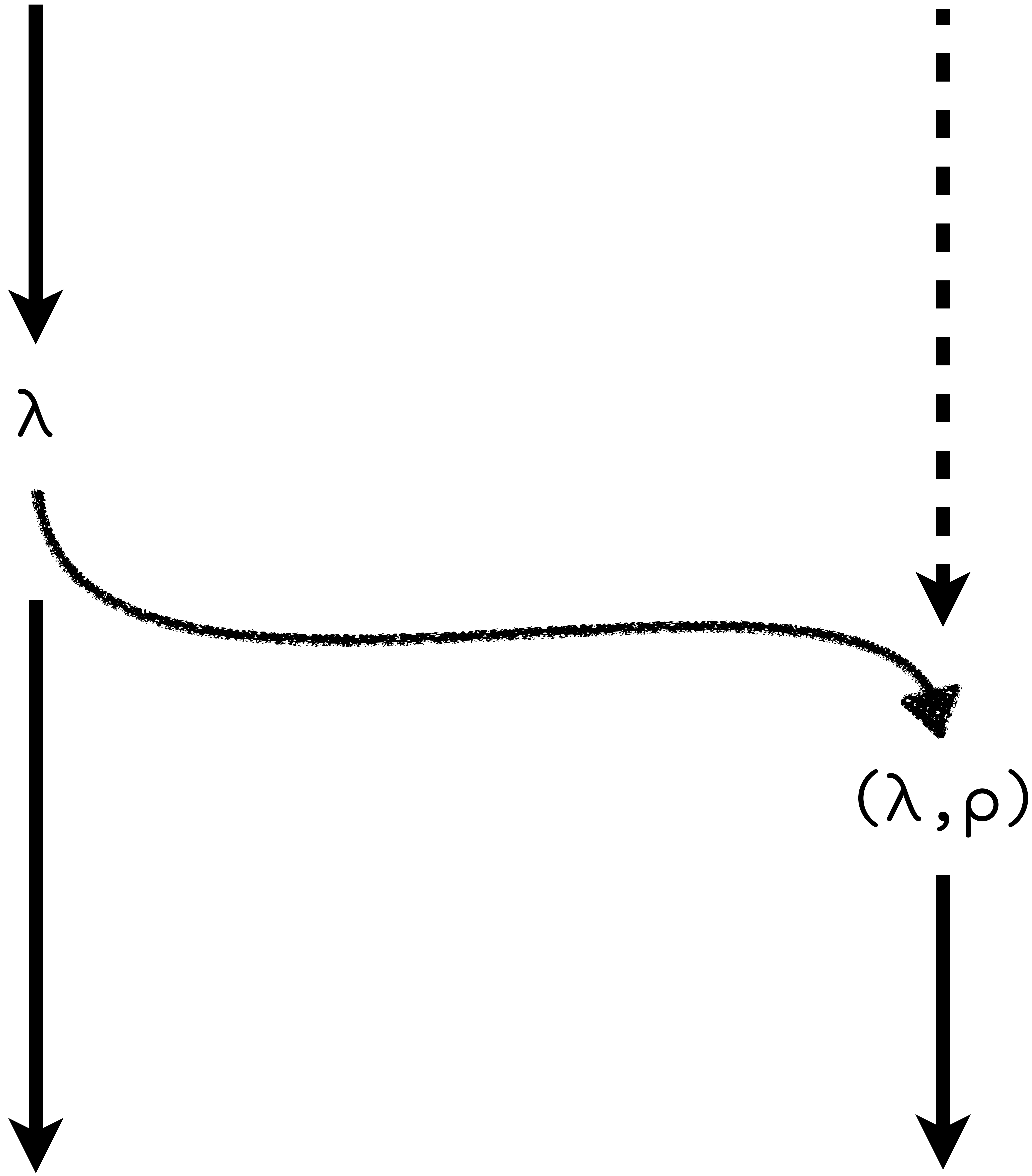
$$\Big\downarrow$$

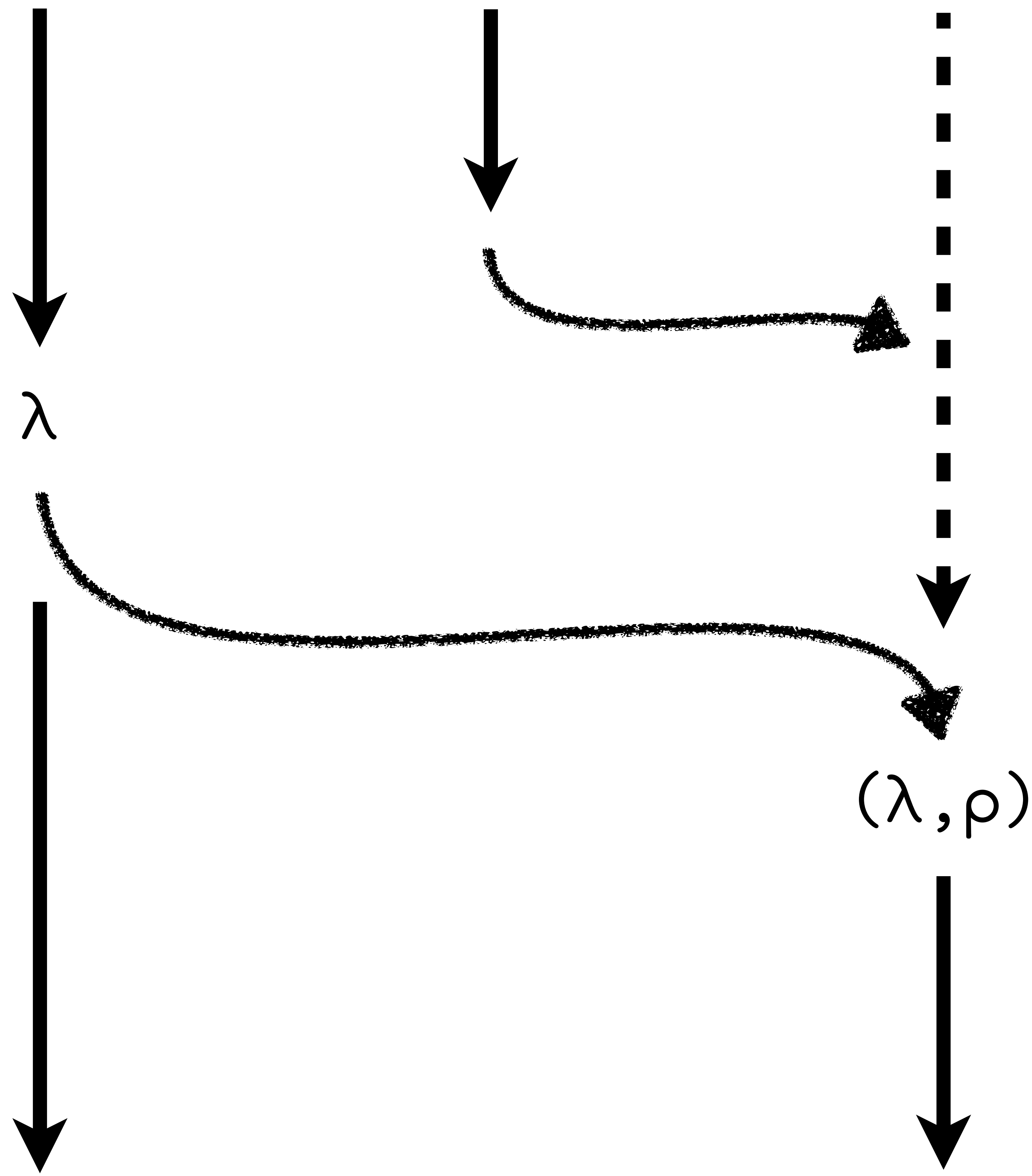
λ

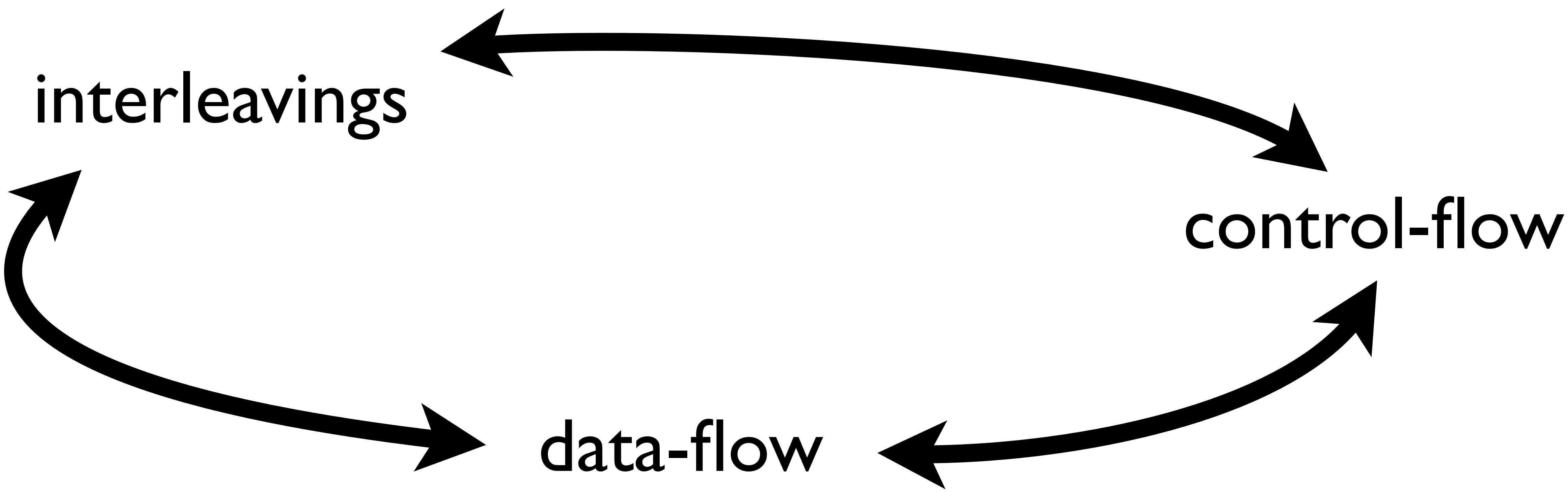
$$\Big\downarrow$$

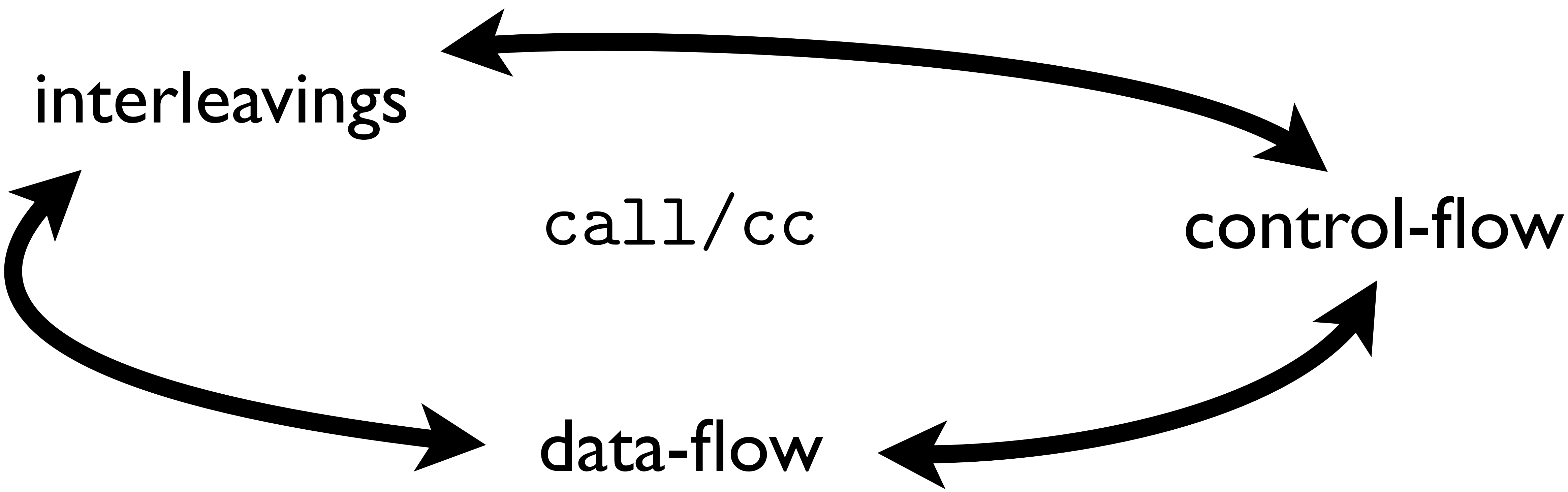
(λ,ρ)

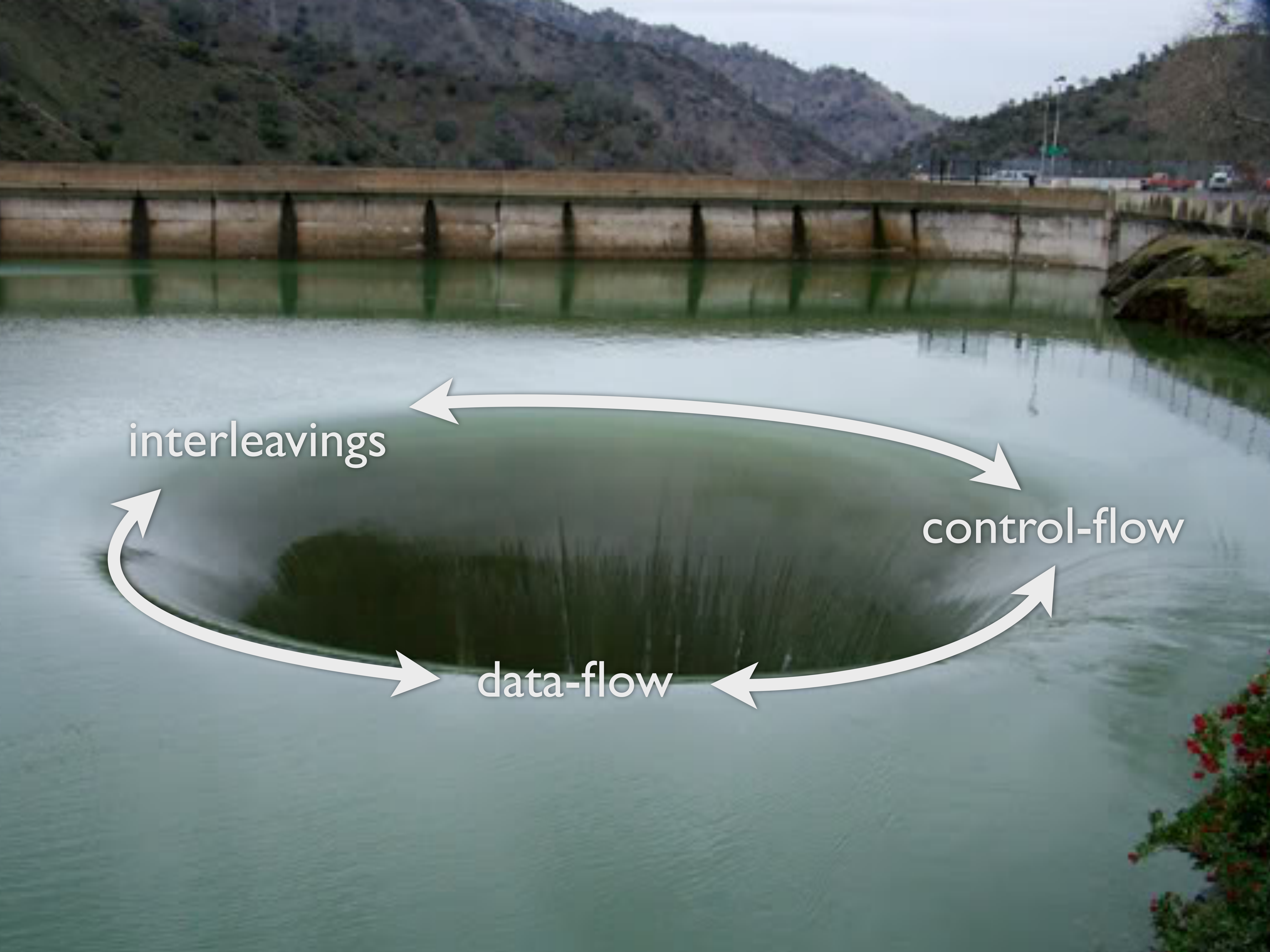

λ
(λ,ρ)


