Linear-$\lambda\mu$ is CP (more or less)

Jennifer Paykin and Steve Zdancewic
University of Pennsylvania

Wadlerfest
April 11, 2016

Linear-$\lambda\mu$ is CP (more or less)

Linear-$\lambda\mu$ is CP (more or less)

1. What is CP?

π-calculus

Linear-$\lambda\mu$ is CP
(more or less)

1. What is CP?

Linear-$\lambda\mu$ is CP

π-calculus

(more or less)

1. What is CP?

2. What is linear-$\lambda\mu$?

λ-calculus

π-calculus

Linear-$\lambda\mu$ is CP
(more or less)

1. What is CP?

2. What is linear-$\lambda\mu$?

λ-calculus

π-calculus

Linear-$\lambda\mu$ is CP (more or less)

1. What is CP?
2. What is linear-$\lambda\mu$?
3. Is linear-$\lambda\mu$ = CP?

CP & Session Types

Wadler 2012

$$P \vdash x_1 : X_1, \ldots, x_n : X_n$$

CP & Session Types

Wadler
2012

$$\underline{P} \vdash x_1 : X_1, \ldots, x_n : X_n$$

process

CP & Session Types

Wadler
2012

$$P \vdash x_1 : X_1, \ldots, x_n : X_n$$

process **channels**

CP & Session Types

Wadler
2012

$$P \vdash x_1 : X_1, \ldots, x_n : X_n$$

process **channels** **session protocols**

A Session-typed Store

Caires and Pfenning 2010

Store

$$(\mathsf{Prod} \multimap \mathsf{CC} \multimap 1 \oplus 1)$$

$$\& \, (\mathsf{Prod} \multimap \mathsf{Cost} \otimes 1)$$

A Session-typed Store

Caires and Pfenning 2010

`Store`

input choice of buy

$$(\mathsf{Prod} \multimap \mathsf{CC} \multimap 1 \oplus 1)$$

or quote

$$\& \; (\mathsf{Prod} \multimap \mathsf{Cost} \otimes 1)$$

A Session-typed Store

Caires and Pfenning 2010

`Store`

input choice of buy

$$(\text{Prod} \multimap \text{CC} \multimap \mathbf{1} \oplus \mathbf{1})$$

input product

or quote

$$\& \, (\text{Prod} \multimap \text{Cost} \otimes \mathbf{1})$$

A Session-typed Store

Caires and Pfenning 2010

`Store`

input choice of buy

$$(\mathrm{Prod} \multimap \mathrm{CC} \multimap 1 \oplus 1)$$

input product

input credit card

or quote

$$\& \, (\mathrm{Prod} \multimap \mathrm{Cost} \otimes 1)$$

A Session-typed Store

Caires and Pfenning 2010

`Store`

input choice of buy

$$(\mathsf{Prod} \multimap \mathsf{CC} \multimap \mathbf{1} \oplus \mathbf{1})$$

input product — **input credit card** — **output choice: succeeds or fails**

or quote

$$\& \; (\mathsf{Prod} \multimap \mathsf{Cost} \otimes \mathbf{1})$$

A Session-typed Store

Caires and Pfenning 2010

`Store`

input choice of buy

$$(\mathsf{Prod} \multimap \mathsf{CC} \multimap 1 \oplus 1)$$

input product

input credit card

output choice: succeeds or fails

or quote

$$\& \, (\mathsf{Prod} \multimap \mathsf{Cost} \otimes 1)$$

input product

A Session-typed Store

Caires and Pfenning 2010

`Store`

input choice of buy

$$(\mathrm{Prod} \multimap \mathrm{CC} \multimap 1 \oplus 1)$$

input product

input credit card

output choice: succeeds or fails

or quote

$$\& (\mathrm{Prod} \multimap \mathrm{Cost} \otimes 1)$$

input product

output cost, then terminates

A Session-typed Customer

`Customer`

$$(\mathsf{Prod} \otimes \mathsf{CC} \otimes (\bot \& \bot))$$

$$\oplus (\mathsf{Prod} \otimes (\mathsf{Cost} \multimap \bot))$$

A Session-typed Customer

`Customer`

output choice of buy

$$(\mathsf{Prod} \otimes \mathsf{CC} \otimes (\bot \,\&\, \bot))$$

or quote

$$\oplus\, (\mathsf{Prod} \otimes (\mathsf{Cost} \multimap \bot))$$

A Session-typed Customer

`Customer`

output choice of buy

$$(\mathsf{Prod} \otimes \mathsf{CC} \otimes (\bot \& \bot)$$

output product

or quote

$$\oplus (\mathsf{Prod} \otimes (\mathsf{Cost} \multimap \bot))$$

A Session-typed Customer

`Customer`

output choice of buy

$$(\mathsf{Prod} \otimes \mathsf{CC} \otimes (\bot \,\&\, \bot))$$

output product

output credit card

or quote

$$\oplus\, (\mathsf{Prod} \otimes (\mathsf{Cost} \multimap \bot))$$

A Session-typed Customer

`Customer`

output choice of buy

$$(\mathsf{Prod} \otimes \mathsf{CC} \otimes (\bot \& \bot))$$

output product

output credit card

input choice: succeeds or fails

or quote

$$\oplus\, (\mathsf{Prod} \otimes (\mathsf{Cost} \multimap \bot))$$

A Session-typed Customer

`Customer`

output choice of buy

$$(\mathsf{Prod} \otimes \mathsf{CC} \otimes (\bot \,\&\, \bot))$$

output product — **output credit card** — **input choice: succeeds or fails**

or quote

$$\oplus\, (\mathsf{Prod} \otimes (\mathsf{Cost} \multimap \bot)))$$

output product

A Session-typed Customer

`Customer`

output choice of buy

$$(\mathsf{Prod} \otimes \mathsf{CC} \otimes (\bot \,\&\, \bot)$$

output product

output credit card

input choice: succeeds or fails

or quote

$$\oplus\, (\mathsf{Prod} \otimes (\mathsf{Cost} \multimap \bot))$$

output product

input cost, then continue

A Session-typed Customer

$$\boxed{\texttt{Customer}} = \boxed{(\texttt{Store})^{\perp}}$$

output choice of buy

$$(\mathsf{Prod} \otimes \mathsf{CC} \otimes (\perp \& \perp))$$

output product

output credit card

input choice: succeeds or fails

or quote

$$\oplus\, (\mathsf{Prod} \otimes (\mathsf{Cost} \multimap \perp))$$

output product

input cost, then continue

Session-Typed Processes

processes = linear proofs

Session-Typed Processes

$$s[\texttt{inr}].s[p].(\mathsf{bread}_p \mid s(c).\qquad\qquad\qquad)$$

$$\vdash s : \boxed{\texttt{Customer}}$$

processes = linear proofs

Session-Typed Processes

$$s[\mathtt{inr}].s[p].(\mathsf{bread}_p \mid s(c). \qquad\qquad)$$

$$\vdash s : \boxed{\mathtt{Customer}}$$
$$= (\mathsf{Prod} \otimes \mathsf{CC} \otimes (\bot \& \bot)$$
$$\oplus (\mathsf{Prod} \otimes (\mathsf{Cost} \multimap \bot)))$$

processes = linear proofs

Session-Typed Processes

$$s[\mathtt{inr}].s[p].(\mathsf{bread}_p \mid s(c). \qquad)$$

$$\vdash s : \boxed{\mathtt{Customer}}$$
$$= (\mathsf{Prod} \otimes \mathsf{CC} \otimes (\bot \& \bot)$$
$$\oplus (\mathsf{Prod} \otimes (\mathsf{Cost} \multimap \bot)))$$

processes = linear proofs

Session-Typed Processes

$$s[\mathtt{inr}].s[p].(\mathsf{bread}_p \mid s(c). \qquad\qquad)$$

$$\vdash s : \boxed{\mathtt{Customer}}$$
$$= (\mathsf{Prod} \otimes \mathsf{CC} \otimes (\bot \,\&\, \bot)$$
$$\oplus (\mathsf{Prod} \otimes (\mathsf{Cost} \multimap \bot))$$

processes = linear proofs

Session-Typed Processes

$$s[\texttt{inr}].s[p].(\mathsf{bread}_p \mid s(c). \qquad\qquad)$$

$$\vdash s : \boxed{\texttt{Customer}}$$

$$= (\mathsf{Prod} \otimes \mathsf{CC} \otimes (\bot \,\&\, \bot)$$

$$\oplus (\mathsf{Prod} \otimes (\mathsf{Cost} \multimap \bot)))$$

processes = linear proofs

Session-Typed Processes

$$s[\texttt{inr}].s[p].(\mathsf{bread}_p \mid s(c).\quad c \leftrightarrow x \quad)$$

$$\vdash s : \boxed{\texttt{Customer}},\; x : \mathsf{Cost}$$

$$= (\mathsf{Prod} \otimes \mathsf{CC} \otimes (\bot \& \bot)$$
$$\oplus (\mathsf{Prod} \otimes (\mathsf{Cost} \multimap \bot)))$$

processes = linear proofs

Session-Typed Processes

$$s[\texttt{inr}].s[p].(\mathsf{bread}_p \mid s(c).\quad c \leftrightarrow x \quad)$$

$$\vdash s : \boxed{\texttt{Customer}}, x : \mathsf{Cost}$$

processes = linear proofs

Session-Typed Processes

$$s[\texttt{inr}].s[p].(\mathsf{bread}_p \mid s(c).\quad c \leftrightarrow x \quad)$$

$$\vdash s : \boxed{(\texttt{Store})^{\perp}}, x : \mathsf{Cost}$$

processes = linear proofs

Session-Typed Processes

$$s : \boxed{\texttt{Store}} \vdash$$

$$\underline{s[\texttt{inr}]}.\underline{s[p].(\mathsf{bread}_p} \mid \underline{s(c)}.\quad c \leftrightarrow x \quad)$$

$$\vdash x : \mathsf{Cost}$$

processes = linear proofs

π-DILL

Caires and Pfenning 2010

$$x_1 : X_1, \ldots, x_n : X_n \vdash P :: x : X$$

π-DILL

Caires and Pfenning 2010

$$x_1 : X_1, \ldots, x_n : X_n \vdash P :: x : X$$

input

π-DILL

Caires and Pfenning 2010

$$x_1 : X_1, \ldots, x_n : X_n \vdash P :: x : X$$

input

process

π-DILL

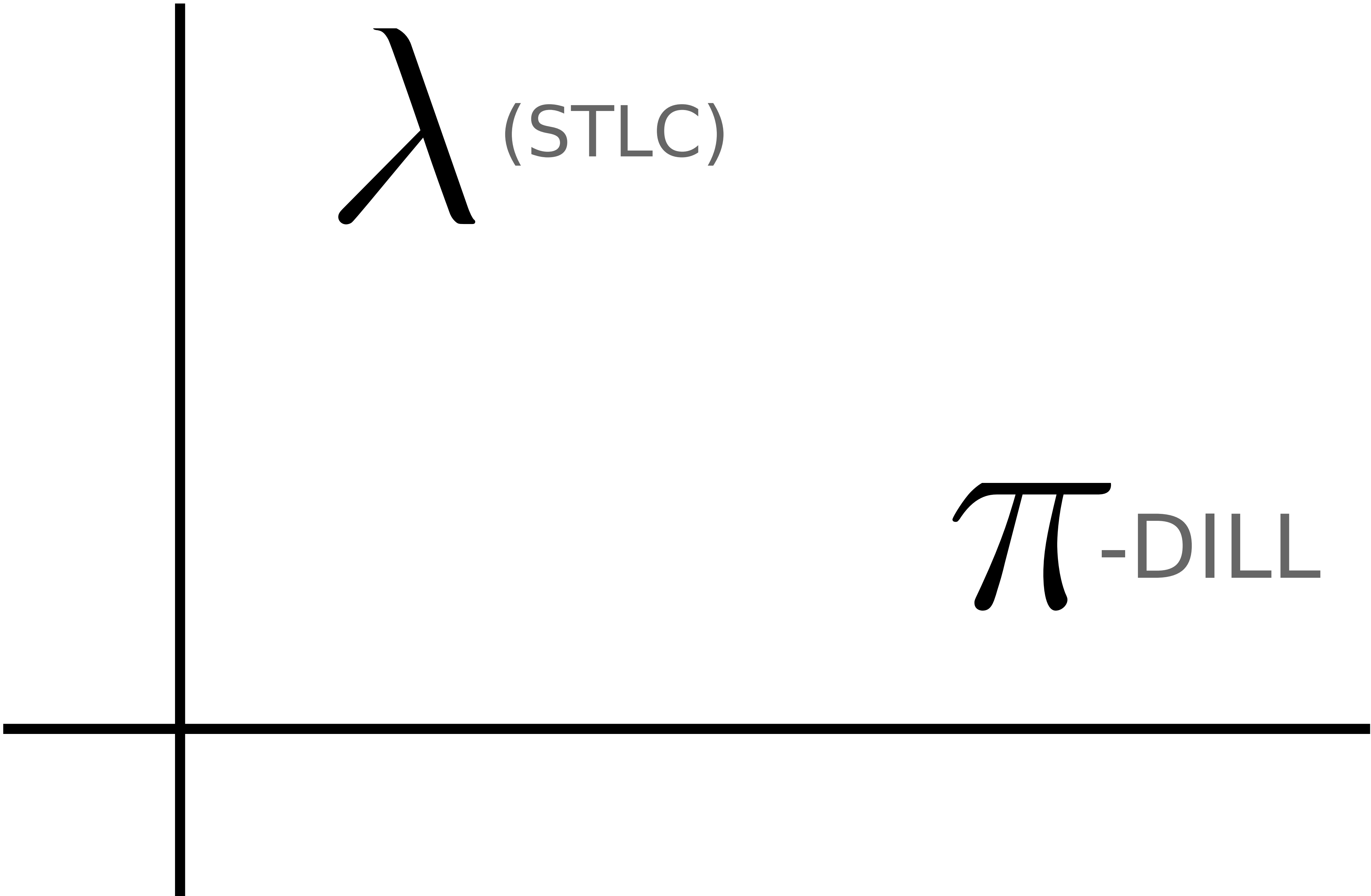
Caires and Pfenning 2010

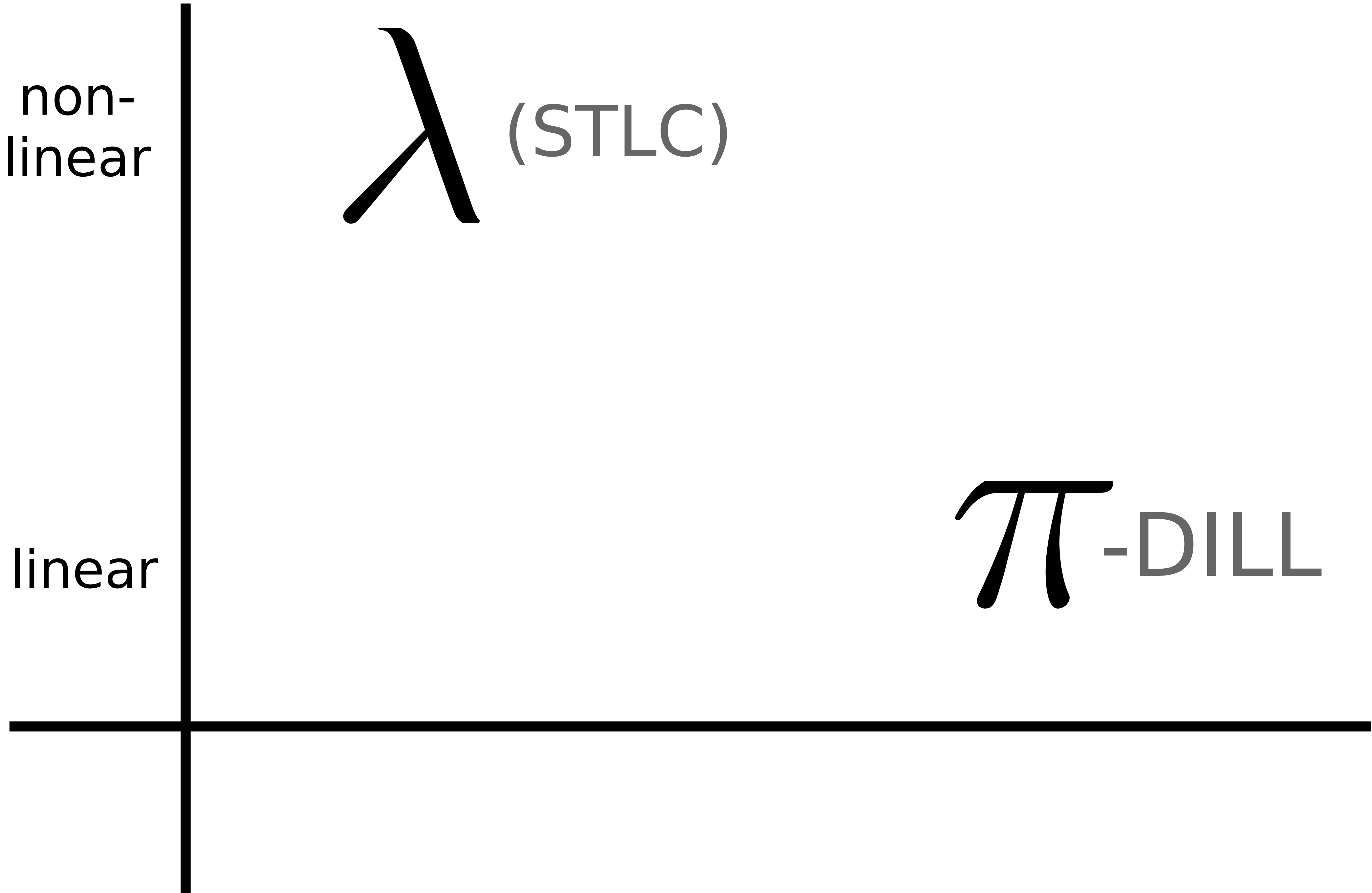
$$x_1 : X_1, \ldots, x_n : X_n \vdash P :: x : X$$