λ
(λ,ρ)


λ
(λ, ρ)


interleavings
control-flow
data-flow


interleavings
call/cc
control-flow
data-flow


interleavings
control-flow
data-flow

Catch in one; throw in another.

```
(call/cc
```

```
(call/cc (λ (cc)
```

```
(call/cc (λ (cc)
  (spawn
```

```
(call/cc (λ (cc)
  (spawn (cc #t))
```

```
(call/cc (λ (cc)
 (spawn (cc #t))
 (cc #f)))
```

Wanted: MHP & CFA

Reëngineer the CESK machine

Reëngineer the CESK machine

Analyze thread ‘shape’ for MHP

Reëngineer the CESK machine

Analyze thread ‘shape’ for MHP

Abstract MHP again for CFA

A-normalized λ-calculus

+ `spawn`, `join`, `cas`

+ `call/cc`, `if`, `set!`

CESK

(Felleisen & Friedman, 1986)

P(CEK)S

P(CEK*)S

P(CEK*)S

$$\Sigma = \mathsf{Exp} \times Env \times Store \times Kont$$

C E S K

$$\underbrace{\Sigma}_{\text{state-space}} = \overset{\text{C}}{\mathsf{Exp}} \times \overset{\text{E}}{Env} \times \overset{\text{S}}{Store} \times \overset{\text{K}}{Kont}$$

state-space

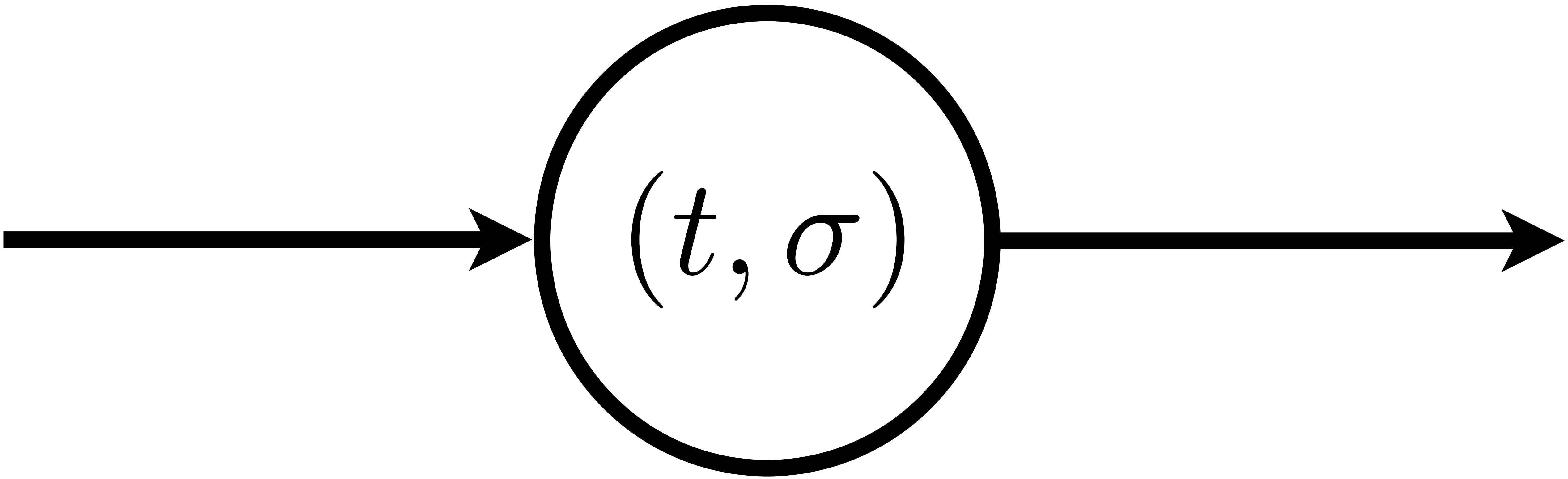
$$\Sigma = \mathsf{Exp} \times Env \times Store \times Kont$$

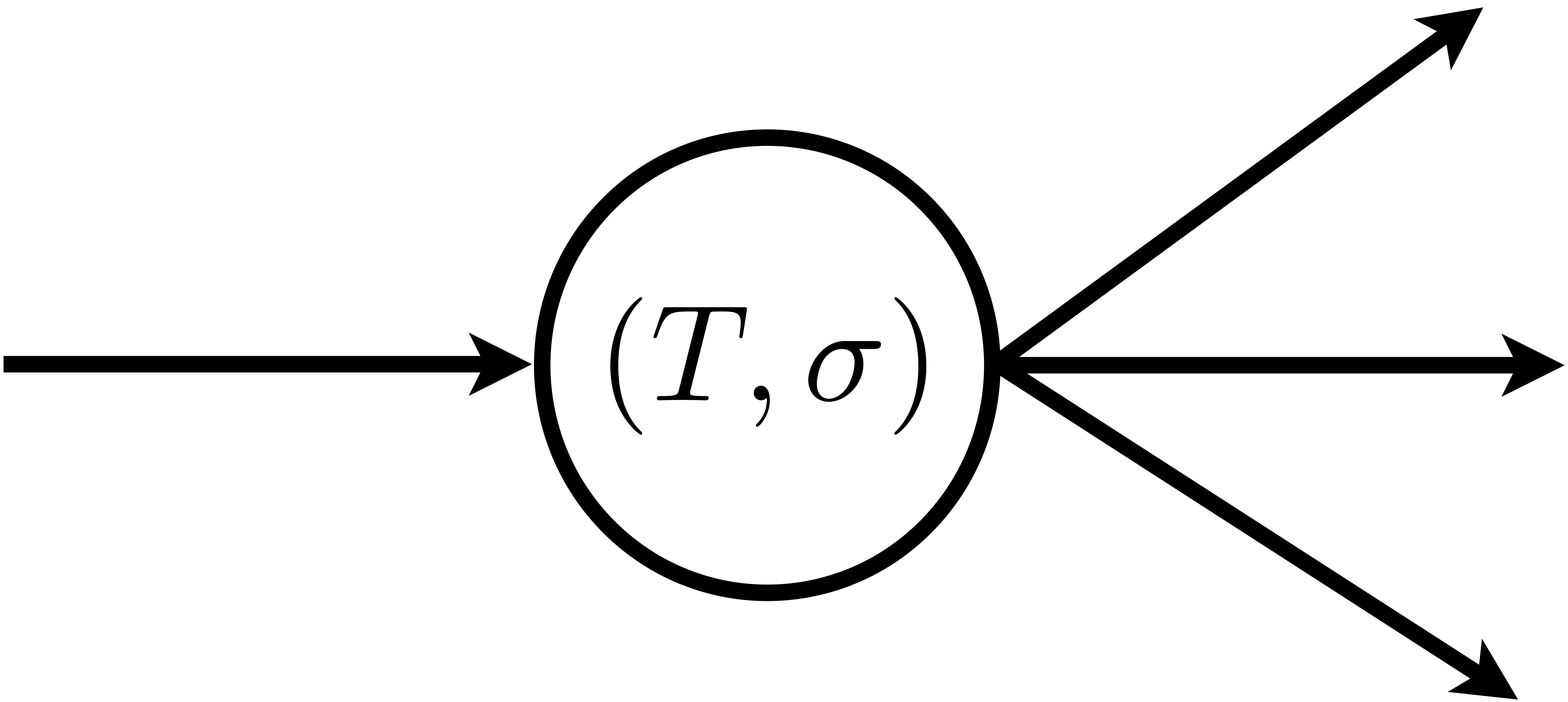
$$\mathsf{Exp} \times Env \times Kont$$

$$Thread = \mathsf{Exp} \times Env \times Kont \times TID$$

$$\Sigma = Thread \times Store$$

$$\Sigma = \mathcal{P}(Thread) \times Store$$


(t, σ)


(T, σ)

Next: Naive abstraction?

$$
\begin{aligned}
\Sigma &= \mathcal{P}(\mathit{Thread}) \times \mathit{Store} \\
\mathit{Thread} &= \mathsf{Exp} \times \mathit{Env} \times \mathit{Kont} \times \mathit{TID} \\
\mathit{Store} &= \mathit{Addr} \rightarrow \mathit{Clo}
\end{aligned}
$$

$$
\begin{aligned}
\mathit{Env} &= \mathsf{Var} \rightarrow \mathit{Addr} \\
\mathit{Clo} &= \mathsf{Lam} \times \mathit{Env} \\
\kappa \in \mathit{Kont} &::= \mathbf{letk}(v, e, \rho, \kappa) \\
&\quad \mid \ \mathbf{halt}
\end{aligned}
$$

$$
\begin{aligned}
\Sigma &= \mathcal{P}(\widehat{Thread}) \times \widehat{Store} \\
\widehat{Thread} &= \mathsf{Exp} \times \widehat{Env} \times \widehat{Kont} \times \widehat{TID} \\
\widehat{Store} &= \widehat{Addr} \to \widehat{Clo}
\end{aligned}
$$

$$
\begin{aligned}
\widehat{Env} &= \mathsf{Var} \to \widehat{Addr} \\
\widehat{Clo} &= \mathsf{Lam} \times \widehat{Env} \\
\kappa \in \widehat{Kont} &::= \mathbf{letk}(v, e, \hat{\rho}, \hat{\kappa}) \\
&\quad\mid \mathbf{halt}
\end{aligned}
$$

$$\Sigma = \mathcal{P}(\widehat{Thread}) \times \widehat{Store}$$
$$\widehat{Thread} = \mathsf{Exp} \times \widehat{Env} \times \widehat{Kont} \times \widehat{TID}$$
$$\widehat{Store} = \widehat{Addr} \to \widehat{Clo}$$

$$\widehat{Env} = \mathsf{Var} \to \widehat{Addr}$$
$$\widehat{Clo} = \mathsf{Lam} \times \widehat{Env}$$
$$\boxed{\kappa \in \widehat{Kont} ::= \mathbf{letk}(v, e, \hat{\rho}, \hat{\kappa})}$$
$$\mid \ \mathbf{halt}$$

$$\boxed{\kappa \in Kont ::= \mathbf{letk}(v, e, \rho, \kappa)}$$

$$\kappa \in Kont ::= \mathbf{letk}(v, e, \rho, \kappa)$$

$$\kappa \in Kont ::= \mathbf{letk}(v, e, \rho, \kappa)$$

$$\hat{\kappa} \in \widehat{Kont} ::= \mathbf{letk}(v, e, \hat{\rho}, \hat{\kappa})$$

Store-allocate $Kont$

(Might, SAS 2010)

$$Store = Addr \rightarrow Clo$$

$$Store = Addr \to Clo + Kont$$

$$Thread = \mathsf{Exp} \times Env \times Kont \times TID$$

$$Thread = \mathsf{Exp} \times Env \times Addr \times TID$$

$$\kappa \in \mathit{Kont} ::= \mathbf{letk}(v, e, \rho, \kappa)$$

$$\kappa \in Kont ::= \mathbf{letk}(v, e, \rho, a)$$

$$\kappa \in Kont ::= \mathbf{letk}(v, e, \rho, a)$$

Now we can abstract!

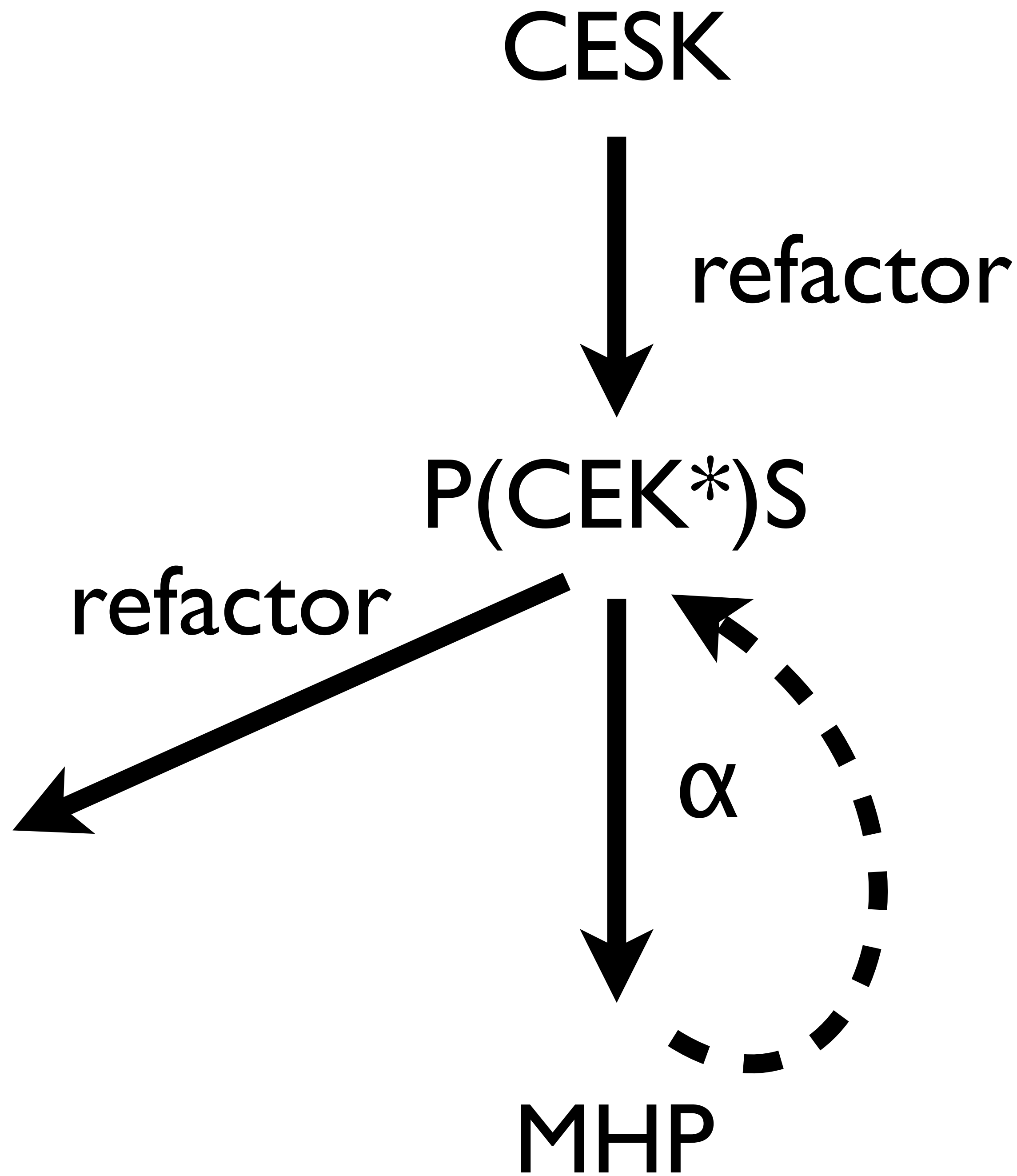
But, wait...