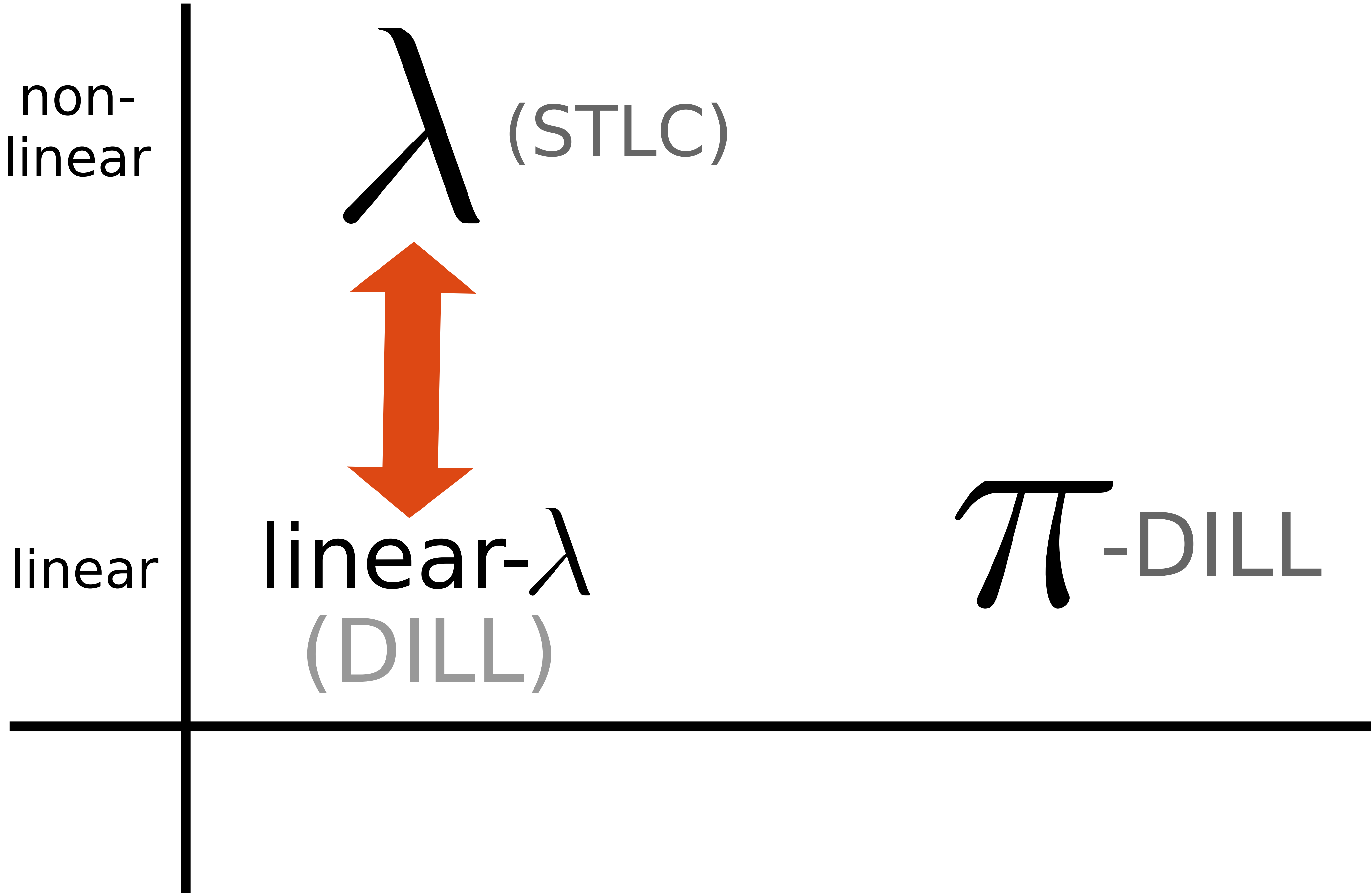
input

process

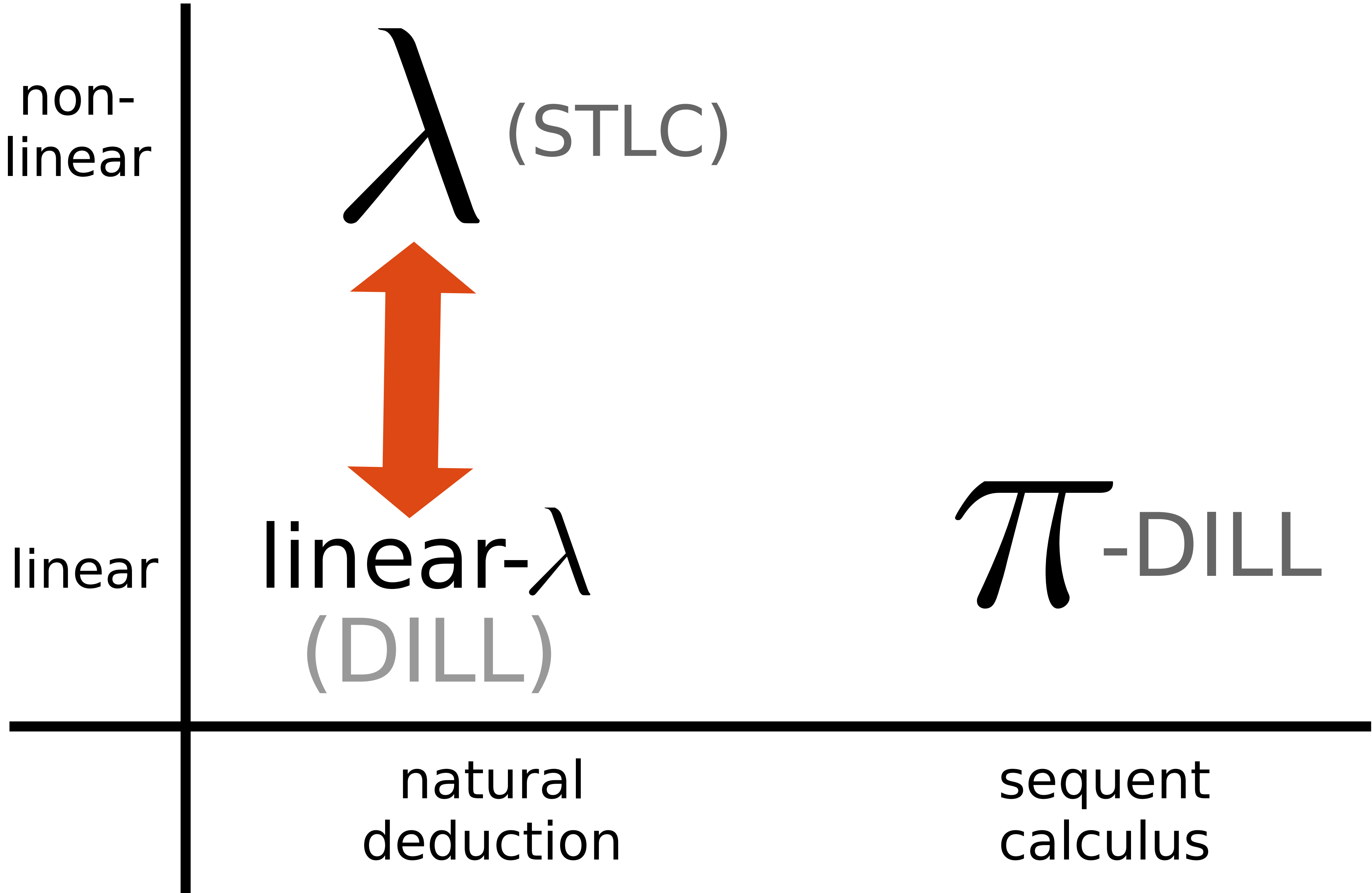
output

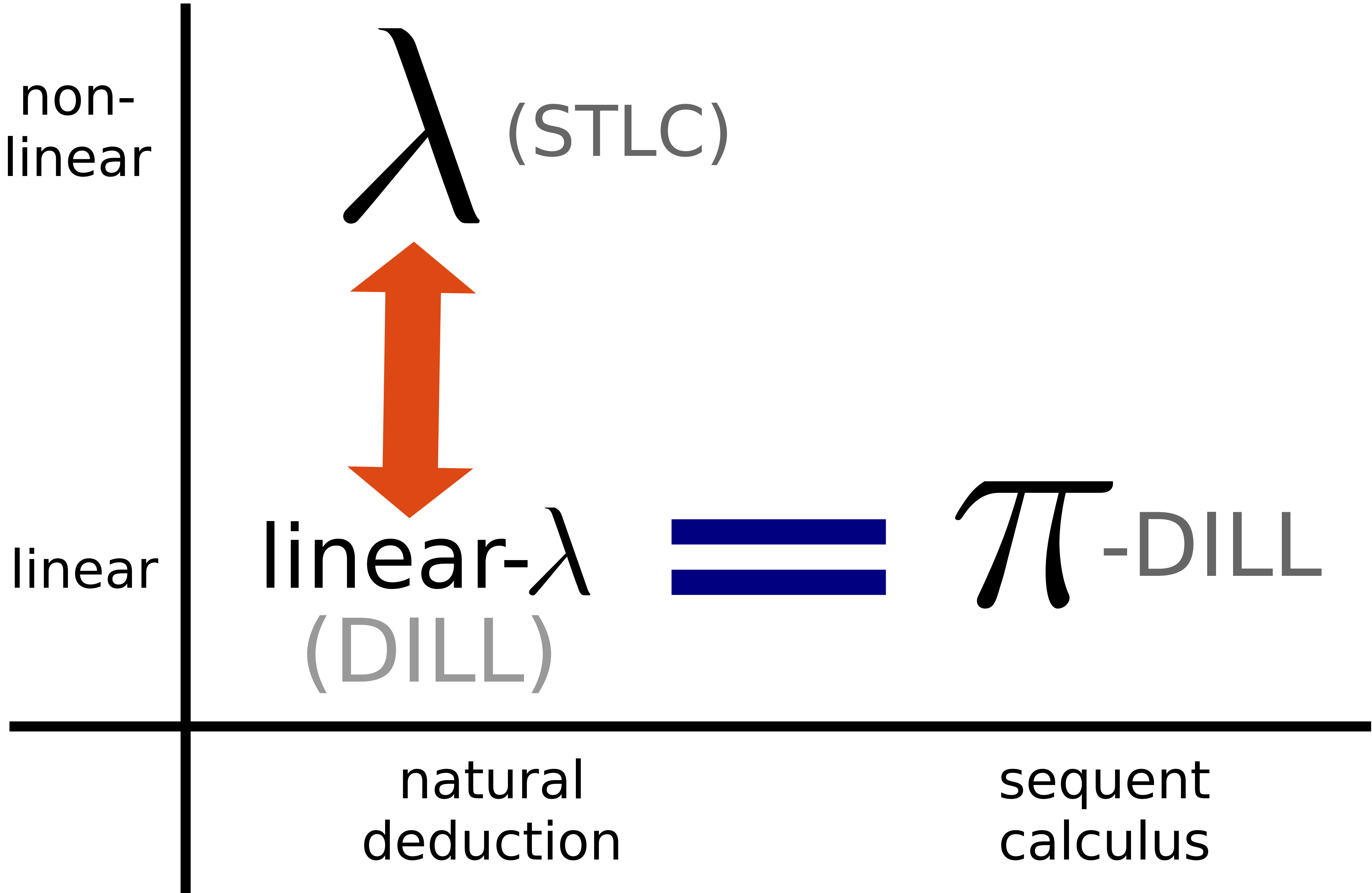

λ (STLC)
π-DILL

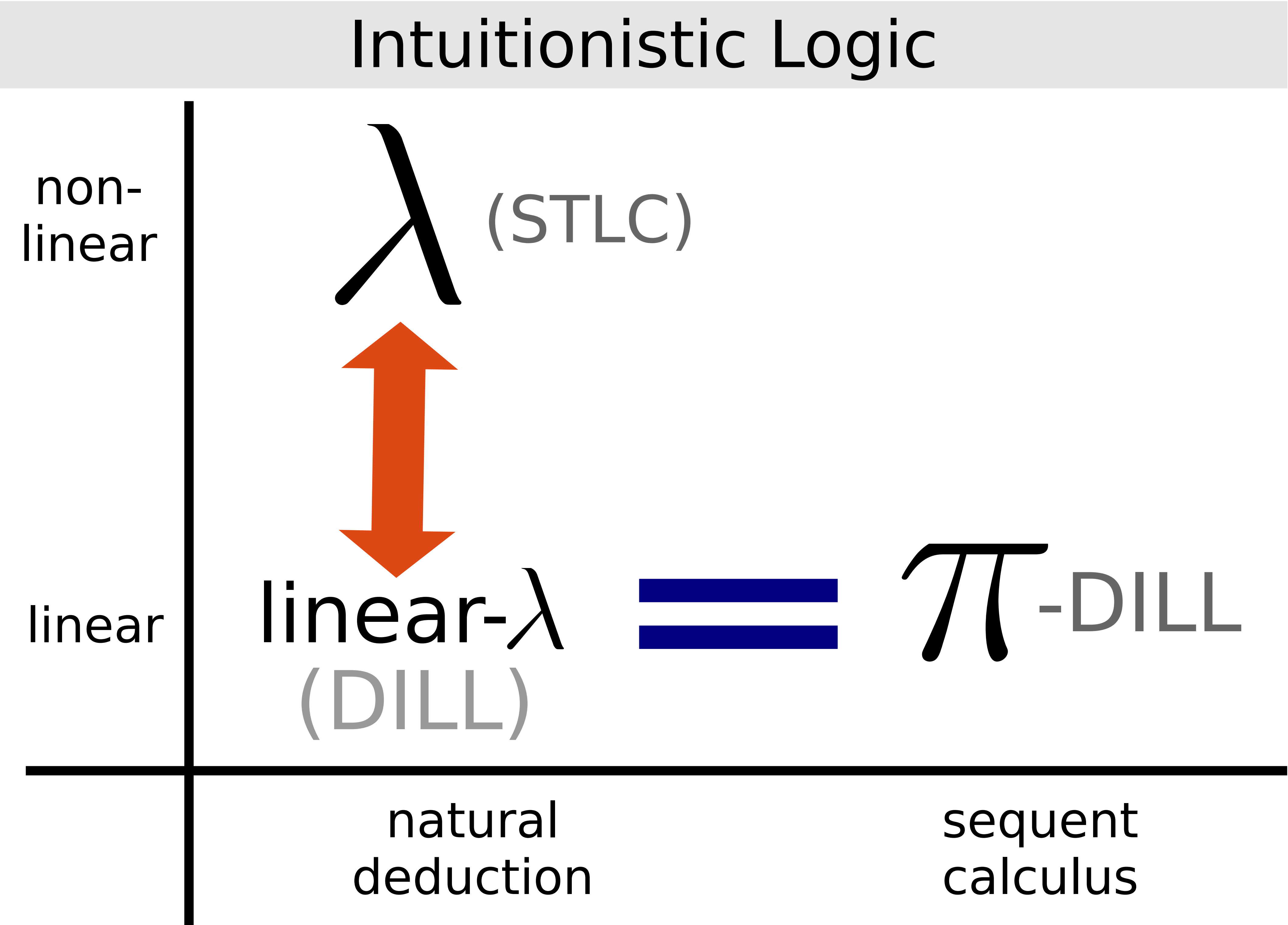

non-
linear
linear
λ (STLC)
π-DILL



non-
linear
λ (STLC)
linear
linear-λ
(DILL)
π-DILL

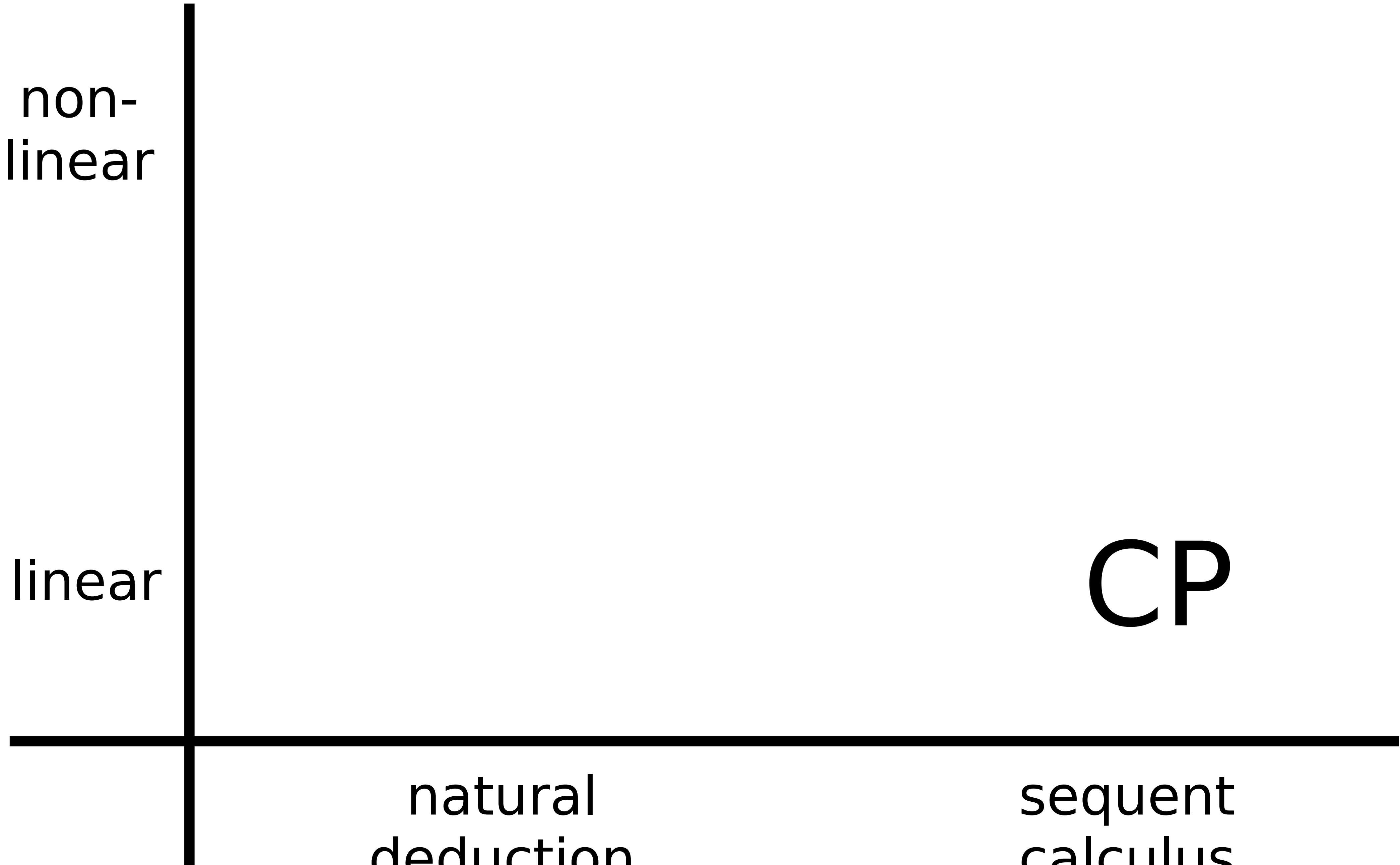

non-
linear
λ (STLC)
linear
linear-λ
(DILL)
π-DILL
natural
deduction
sequent
calculus


non-
linear
λ (STLC)
linear
linear-λ
(DILL)
π-DILL
natural
deduction
sequent
calculus

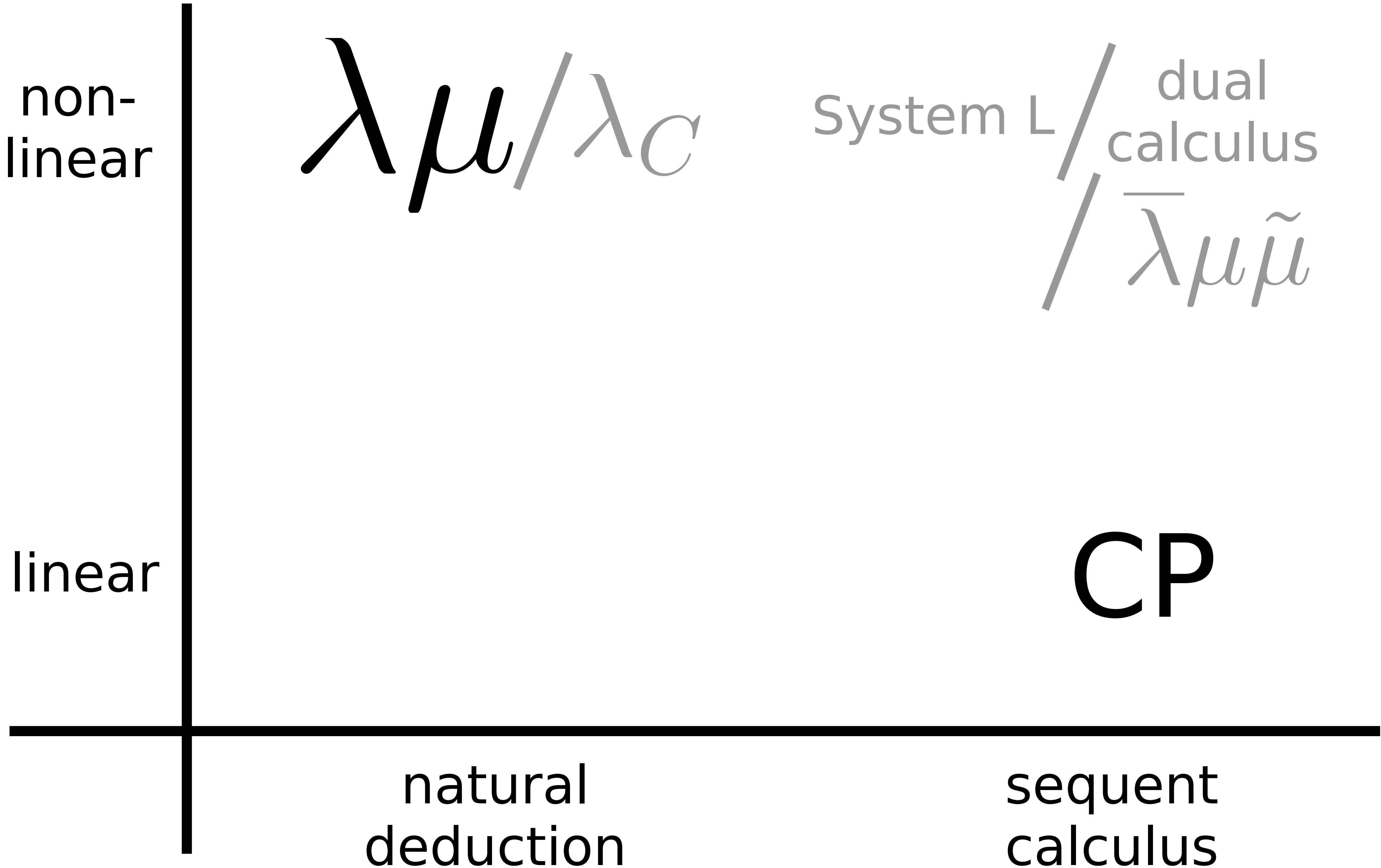


Intuitionistic Logic
non-
linear