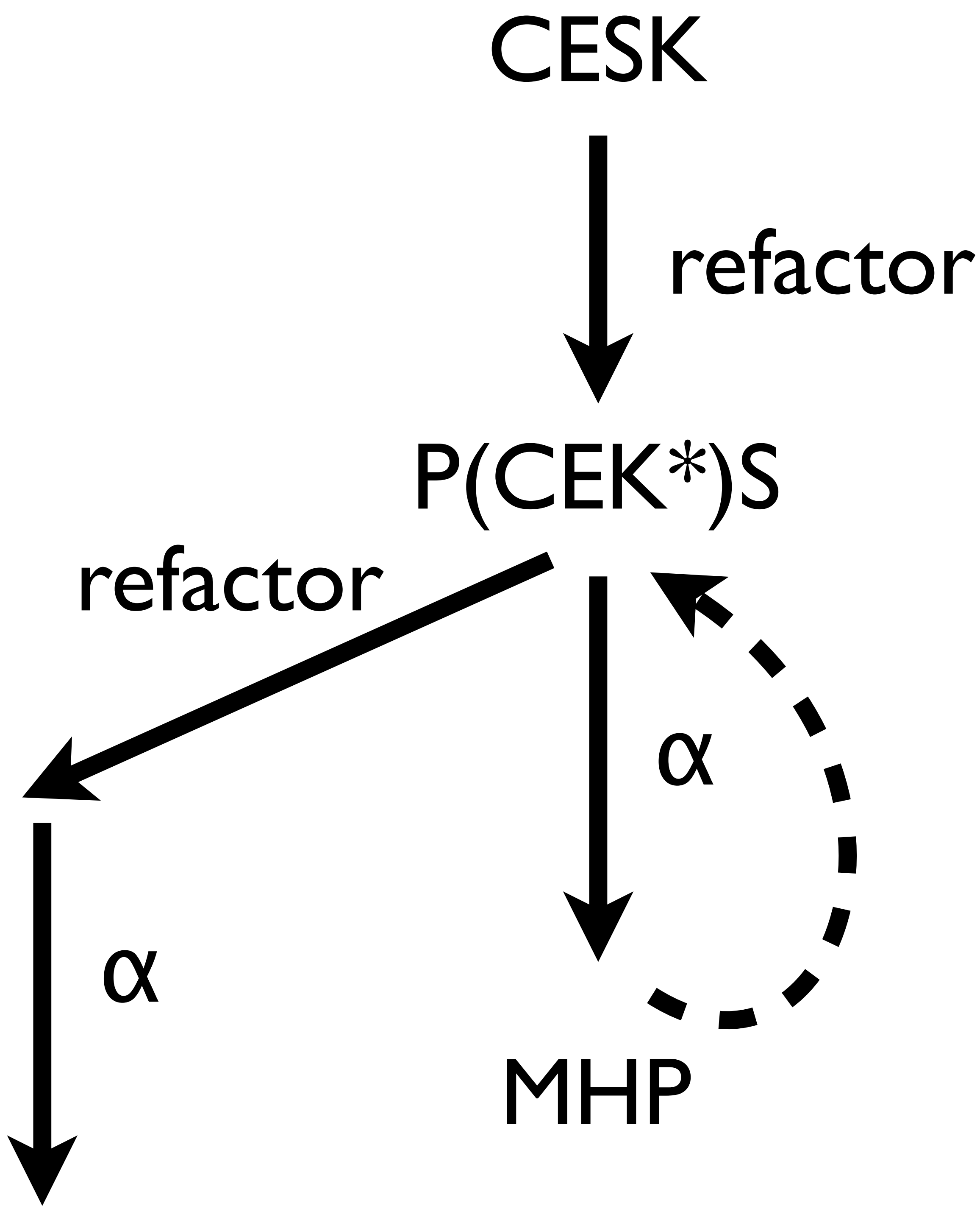
CESK

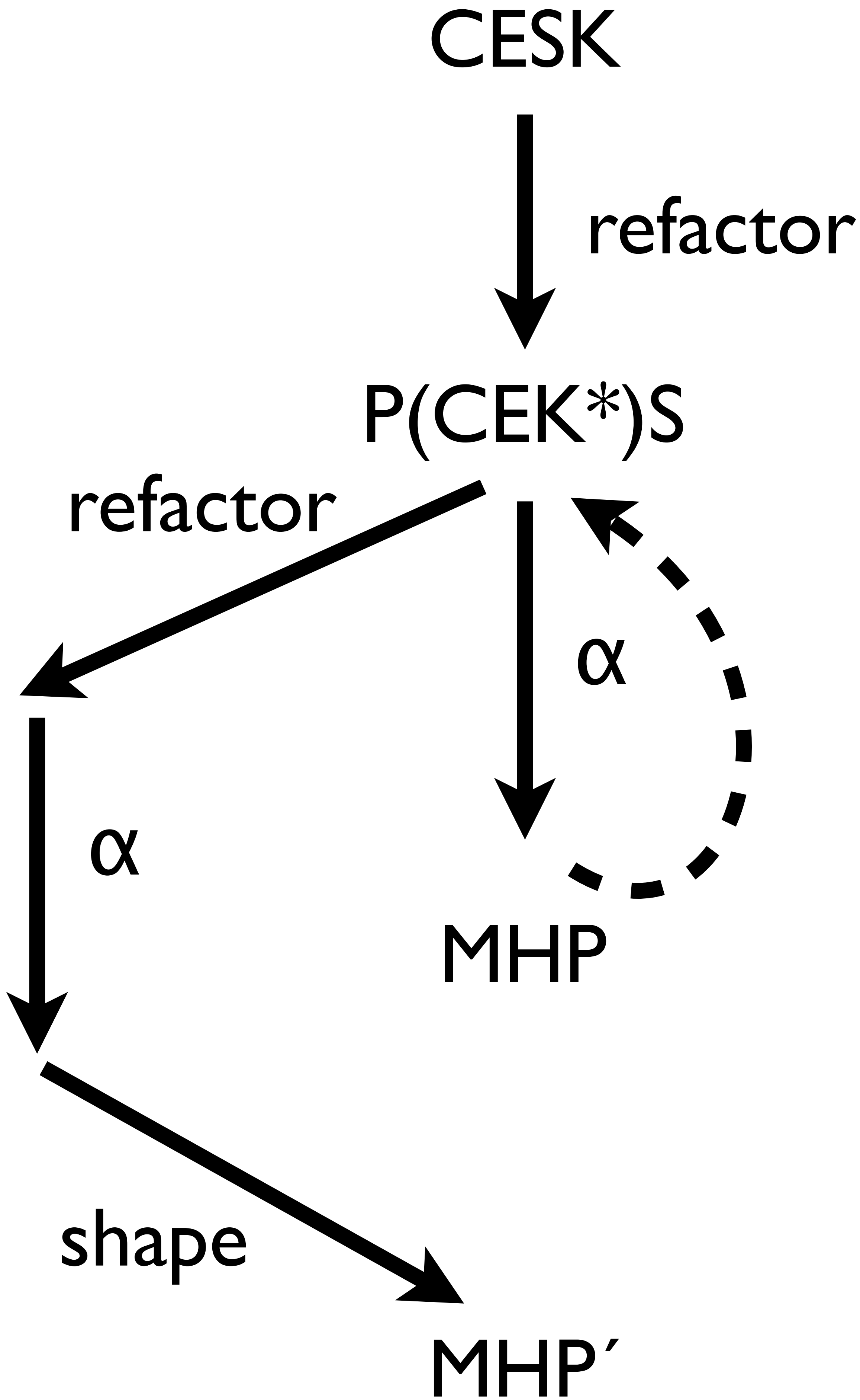
$$\begin{tikzcd} \text{CESK} \arrow[d, "\text{refactor}"] \\ \text{P(CEK*)S} \end{tikzcd}$$

$$
\begin{tikzcd}
\text{CESK} \arrow[d, "\text{refactor}"] \\
\text{P(CEK*)S} \arrow[d, "\alpha"] \\
\text{MHP}
\end{tikzcd}
$$

$$\begin{tikzcd}
\mathrm{CESK} \arrow[d, "\text{refactor}"] \\
\mathrm{P(CEK^*)S} \arrow[d, "\alpha"] \\
\mathrm{MHP} \arrow[u, bend right, dashed]
\end{tikzcd}$$


CESK
refactor
P(CEK*)S
refactor
α
MHP


CESK
refactor
P(CEK*)S
refactor
α
α
MHP


CESK
refactor
P(CEK*)S
refactor
α
α
MHP
shape
MHP′

$$\Sigma = \mathcal{P}(Thread) \times Store$$

$$\mathcal{P}(\mathit{Thread})$$

$$\mathcal{P}(\mathit{Thread}) = \mathcal{P}(\mathsf{Exp} \times \mathit{Env} \times \mathit{Addr} \times \mathit{TID})$$

$$
\begin{aligned}
\mathcal{P}(\mathit{Thread}) &= \mathcal{P}(\mathsf{Exp} \times \mathit{Env} \times \mathit{Addr} \times \mathit{TID}) \\
&\cong \mathit{TID} \to \mathcal{P}(\mathsf{Exp} \times \mathit{Env} \times \mathit{Addr})
\end{aligned}
$$

$$
\begin{aligned}
\mathcal{P}(\mathit{Thread}) &= \mathcal{P}(\mathsf{Exp} \times \mathit{Env} \times \mathit{Addr} \times \mathit{TID}) \\
&\cong \mathit{TID} \to \mathcal{P}(\mathsf{Exp} \times \mathit{Env} \times \mathit{Addr}) \\
&\approx \mathit{TID} \to \mathsf{Exp} \times \mathit{Env} \times \mathit{Addr}
\end{aligned}
$$

$$
\begin{aligned}
\mathcal{P}(\mathit{Thread}) &= \mathcal{P}(\mathsf{Exp} \times \mathit{Env} \times \mathit{Addr} \times \mathit{TID}) \\
&\cong \mathit{TID} \to \mathcal{P}(\mathsf{Exp} \times \mathit{Env} \times \mathit{Addr}) \\
&\approx \mathit{TID} \to \mathsf{Exp} \times \mathit{Env} \times \mathit{Addr} \\
&= \mathit{Threads}
\end{aligned}
$$

$$\Sigma = Threads \times Store$$

$$\begin{aligned}
\Sigma &= \mathit{Threads} \times \mathit{Store} \\
\mathit{Threads} &= \mathit{TID} \rightarrow \mathsf{Exp} \times \mathit{Env} \times \mathit{Addr} \\
\mathit{Store} &= \mathit{Addr} \rightarrow \mathit{Clo} + \mathit{Kont}
\end{aligned}$$

$$\begin{aligned}
\mathit{Env} &= \mathsf{Var} \rightarrow \mathit{Addr} \\
\mathit{Clo} &= \mathsf{Lam} \times \mathit{Env} \\
\mathit{Kont} &::= \mathbf{letk}(v, e, \rho, a) \\
&\quad\mid \mathbf{halt}
\end{aligned}$$

$$\begin{aligned}
\Sigma &= \mathit{Threads} \times \mathit{Store} \\
\mathit{Threads} &= \mathit{TID} \rightharpoonup \mathsf{Exp} \times \mathit{Env} \times \mathit{Addr} \\
\mathit{Store} &= \mathit{Addr} \rightharpoonup \mathit{Clo} + \mathit{Kont}
\end{aligned}$$

$$\begin{aligned}
\mathit{Env} &= \mathsf{Var} \rightharpoonup \mathit{Addr} \\
\mathit{Clo} &= \mathsf{Lam} \times \mathit{Env} \\
\mathit{Kont} &::= \mathbf{letk}(v, e, \rho, a) \\
&\quad\mid \mathbf{halt}
\end{aligned}$$

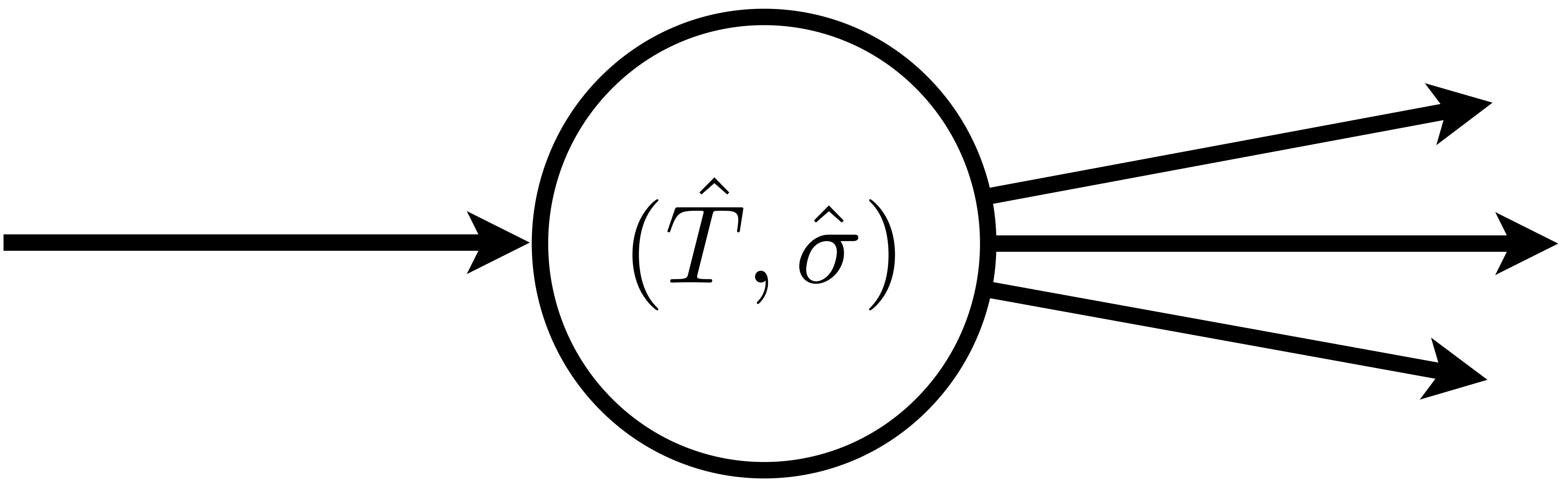
$$
\begin{aligned}
\Sigma &= \mathit{Threads} \times \mathit{Store} \\
\mathit{Threads} &= \mathit{TID} \to \mathcal{P}(\mathsf{Exp} \times \mathit{Env} \times \mathit{Addr}) \\
\mathit{Store} &= \mathit{Addr} \to \mathcal{P}(\mathit{Clo} + \mathit{Kont})
\end{aligned}
$$

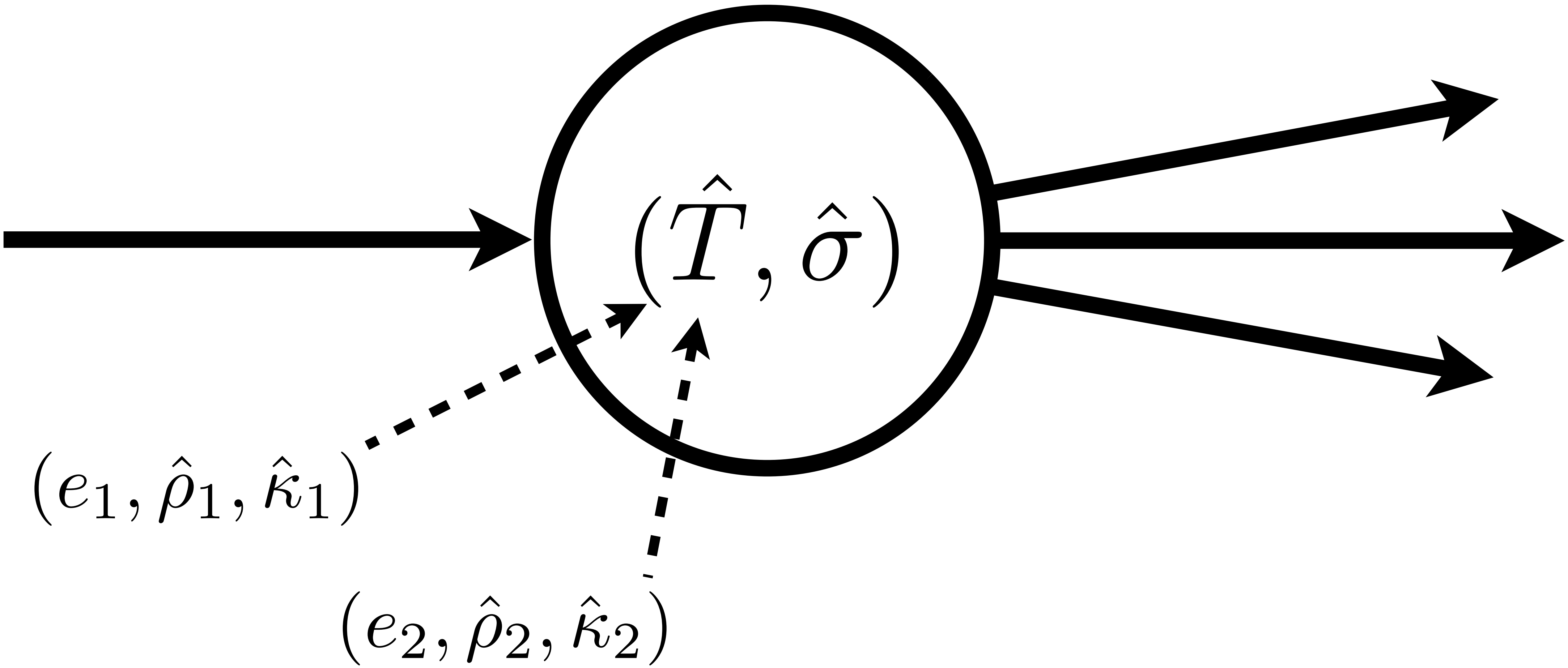
$$
\begin{aligned}
\mathit{Env} &= \mathsf{Var} \to \mathit{Addr} \\
\mathit{Clo} &= \mathsf{Lam} \times \mathit{Env} \\
\mathit{Kont} &::= \mathbf{letk}(v, e, \rho, a) \\
&\quad\mid\ \mathbf{halt}
\end{aligned}
$$

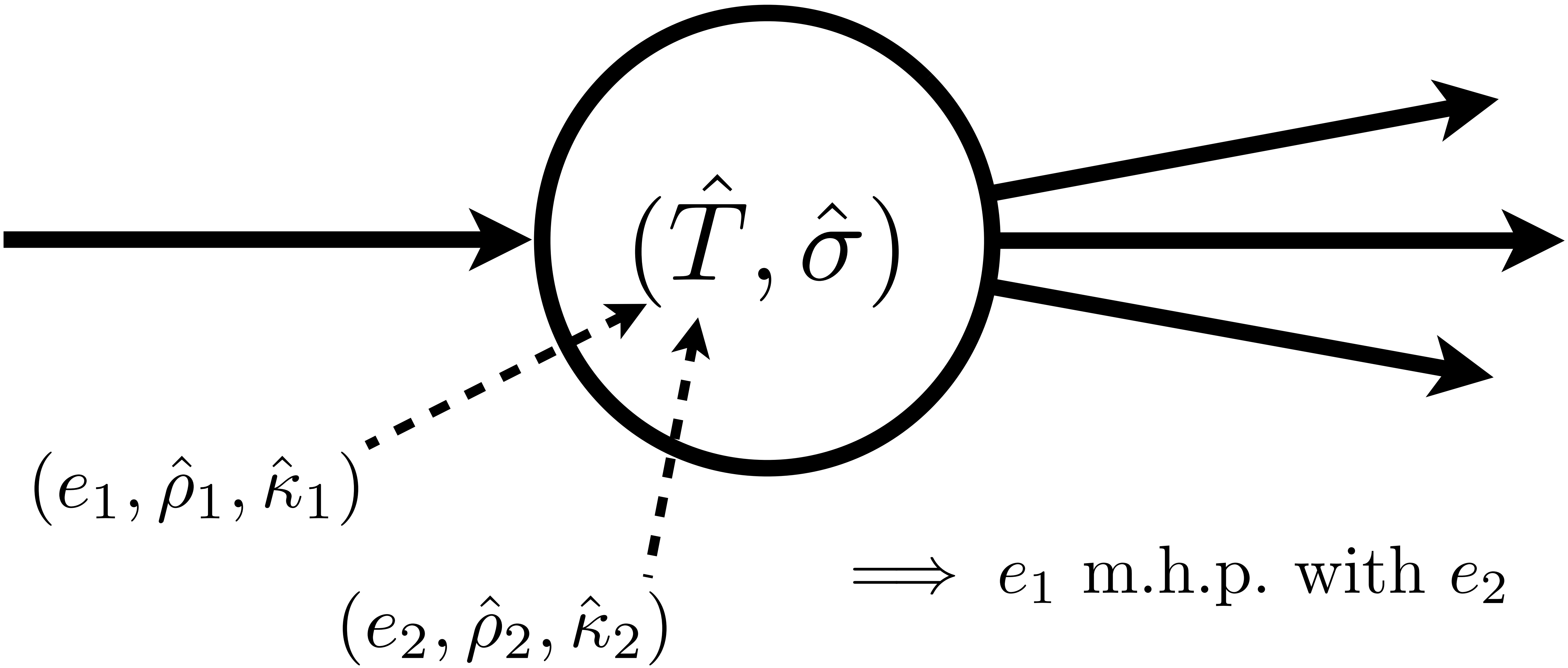
$$
\begin{aligned}
\hat{\Sigma} &= \widehat{Threads} \times \widehat{Store} \\
\widehat{Threads} &= \widehat{TID} \rightarrow \mathcal{P}(\mathsf{Exp} \times \widehat{Env} \times \widehat{Addr}) \\
\widehat{Store} &= \widehat{Addr} \rightarrow \mathcal{P}(\widehat{Clo} + \widehat{Kont})
\end{aligned}
$$

$$
\begin{aligned}
\widehat{Env} &= \mathsf{Var} \rightarrow \widehat{Addr} \\
\widehat{Clo} &= \mathsf{Lam} \times \widehat{Env} \\
\widehat{Kont} &::= \mathbf{letk}(v, e, \hat{\rho}, \hat{a}) \\
&\mid\ \mathbf{halt}
\end{aligned}
$$

Analysis: MHP


$(\hat{T}, \hat{\sigma})$

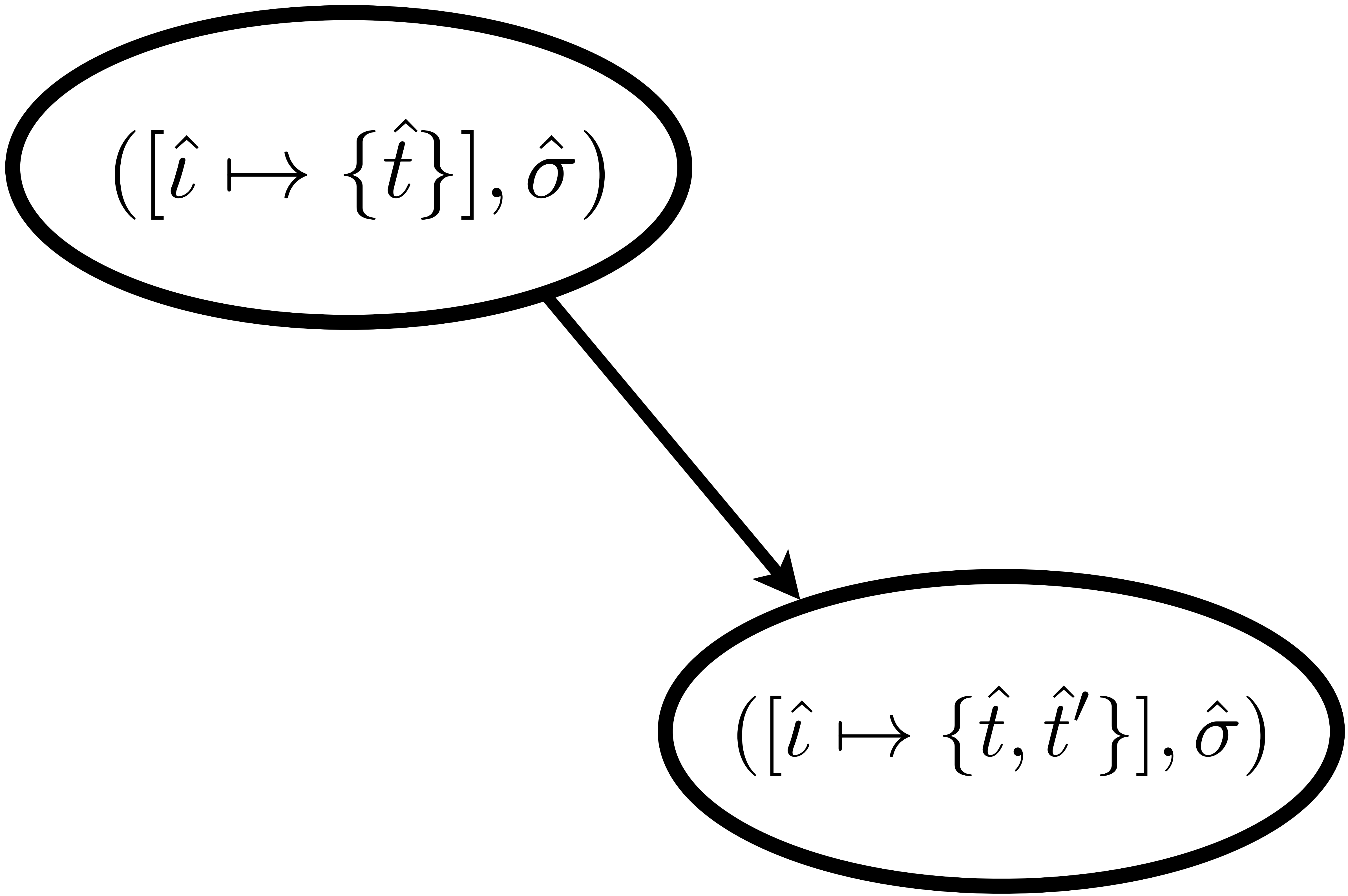

$(\hat{T}, \hat{\sigma})$
$(e_1, \hat{\rho}_1, \hat{\kappa}_1)$
$(e_2, \hat{\rho}_2, \hat{\kappa}_2)$


$(\hat{T}, \hat{\sigma})$
$(e_1, \hat{\rho}_1, \hat{\kappa}_1)$
$(e_2, \hat{\rho}_2, \hat{\kappa}_2)$
$\implies e_1$ m.h.p. with e_2

Problem: Precision

$\widehat{Threads}$ grow monotonically

$$([\hat{\imath} \mapsto \{\hat{t}\}], \hat{\sigma})$$


$([\hat{\imath} \mapsto \{\hat{t}\}], \hat{\sigma})$
$([\hat{\imath} \mapsto \{\hat{t}, \hat{t}'\}], \hat{\sigma})$

Solution: Shape analysis

$$\hat{\Sigma} = \widehat{Threads} \times \widehat{Store}$$

(Chase et al., 1990)

$$\hat{\Sigma} = \widehat{Threads} \times \widehat{Store} \times \widehat{ACount}$$