λ (STLC)
linear
linear-λ
(DILL)
π-DILL
natural
deduction
sequent
calculus

Classical Logic



Classical Logic



Linear-$\lambda\mu$ is CP (more or less)

1. What is CP?
2. What is linear-$\lambda\mu$?
3. Is linear-$\lambda\mu$ = CP?

$\lambda\mu$

Parigot
1992

$$x_i : A_i \vdash t : A \mid \alpha_j : A'_j$$

$$x_i : A_i \vdash c \mid \alpha_j : A'_j$$

$\lambda\mu$

Parigot 1992

multiple inputs

$$\underline{x_i : A_i} \vdash t : A \mid \alpha_j : A'_j$$

$$\underline{x_i : A_i} \vdash c \mid \alpha_j : A'_j$$

$\lambda\mu$

Parigot 1992

multiple inputs

multiple outputs

$$\underline{x_i : A_i} \vdash t : \underline{A} \mid \underline{\alpha_j : A'_j}$$

$$\underline{x_i : A_i} \vdash c \mid \underline{\alpha_j : A'_j}$$

$\lambda\mu$

Parigot
1992

$\lambda\mu$

Parigot
1992

$$\frac{\vdash t : A \mid}{\vdash [\alpha]t \mid \alpha : A}$$

$\lambda\mu$

Parigot 1992

$$\frac{\vdash t : A \mid}{\vdash [\alpha]t \mid \alpha : A} \qquad \frac{\vdash c \mid \alpha : A}{\vdash \mu\alpha.c : A \mid}$$

$\lambda\mu$

Parigot
1992

$$\frac{\vdash t : A \mid}{\vdash [\alpha]t \mid \alpha : A} \qquad \frac{\vdash c \mid \alpha : A}{\vdash \mu\alpha.c : A \mid}$$