(Chase et al., 1990)

$$\hat{\Sigma} = \widehat{Threads} \times \widehat{Store} \times \widehat{ACount}$$

$$\widehat{ACount} = \widehat{Addr} \rightarrow \{0, 1, \infty\}$$

(Chase et al., 1990)

$$\hat{\Sigma} = \widehat{Threads} \times \widehat{Store} \times \widehat{ACount}$$

$$\widehat{ACount} = \widehat{Addr} \rightarrow \{0, 1, \infty\}$$

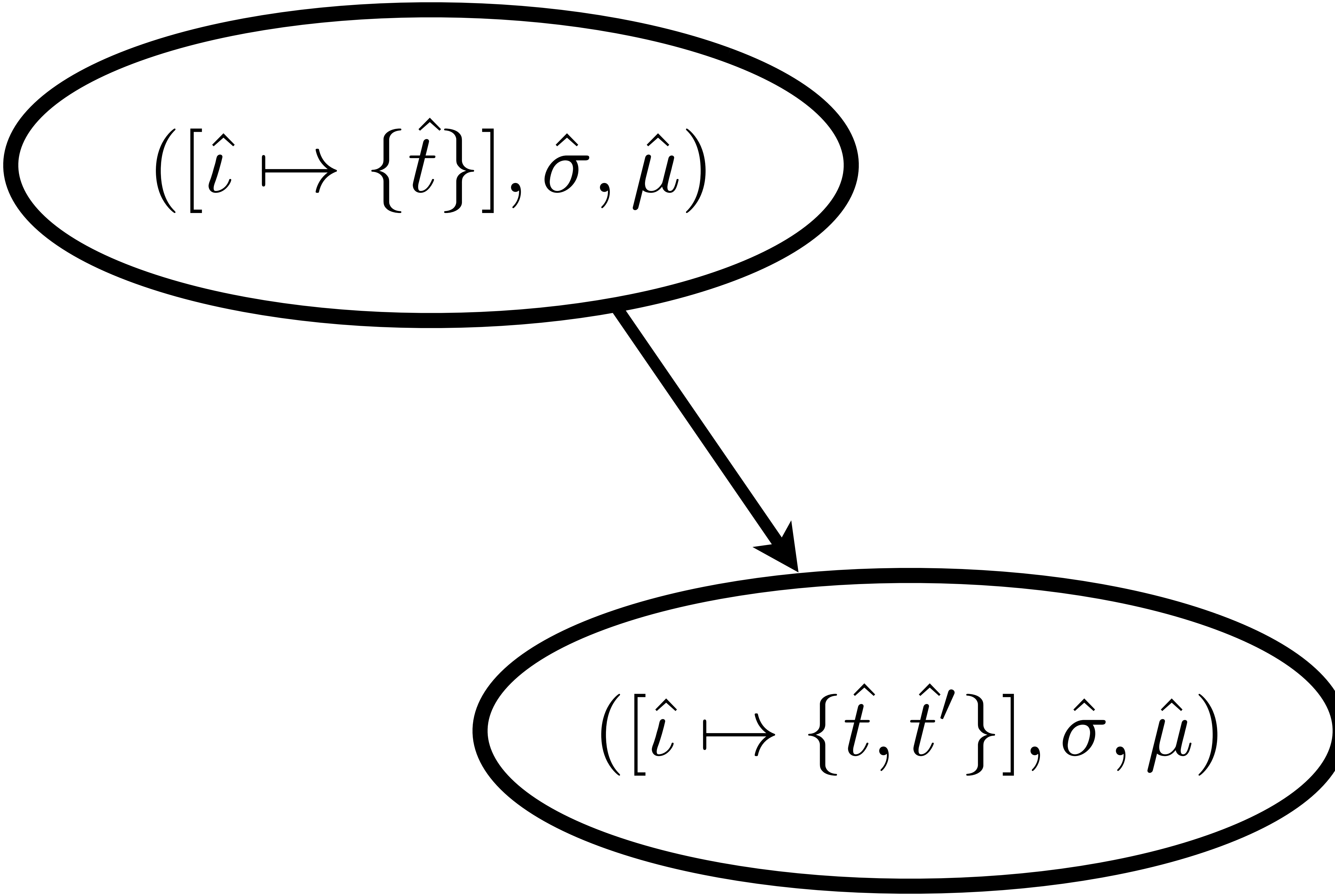
(Chase et al., 1990)

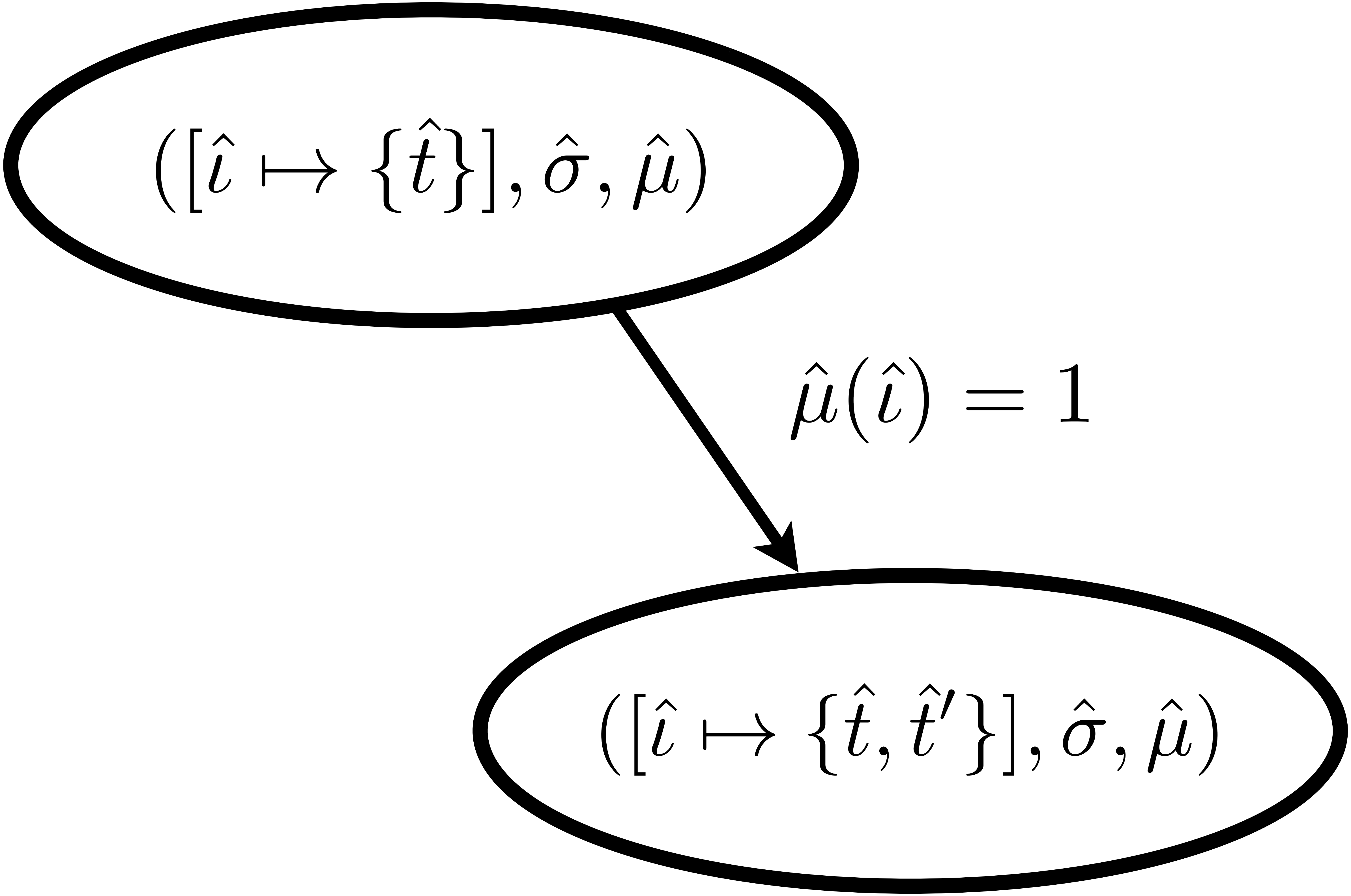
Strong update

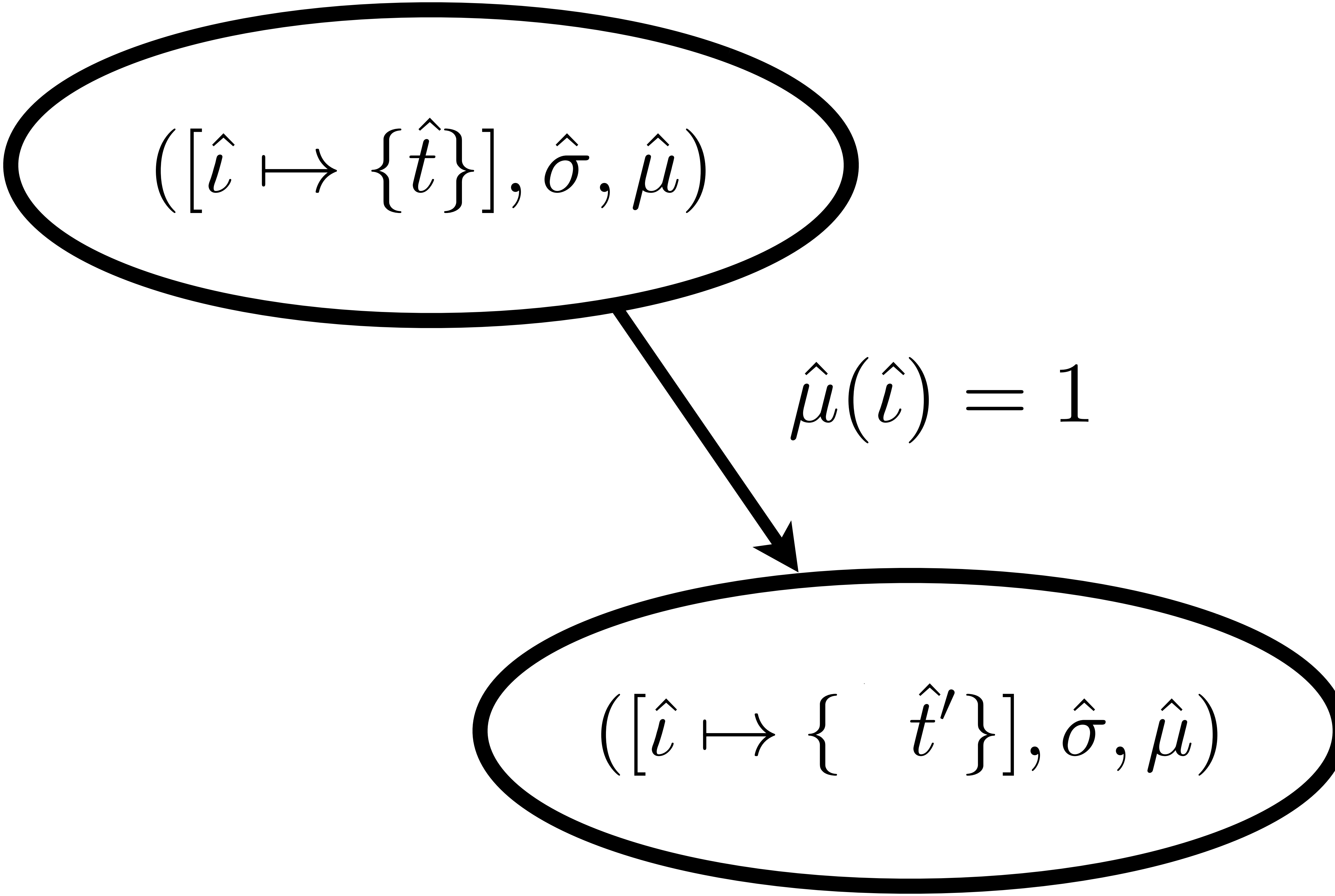
Strong transition

$$\widehat{TCount} = \widehat{TID} \to \{0, 1, \infty\}$$

$$\hat{\Sigma} = \widehat{Threads} \times \widehat{Store} \times \widehat{TCount}$$


$([\hat{\imath} \mapsto \{\hat{t}\}], \hat{\sigma}, \hat{\mu})$
$([\hat{\imath} \mapsto \{\hat{t}, \hat{t}'\}], \hat{\sigma}, \hat{\mu})$


$([\hat{\imath} \mapsto \{\hat{t}\}], \hat{\sigma}, \hat{\mu})$
$\hat{\mu}(\hat{\imath}) = 1$
$([\hat{\imath} \mapsto \{\hat{t}, \hat{t}'\}], \hat{\sigma}, \hat{\mu})$


$([\hat{\imath} \mapsto \{\hat{t}\}], \hat{\sigma}, \hat{\mu})$
$\hat{\mu}(\hat{\imath}) = 1$
$([\hat{\imath} \mapsto \{\hat{t}'\}], \hat{\sigma}, \hat{\mu})$

Analysis: CFA

$$\lambda \xrightarrow{?} \texttt{(f e)}$$

Issue: MHP is overkill

Solution: Abstract again

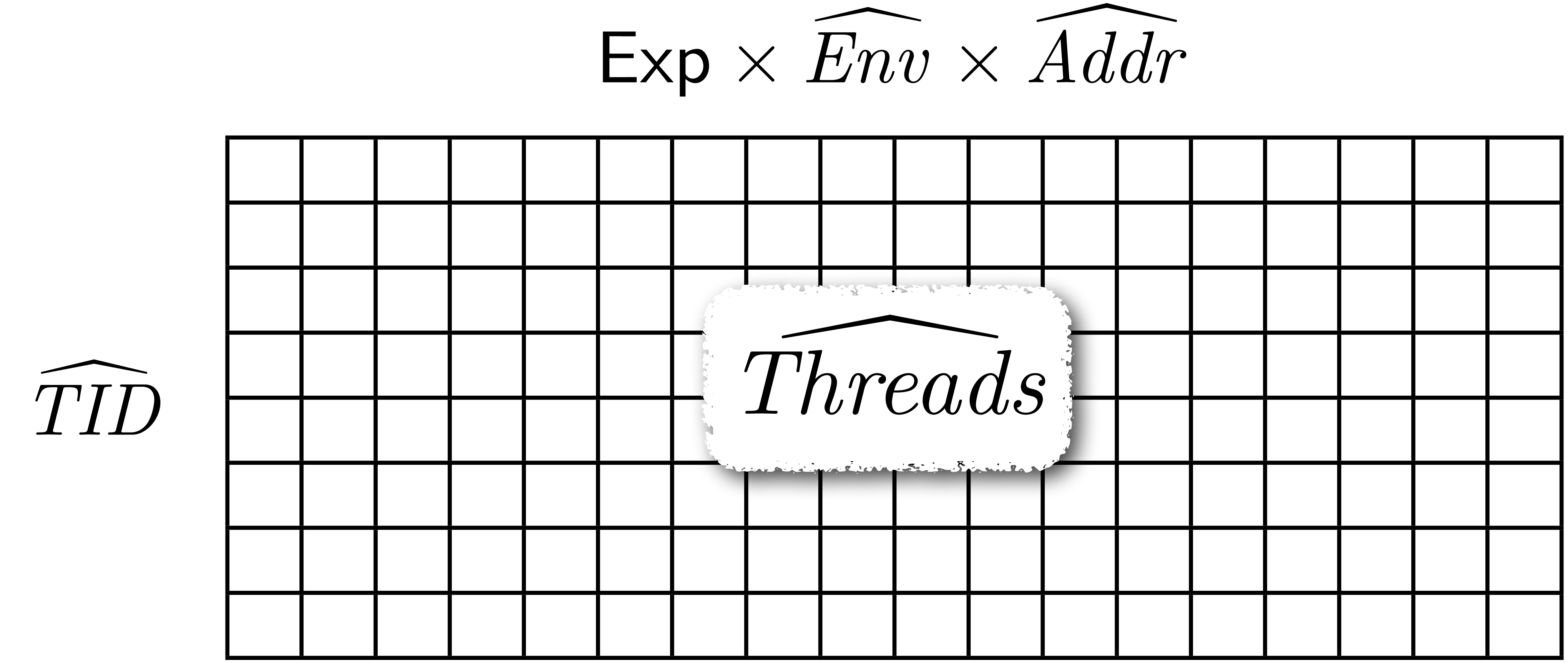
$$\alpha_2(\hat{S}) = \bigsqcup \hat{S}$$

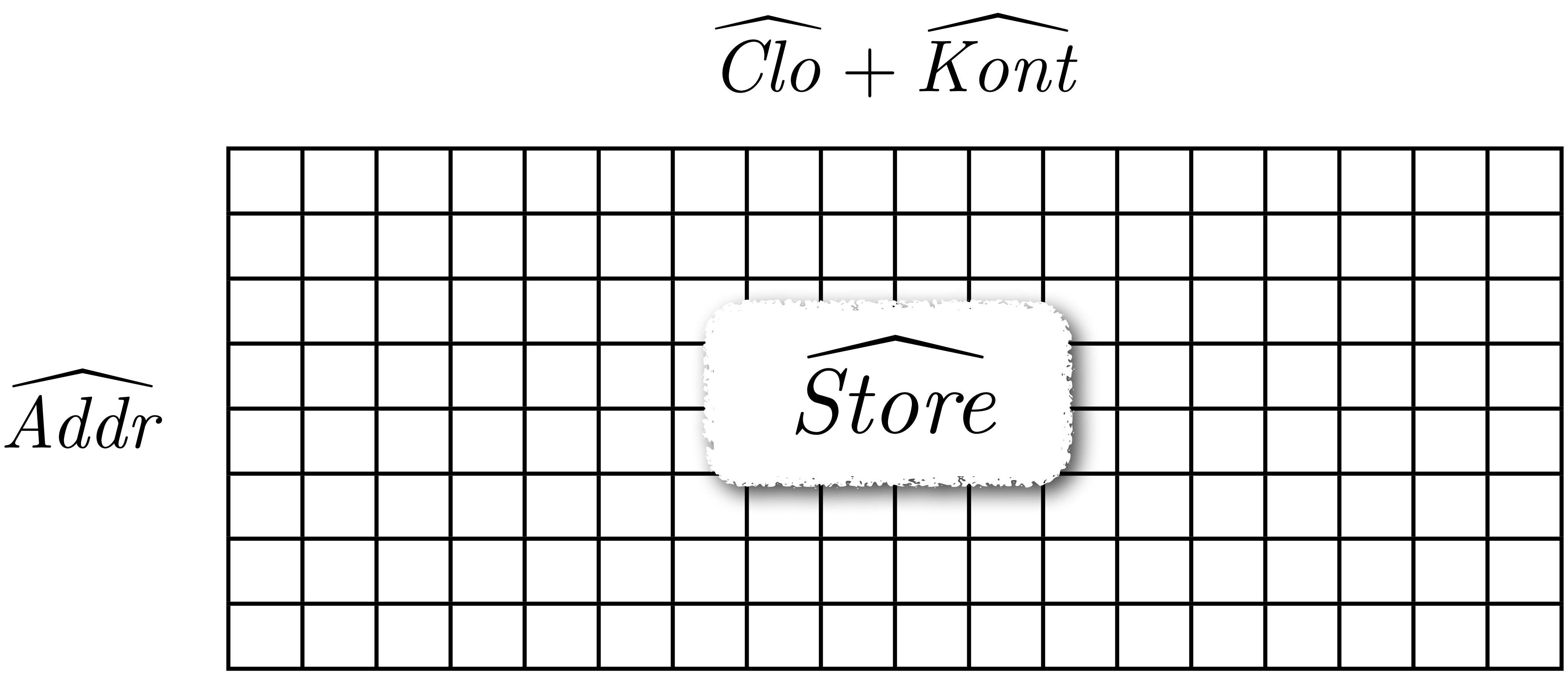
$$\alpha_2 : \mathcal{P}(\hat{\Sigma}) \to \hat{\Sigma}$$

$$\hat{\Sigma} = \widehat{Threads} \times \widehat{Store}$$

$$\widehat{Threads} = \widehat{TID} \to \mathcal{P}(\mathsf{Exp} \times \widehat{Env} \times \widehat{Addr})$$

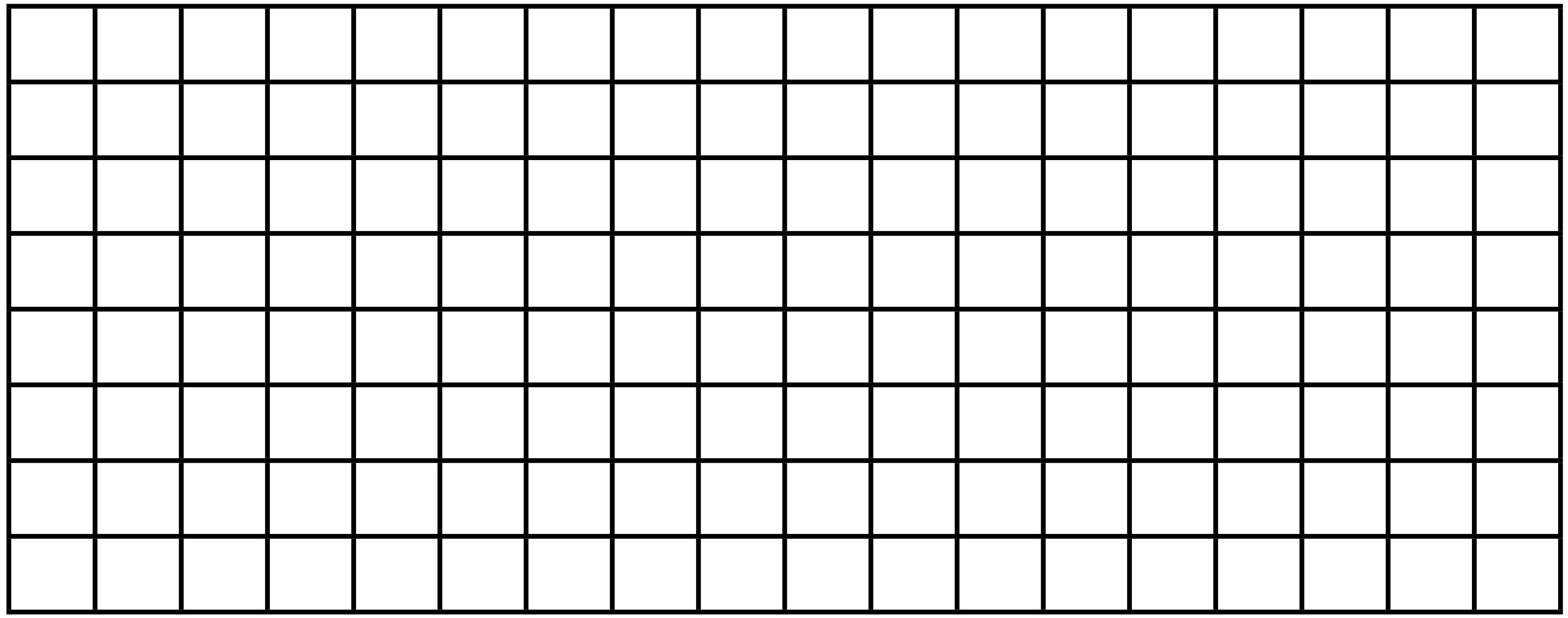
$$\widehat{Store} = \widehat{Addr} \to \mathcal{P}(\widehat{Clo} + \widehat{Kont})$$


$\mathsf{Exp} \times \widehat{Env} \times \widehat{Addr}$
$\widehat{TID}$
$\widehat{Threads}$


$\widehat{Clo} + \widehat{Kont}$
$\widehat{Addr}$
$\widehat{Store}$

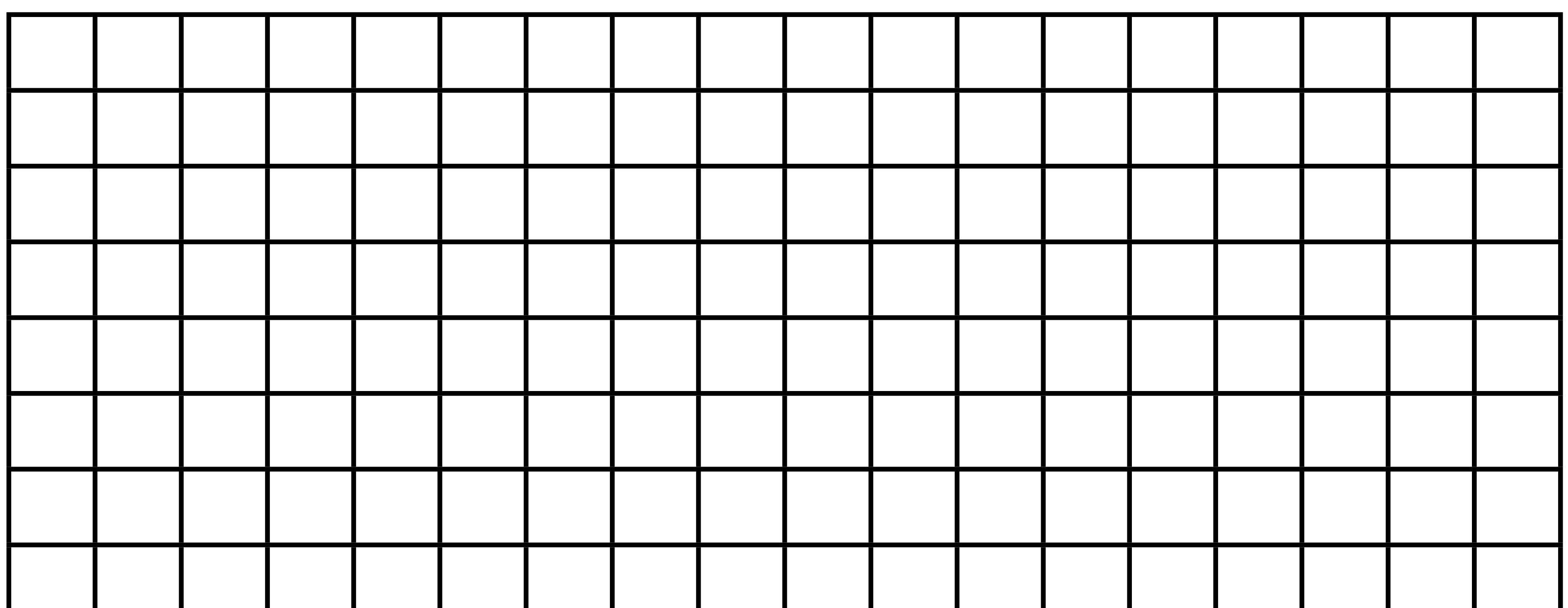
$$\mathsf{Exp} \times \widehat{Env} \times \widehat{Addr}$$

$$\widehat{TID}$$



$$\widehat{Clo} + \widehat{Kont}$$

$$\widehat{Addr}$$



$$\mathsf{Exp} \times \widehat{Env} \times \widehat{Addr}$$

$$\widehat{TID}$$

										√							
				√													
								√						√			
		√															
					√												
											√						
√																√	

$$\widehat{Clo} + \widehat{Kont}$$

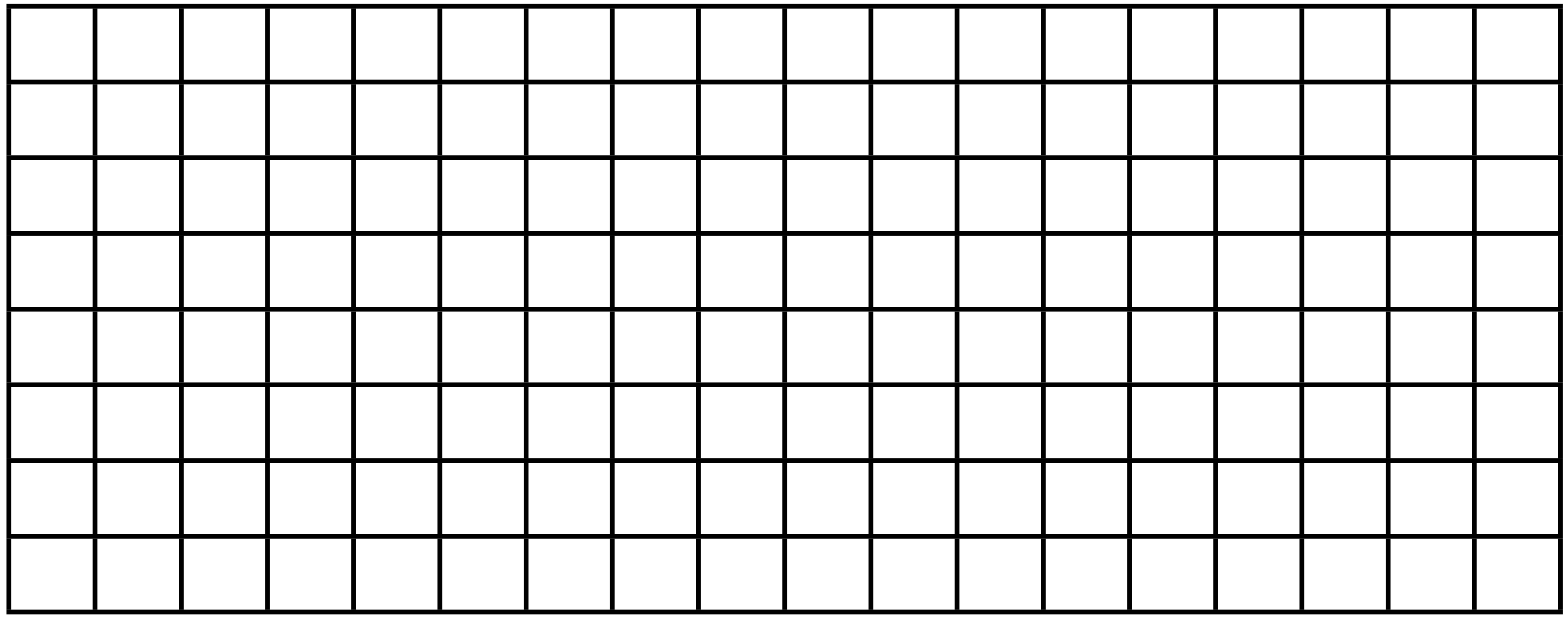
$$\widehat{Addr}$$

																	√
	√					√											
														√			
					√					√							
		√															
											√						√
				√													