$$A ::= 1 \mid A_1 \times A_2 \mid 0 \mid A_1 + A_2 \mid \neg A$$

$\lambda\mu$

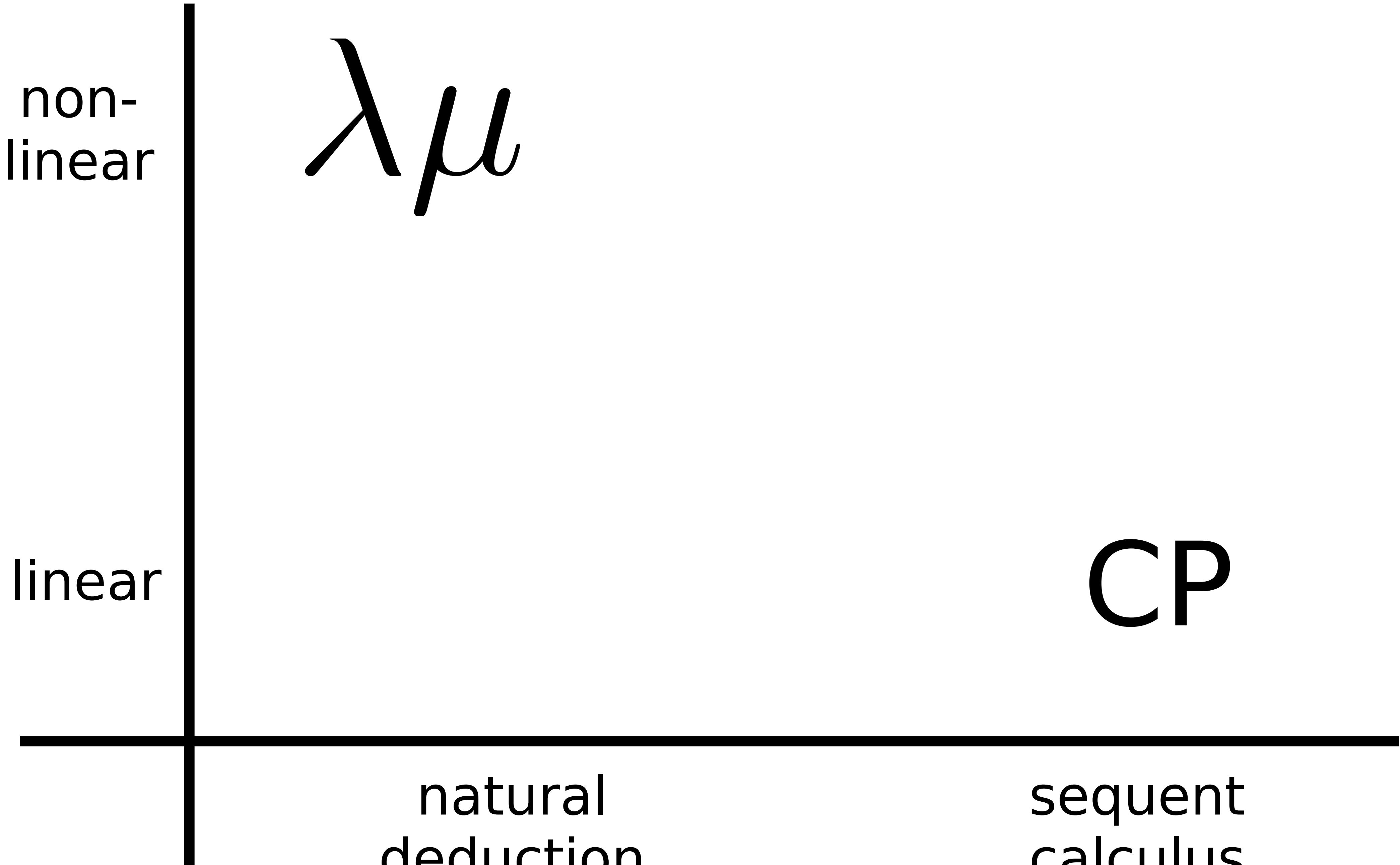
Parigot 1992

$$\frac{\vdash t : A \mid}{\vdash [\alpha]t \mid \alpha : A} \qquad \frac{\vdash c \mid \alpha : A}{\vdash \mu\alpha.c : A \mid}$$

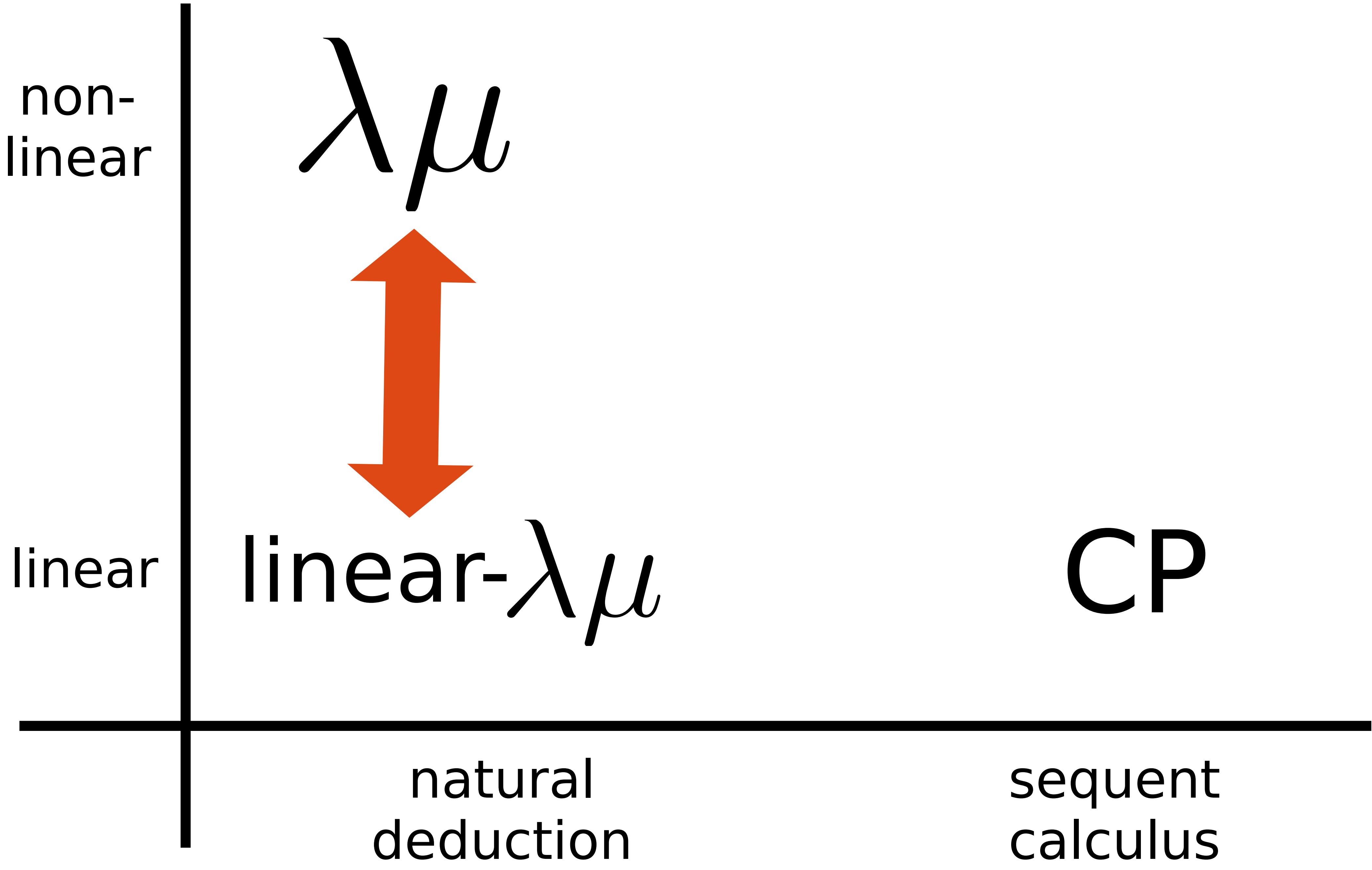
$$A ::= 1 \mid A_1 \times A_2 \mid 0 \mid A_1 + A_2 \mid \neg A$$