Monovariant collapse

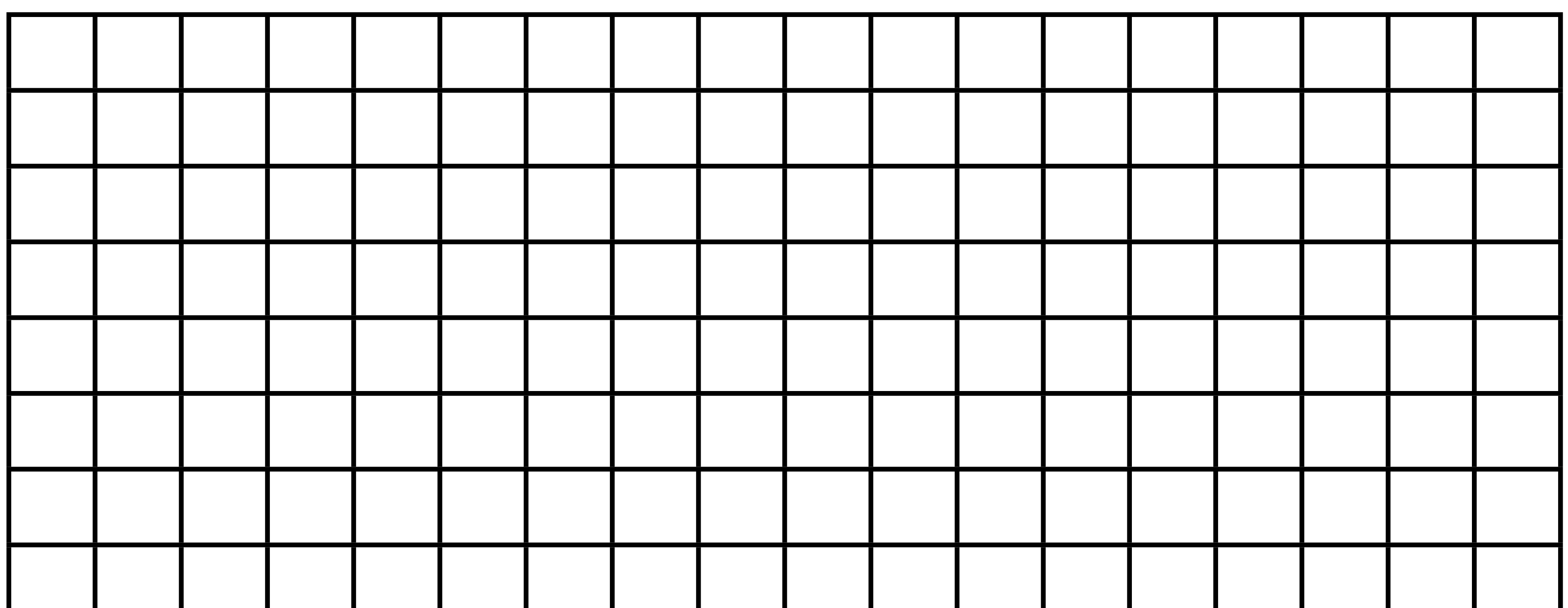
$$\mathsf{Exp} \times \widehat{Env} \times \widehat{Addr}$$

$$\widehat{TID}$$



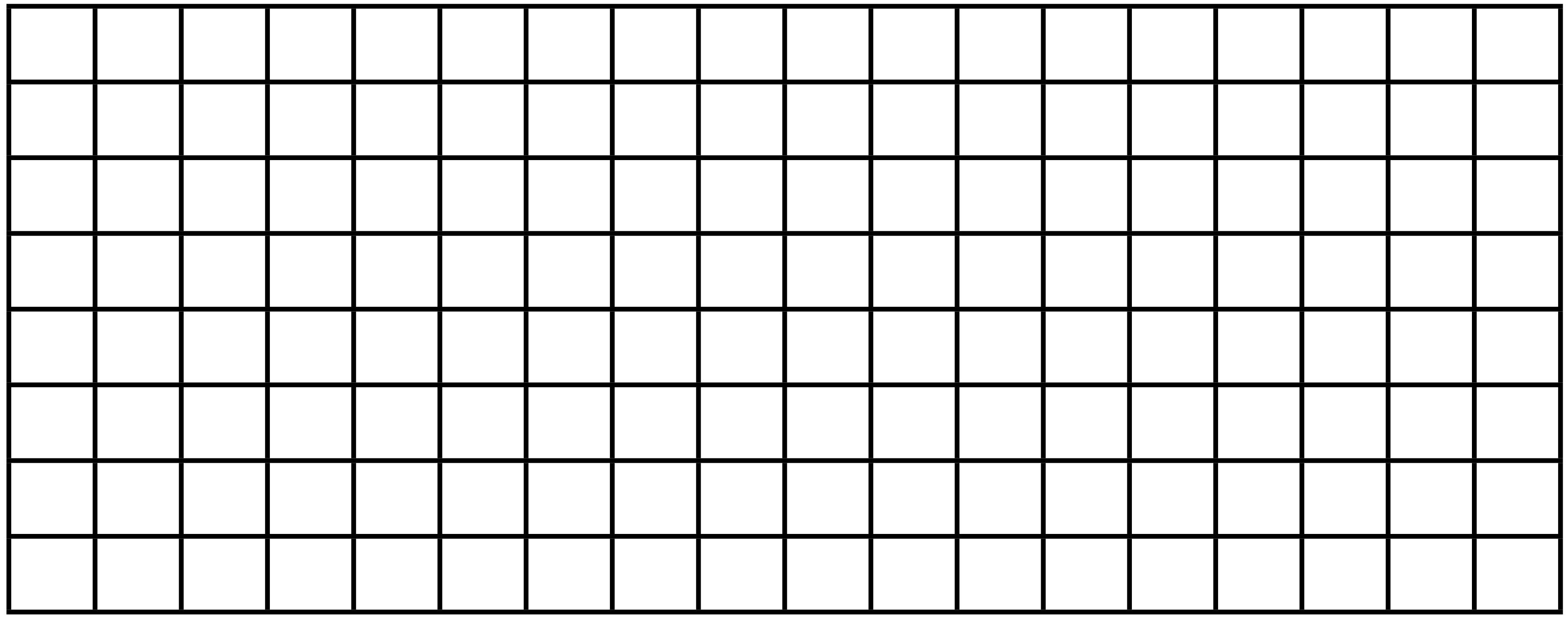
$$\widehat{Clo} + \widehat{Kont}$$

$$\widehat{Addr}$$



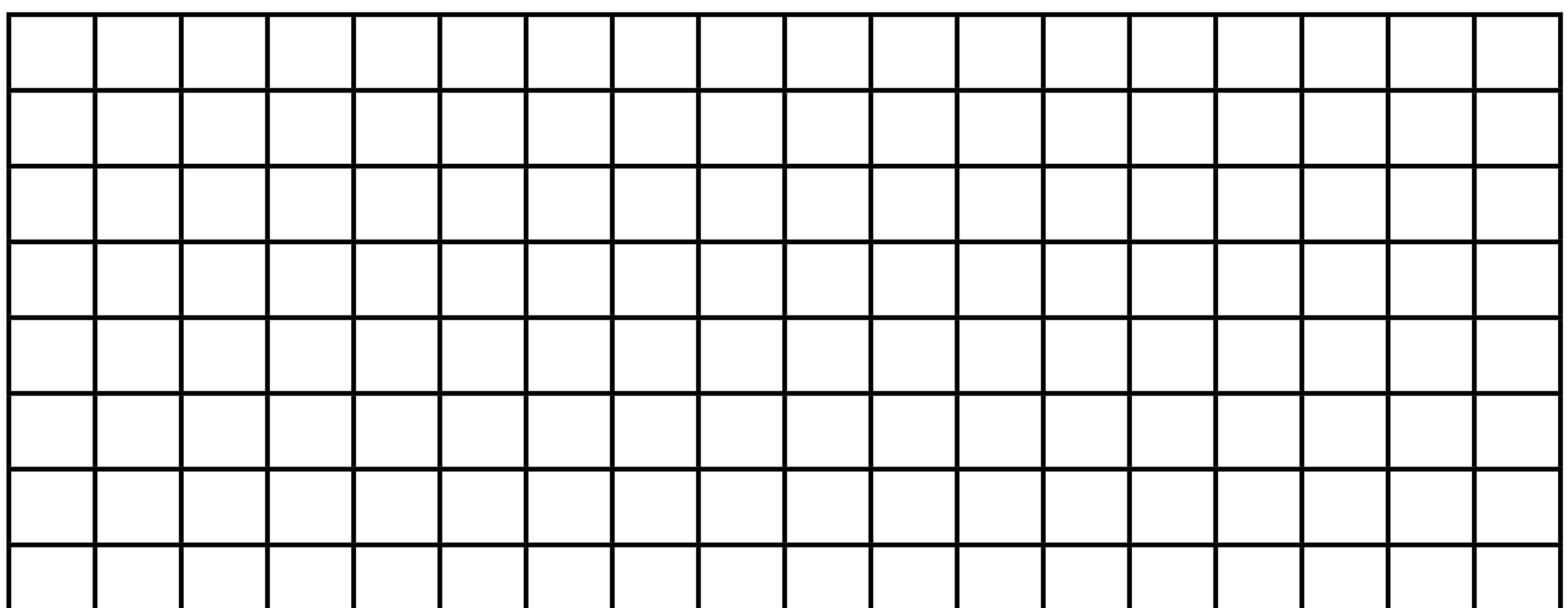
$$\mathsf{Exp} \times \cancel{\widehat{Env}} \times \widehat{Addr}$$

$$\widehat{TID}$$



$$\widehat{Clo} + \widehat{Kont}$$

$$\widehat{Addr}$$



$$\mathsf{Exp} \times \cancel{\widehat{Env}} \times \cancel{\widehat{Addr}} \quad \mathsf{Exp}$$

$$\widehat{TID}$$

$$\widehat{Clo} + \widehat{Kont}$$

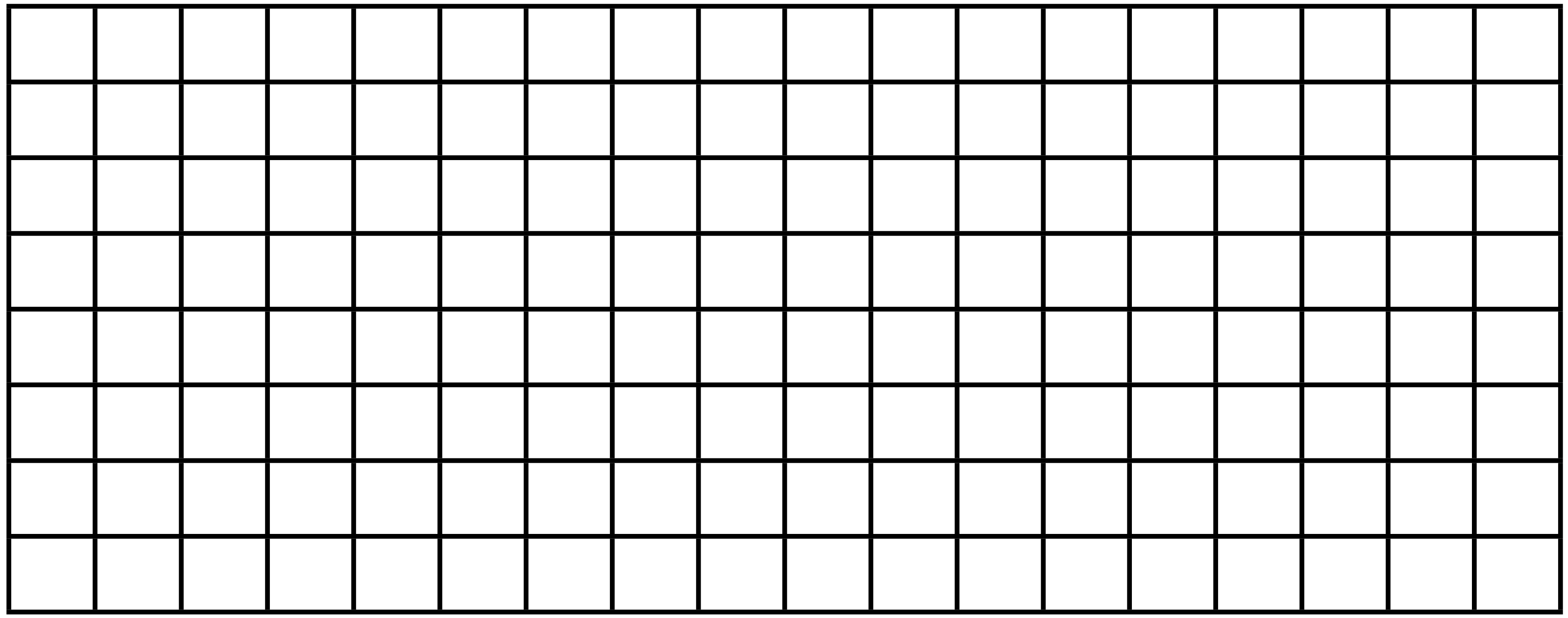
$$\mathsf{Exp}$$

$$\cancel{\widehat{Addr}}$$

$\mathsf{Exp} \times \cancel{\widehat{Env}} \times \cancel{\widehat{Addr}} \quad \mathsf{Exp}$