$$(A_1 \to A_2 = \neg A_1 + A_2)$$

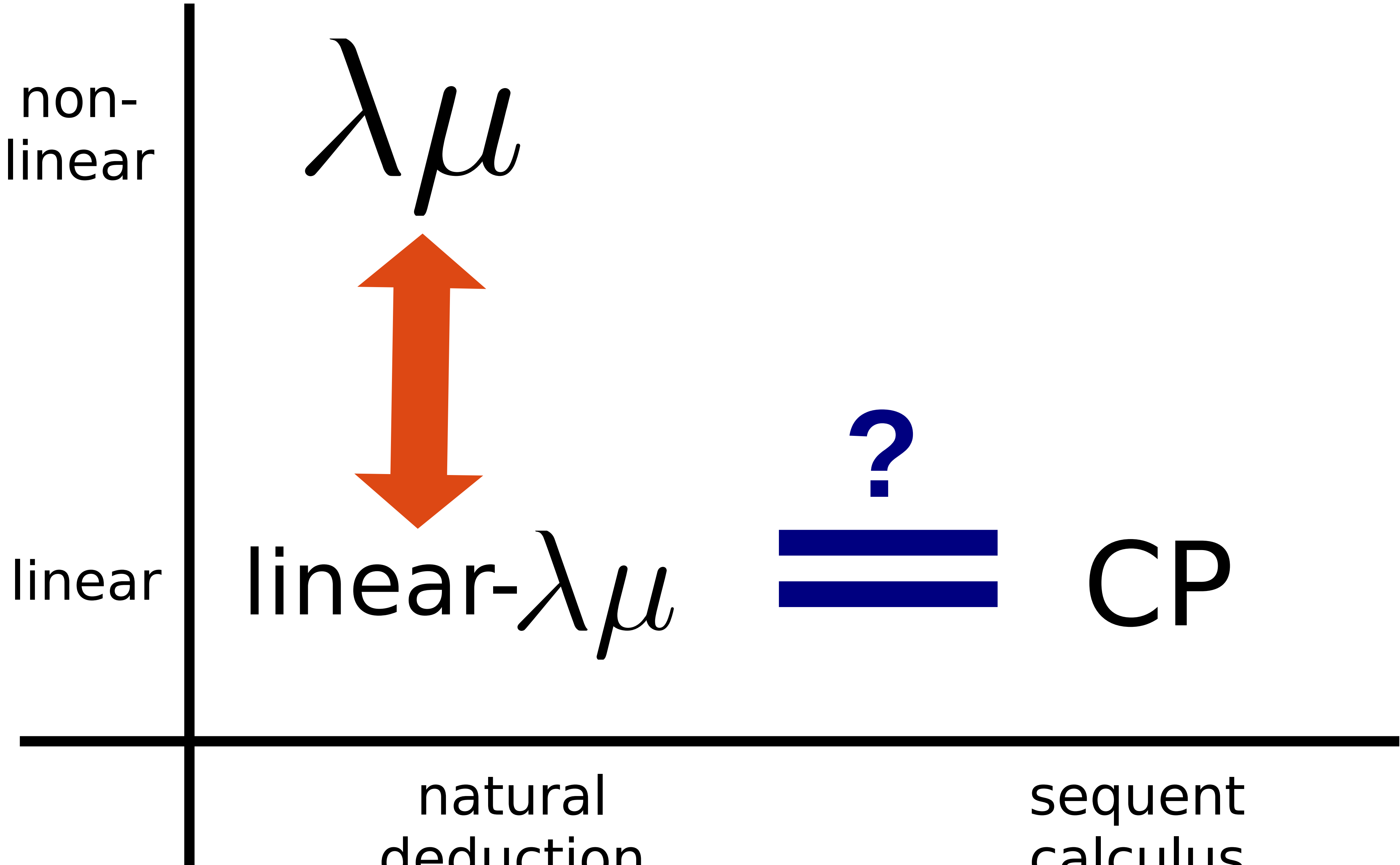
Classical Logic



Classical Logic



Classical Logic



Linear-$\lambda\mu$ is CP (more or less)

1. What is CP?
2. What is linear-$\lambda\mu$?
3. Is linear-$\lambda\mu$ = CP?

linear-$\lambda\mu$ vs CP

$$\Gamma \vdash t : A \mid \Pi \qquad \Gamma \vdash c \mid \Pi$$

$$P \vdash \Omega$$

linear-$\lambda\mu$ vs CP

input

$$\underline{\Gamma} \vdash t : A \mid \Pi \qquad \underline{\Gamma} \vdash c \mid \Pi$$

$$P \vdash \Omega$$

linear-$\lambda\mu$ vs CP

input output

$$\underline{\Gamma} \vdash t : \underline{A} \mid \underline{\Pi} \qquad \underline{\Gamma} \vdash c \mid \underline{\Pi}$$

$$P \vdash \Omega$$

linear-$\lambda\mu$ vs CP

input output

$$\underline{\Gamma} \vdash t : \underline{A} \mid \underline{\Pi} \qquad \underline{\Gamma} \vdash c \mid \underline{\Pi}$$

$$P \vdash \underline{\underline{\Omega}}$$

linear-$\lambda\mu$ vs CP

input output

$$\underline{\Gamma} \vdash t : \underline{A} \mid \underline{\Pi} \qquad \underline{\Gamma} \vdash c \mid \underline{\Pi}$$

$$A ::= 1 \mid A_1 \otimes A_2 \mid 0 \mid A_1 \oplus A_2 \mid \neg A$$

$$P \vdash \underline{\underline{\Omega}}$$

$$X ::= 1 \mid X_1 \otimes X_2 \mid 0 \mid X_1 \oplus X_2$$
$$\mid \bot \mid X_1 \parr X_2 \mid \top \mid X_1 \& X_2$$

dualizing linear-$\lambda\mu$

$$\Gamma \vdash c \mid \Pi$$

dualizing linear-$\lambda\mu$

$$\Gamma \vdash c \mid \Pi \quad \Longrightarrow \quad c^* \vdash \Gamma^\perp ; \Pi$$

dualizing linear-$\lambda\mu$

$$\Gamma \vdash c \mid \Pi \quad \Rightarrow \quad c^* \vdash \Gamma^\perp ; \Pi$$

$$A ::= 1 \mid A_1 \otimes A_2 \mid 0 \mid A_1 \oplus A_2 \mid \neg A$$

dualizing linear-$\lambda\mu$

$$\Gamma \vdash c \mid \Pi \quad \Rightarrow \quad c^* \vdash \underline{\Gamma^\perp}; \underline{\Pi}$$

input output

$$A^+ ::= 1 \mid A_1^+ \otimes A_2^+ \mid 0 \mid A_1^+ \oplus A_2^+ \mid \neg A^+$$

$$A^- ::= \bot \mid A_1^- ⅋ A_2^- \mid \top \mid A_1^- \& A_2^- \mid \neg A^-$$

dualizing linear-$\lambda\mu$

$$c^* \vdash \underline{\Gamma}^{\perp} ; \underline{\Pi}$$

input output

$$A^+ ::= 1 \mid A_1^+ \otimes A_2^+ \mid 0 \mid A_1^+ \oplus A_2^+ \mid \neg A^+$$

$$A^- ::= \perp \mid A_1^- \parr A_2^- \mid \top \mid A_1^- \& A_2^- \mid \neg A^-$$

$$P \vdash \Omega$$ input/output

$$X ::= 1 \mid X_1 \otimes X_2 \mid 0 \mid X_1 \oplus X_2 \mid \perp \mid X_1 \parr X_2 \mid \top \mid X_1 \& X_2$$

dualizing linear-$\lambda\mu$

$$c^* \vdash \underline{\Gamma^\perp}; \underline{\Pi}$$

input output

$$A^+ ::= 1 \mid A_1^+ \otimes A_2^+ \mid 0 \mid A_1^+ \oplus A_2^+ \mid \neg A^+$$

$$A^- ::= \perp \mid A_1^- ⅋ A_2^- \mid \top \mid A_1^- \& A_2^- \mid \neg A^-$$

$$P \vdash \Omega$$

input/output

$$X ::= 1 \mid X_1 \otimes X_2 \mid 0 \mid X_1 \oplus X_2$$
$$\mid \perp \mid X_1 ⅋ X_2 \mid \top \mid X_1 \& X_2$$

negation: switching input and output

$$c^* \vdash \underline{\Gamma^\perp}; \underline{\Pi}$$

input output

negation: switching input and output

$$c^* \vdash \underline{\Gamma^\perp}; \underline{\Pi}$$

input output

input

$$\frac{\Gamma, \underline{A} \vdash \Pi}{\Gamma \vdash \underline{\neg A}, \Pi}$$

output

negation: switching input and output

$$c^* \vdash \underline{\Gamma^\perp}; \underline{\Pi}$$

input output

input

$$\frac{\Gamma, \underline{A} \vdash \Pi}{\Gamma \vdash \underline{\neg A}, \Pi}$$

output

$$\frac{\vdash \Gamma^\perp, (A^+)^\perp; \Pi}{\vdash \Gamma^\perp; \neg A^+, \Pi}$$

shifts: switching input and output

Mellies and Tabareau 2010

$$c^* \vdash \underline{\Gamma^\perp}; \underline{\Pi}$$

input output

shifts: switching input and output

Mellies and Tabareau 2010

$$c^* \vdash \underline{\Gamma^\perp}; \underline{\Pi}$$

input output

input

$$\dfrac{\dfrac{\Gamma, \underline{A} \vdash \Pi}{\vdash \Gamma^\perp, \underline{A}^\perp ; \Pi}}{\vdash \Gamma^\perp ; \underline{\downarrow A}^\perp, \Pi}$$

output

shifts: switching input and output

Mellies and Tabareau 2010

$$c^* \vdash \underline{\Gamma^\perp}; \underline{\Pi}$$

input output

input

$$\dfrac{\dfrac{\Gamma, \underline{A} \vdash \Pi}{\vdash \Gamma^\perp, \underline{A^\perp}; \Pi}}{\vdash \Gamma^\perp; \underline{\downarrow} A^\perp, \Pi}$$

output

input

$$\dfrac{\vdash \Gamma^\perp, \underline{A^-}; \Pi}{\vdash \Gamma^\perp; \underline{\downarrow A^-}, \Pi}$$

output

shifts: switching input and output

Mellies and Tabareau 2010

$$c^* \vdash \underbrace{\Gamma^\perp}_{\text{input}} ; \underbrace{\Pi}_{\text{output}}$$

$$\dfrac{\dfrac{\Gamma, \overset{\text{input}}{\underline{A}} \vdash \Pi}{\vdash \Gamma^\perp, \underline{A^\perp}; \Pi}}{\vdash \Gamma^\perp; \underbrace{\downarrow A^\perp}_{\text{output}}, \Pi}$$