Exp

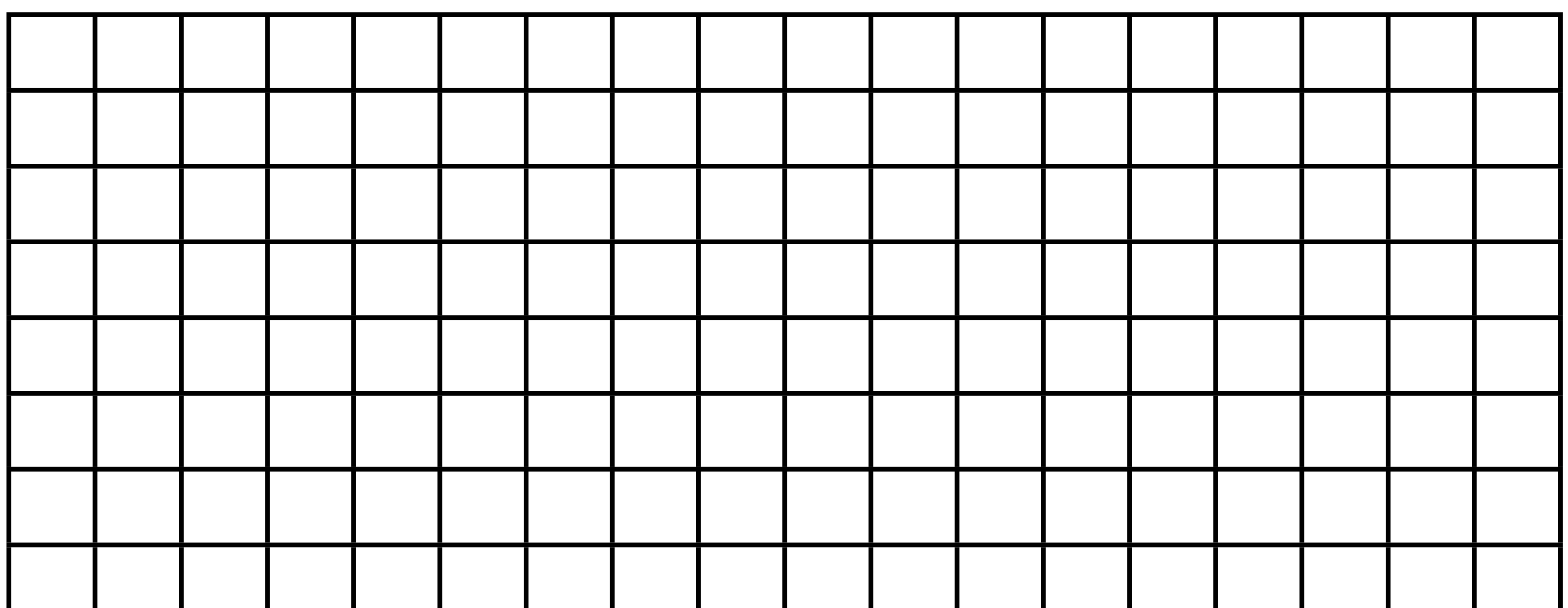
$\cancel{\widehat{TID}}$



$\widehat{Clo} + \widehat{Kont}$

Exp

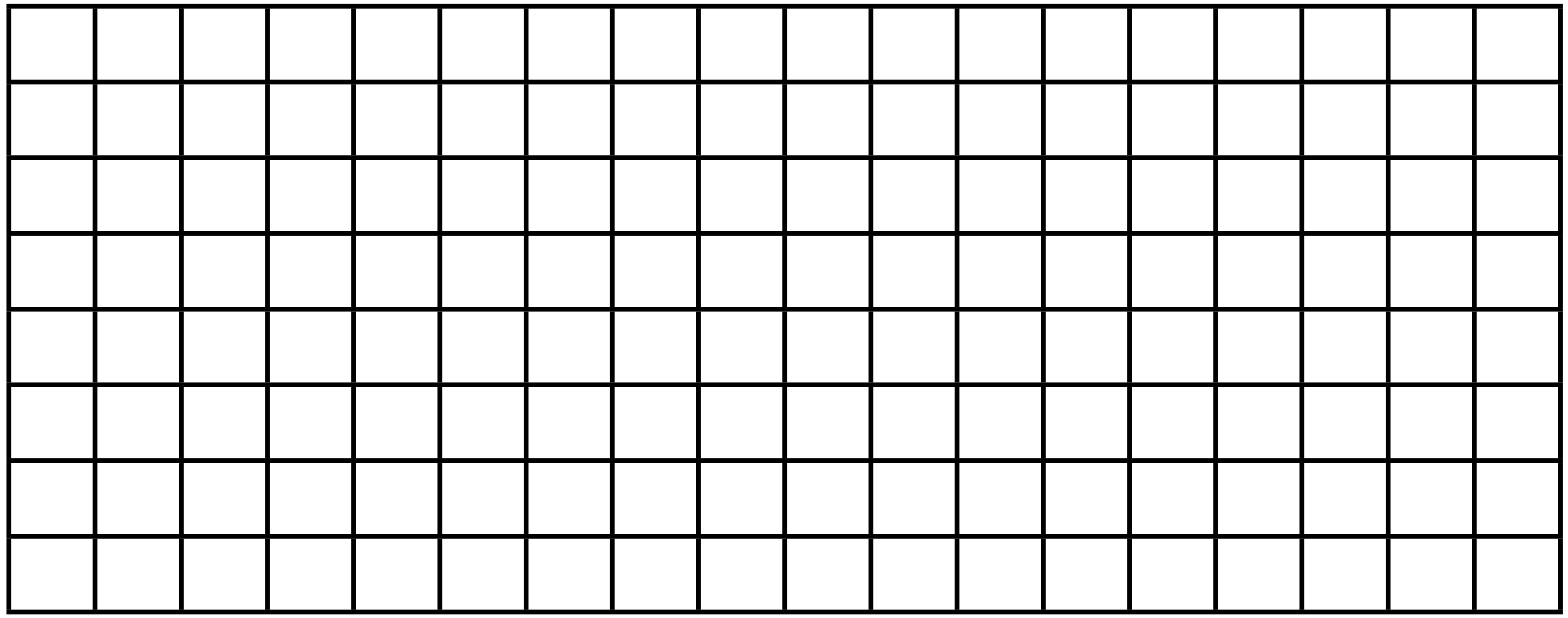
$\cancel{\widehat{Addr}}$



$\mathsf{Exp} \times \cancel{\widehat{Env}} \times \cancel{\widehat{Addr}} \quad \mathsf{Exp}$

Exp

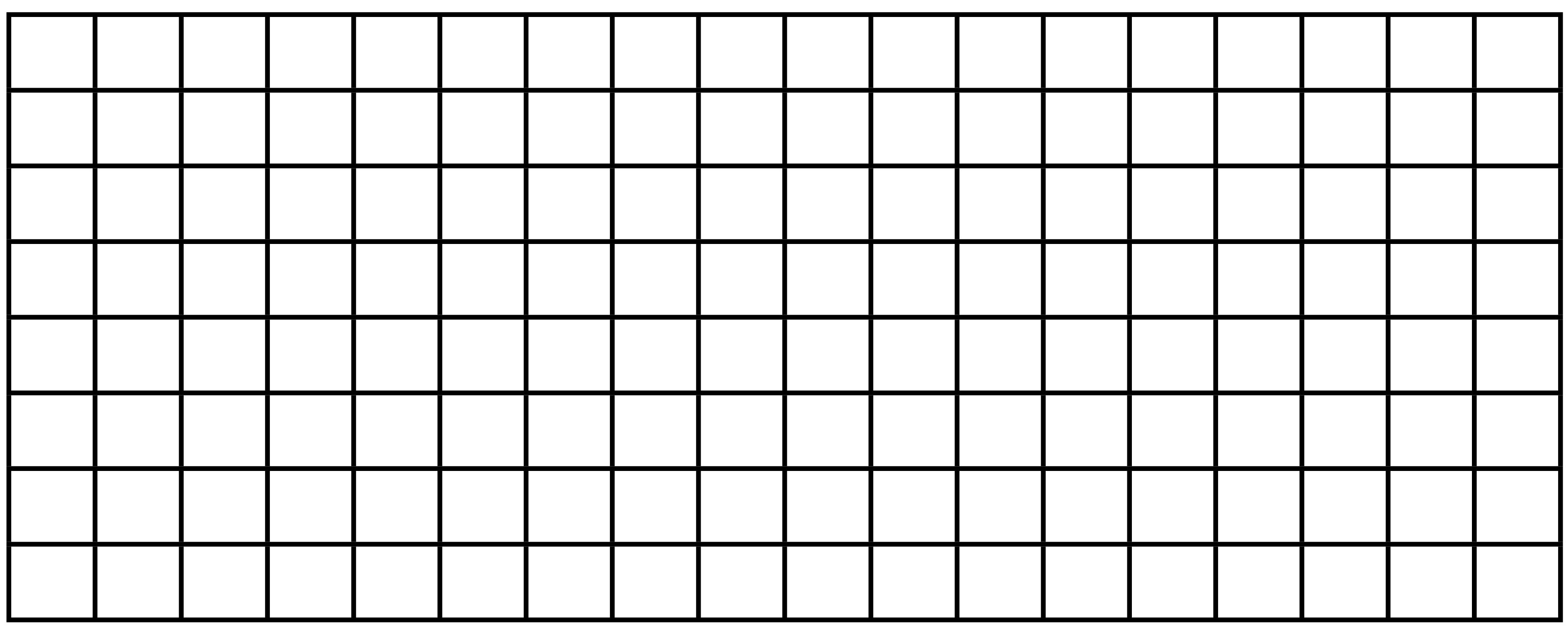
$\cancel{\widehat{TID}}$



$\mathsf{Lam} \quad \cancel{\widehat{Clo}} + \widehat{Kont}$

Exp

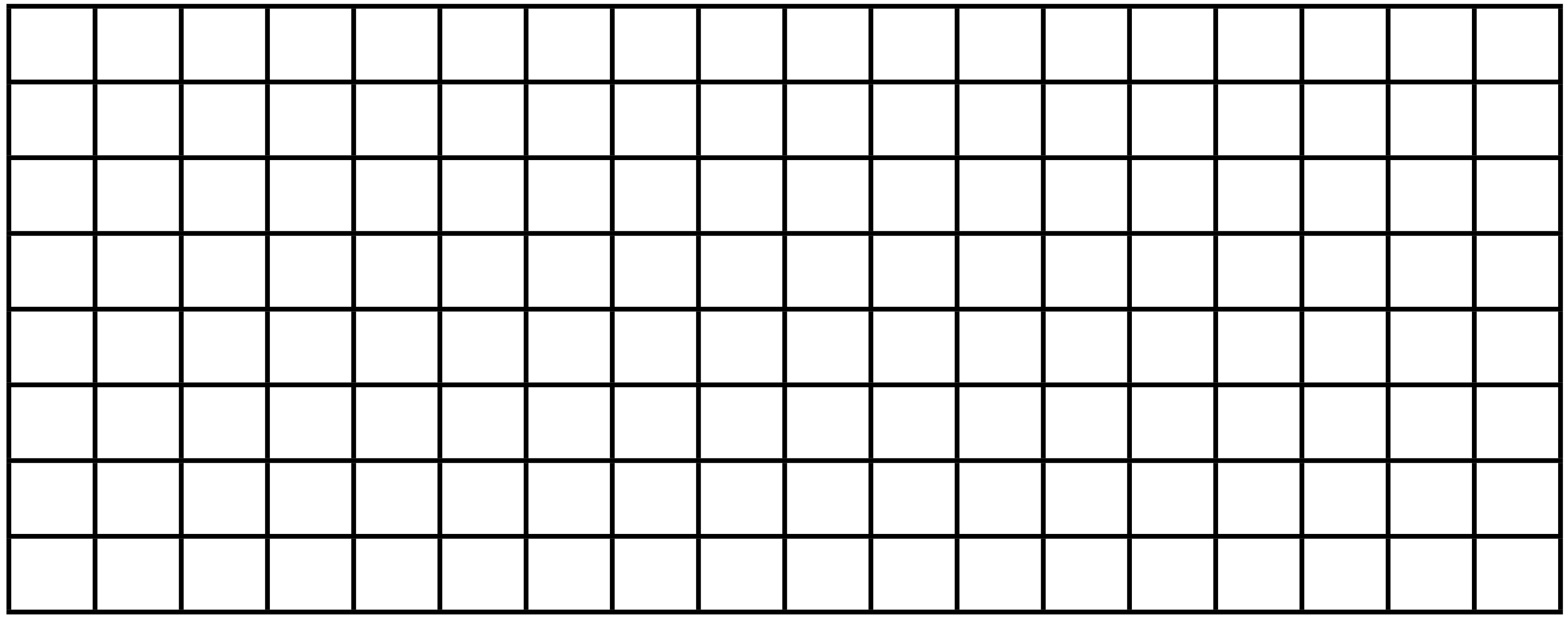
$\cancel{\widehat{Addr}}$



$\mathsf{Exp} \times \cancel{\widehat{Env}} \times \cancel{\widehat{Addr}}$ $\quad \mathsf{Exp}$

Exp

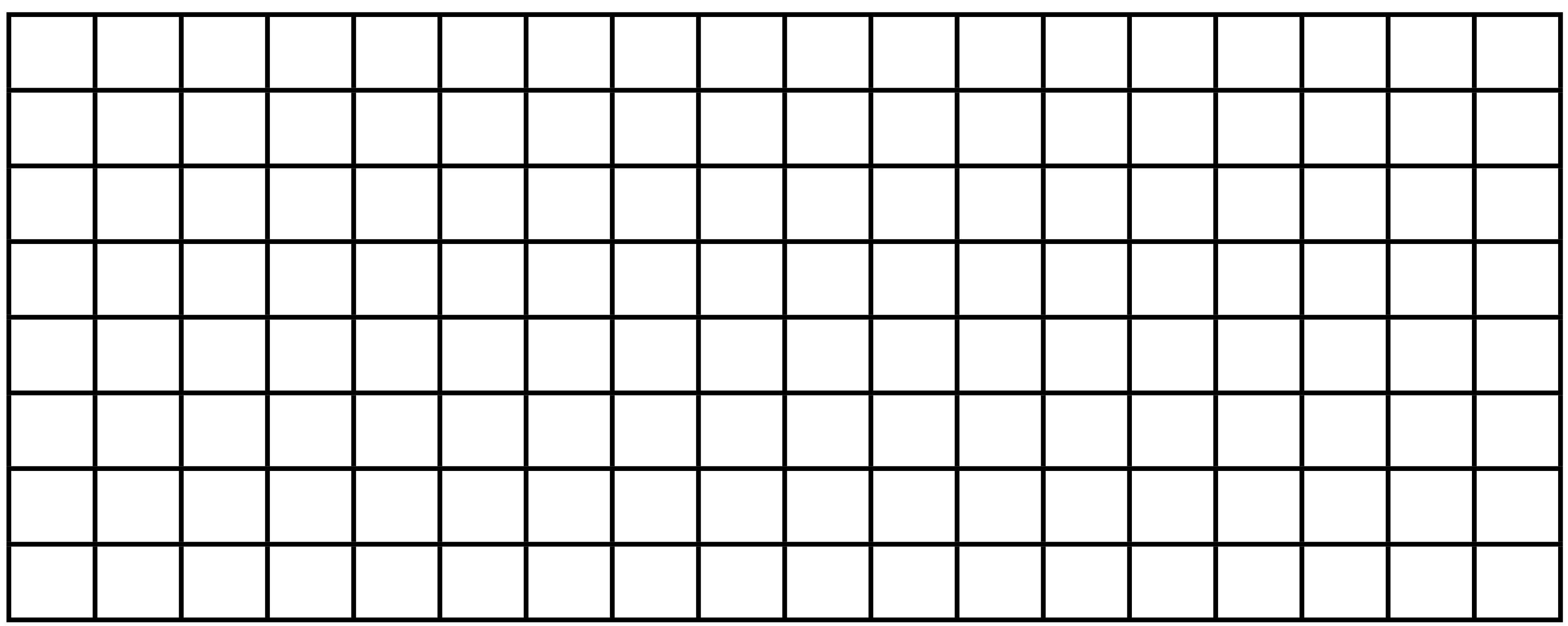
$\cancel{\widehat{TID}}$



Lam $\quad \cancel{\widehat{Clo}} + \cancel{\widehat{Kont}}$ $\quad \mathsf{Exp} \times \cancel{\widehat{Addr}}$ $\quad \mathsf{Exp}$

Exp

$\cancel{\widehat{Addr}}$



$$O(n^3)$$

passes

$$O(n^3)$$

cost per pass

$$O(n^6)$$

for “thread-aware” 0CFA

Related work

- Cousot & Cousot: Abstract interpretation
- Jones; Shivers: CFA for higher-orderness
- Chase et al.: Cardinality for shape analysis
- Jagannathan et al.: Strong-update for CFA
- Yahav: Shape analysis with multithreading

Conclusion

- Abstractable semantics: CESK to P(CEK*)S
- MHP = abstract semantics + thread shape
- CFA = re-abstraction of MHP semantics

Grazie!