$$\dfrac{\vdash \Gamma^\perp, \overset{\text{input}}{\underline{A^-}}; \Pi}{\vdash \Gamma^\perp; \underbrace{\downarrow A^-}_{\text{output}}, \Pi}$$

$$\boxed{\neg A = \downarrow A^\perp}$$

shifts: switching input and output

$$c^* \vdash \underbrace{\Gamma^\perp}_{\text{input}} ; \underbrace{\Pi}_{\text{output}}$$

$$A^+ ::= 1 \mid A_1^+ \otimes A_2^+ \mid 0 \mid A_1^+ \oplus A_2^+ \mid \downarrow A^-$$

$$A^- ::= \perp \mid A_1^- \parr A_2^- \mid \top \mid A_1^- \& A_2^- \mid \uparrow A^+$$

shifts: switching input and output

$$c^* \vdash \underbrace{\Gamma^\perp}_{\text{input}} ; \underbrace{\Pi}_{\text{output}}$$

$$A^+ ::= 1 \mid A_1^+ \otimes A_2^+ \mid 0 \mid A_1^+ \oplus A_2^+ \mid \downarrow A^-$$

$$A^- ::= \bot \mid A_1^- \parr A_2^- \mid \top \mid A_1^- \& A_2^- \mid \uparrow A^+$$

polarized logic

$CP^{\pm}$: a polarized CP

$$P \vdash \Delta^{-} ; \Pi^{+}$$

input output

$$A^{+} ::= 1 \mid A_1^{+} \otimes A_2^{+} \mid 0 \mid A_1^{+} \oplus A_2^{+} \mid \downarrow A^{-}$$

$$A^{-} ::= \bot \mid A_1^{-} \mathbin{\rotatebox[origin=c]{180}{\&}} A_2^{-} \mid \top \mid A_1^{-} \& A_2^{-} \mid \uparrow A^{+}$$

$CP^{\pm}$: a polarized CP

Pfenning and Griffith 2015

$$P \vdash \Delta^{-}; \Pi^{+}$$

input output

$$A^{+} ::= 1 \mid A_1^{+} \otimes A_2^{+} \mid 0 \mid A_1^{+} \oplus A_2^{+} \mid \downarrow A^{-}$$

$$A^{-} ::= \bot \mid A_1^{-} ⅋ A_2^{-} \mid \top \mid A_1^{-} \& A_2^{-} \mid \uparrow A^{+}$$

CP$^{\pm}$: a polarized CP

$\boxed{\texttt{Store}^{+}}$

$$\downarrow((\mathsf{Prod} \multimap \mathsf{CC} \multimap \uparrow(1 \oplus 1))$$

$$\& \, (\mathsf{Prod} \multimap \uparrow(\mathsf{Cost} \otimes 1)))$$

$\mathrm{CP}^{\pm}$: a polarized CP

$\texttt{Store}^{+}$

input choice of buy

$$\downarrow((\mathsf{Prod} \multimap \mathsf{CC} \multimap \uparrow(1 \oplus 1))$$

switch

or quote

$$\&(\mathsf{Prod} \multimap \uparrow(\mathsf{Cost} \otimes 1)))\,)$$

$\text{CP}^{\pm}$: a polarized CP

`Store`$^{+}$

input choice of buy

$$\downarrow((\underline{\text{Prod}} \multimap \text{CC} \multimap \uparrow(1 \oplus 1))$$

switch **input**

or quote

$$\underline{\&}\,(\text{Prod} \multimap \uparrow(\text{Cost} \otimes 1)))\,)$$

$\mathrm{CP}^{\pm}$: a polarized CP

$\texttt{Store}^{+}$

input choice of buy

$$\downarrow((\mathsf{Prod} \multimap \mathsf{CC} \multimap \uparrow(1 \oplus 1))$$

switch input input

or quote

$$\mathbin{\&} (\mathsf{Prod} \multimap \uparrow(\mathsf{Cost} \otimes 1)))$$

$CP^{\pm}$: a polarized CP

`Store+`

input choice of buy

switch

$$\downarrow((\underline{\mathsf{Prod}} \multimap \underline{\mathsf{CC}} \multimap \uparrow(\underline{1 \oplus 1}))$$

switch input input output

or quote

$$\underline{\&}\,(\mathsf{Prod} \multimap \uparrow(\mathsf{Cost} \otimes 1)))\,)$$

$\mathrm{CP}^{\pm}$: a polarized CP

$\mathtt{Store}^{+}$

input choice of buy

switch

$$\downarrow\big((\mathrm{Prod} \multimap \mathrm{CC} \multimap \uparrow(1 \oplus 1))$$

switch

input

input

output

or quote

$$\&\ (\mathrm{Prod} \multimap \uparrow(\mathrm{Cost} \otimes 1))\big)$$

input

$CP^{\pm}$: a polarized CP

$\texttt{Store}^{+}$

input choice of buy
switch

$$\downarrow((\mathsf{Prod} \multimap \mathsf{CC} \multimap \uparrow(1 \oplus 1))$$

switch
input
input
output

or
quote
switch

$$\&\, (\mathsf{Prod} \multimap \uparrow(\mathsf{Cost} \otimes 1)))\,)$$

input
output

$CP^{\pm}$ is CP
(more or less)

CP$^{\pm}$ is CP
(more or less)

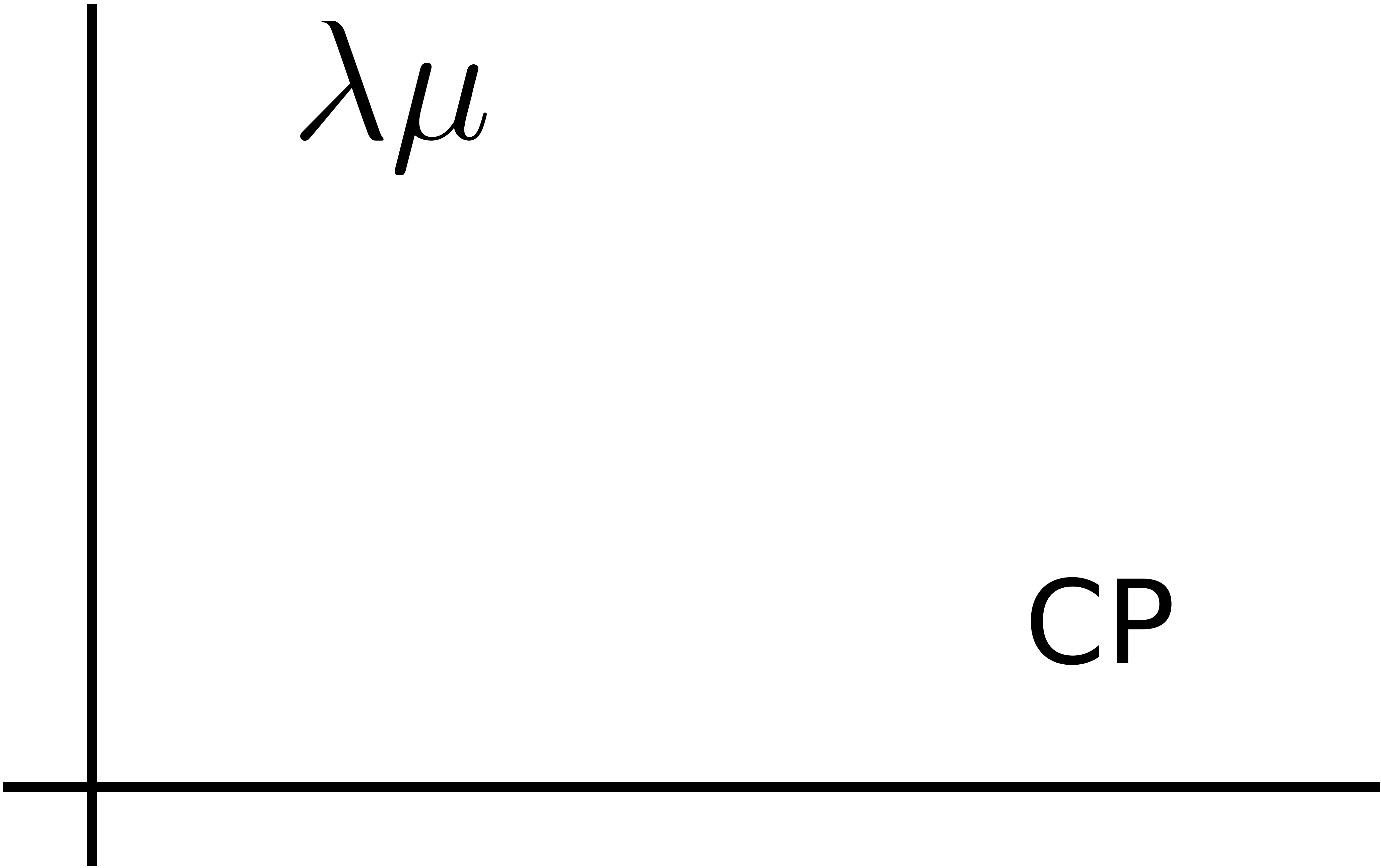
- explicit input and output

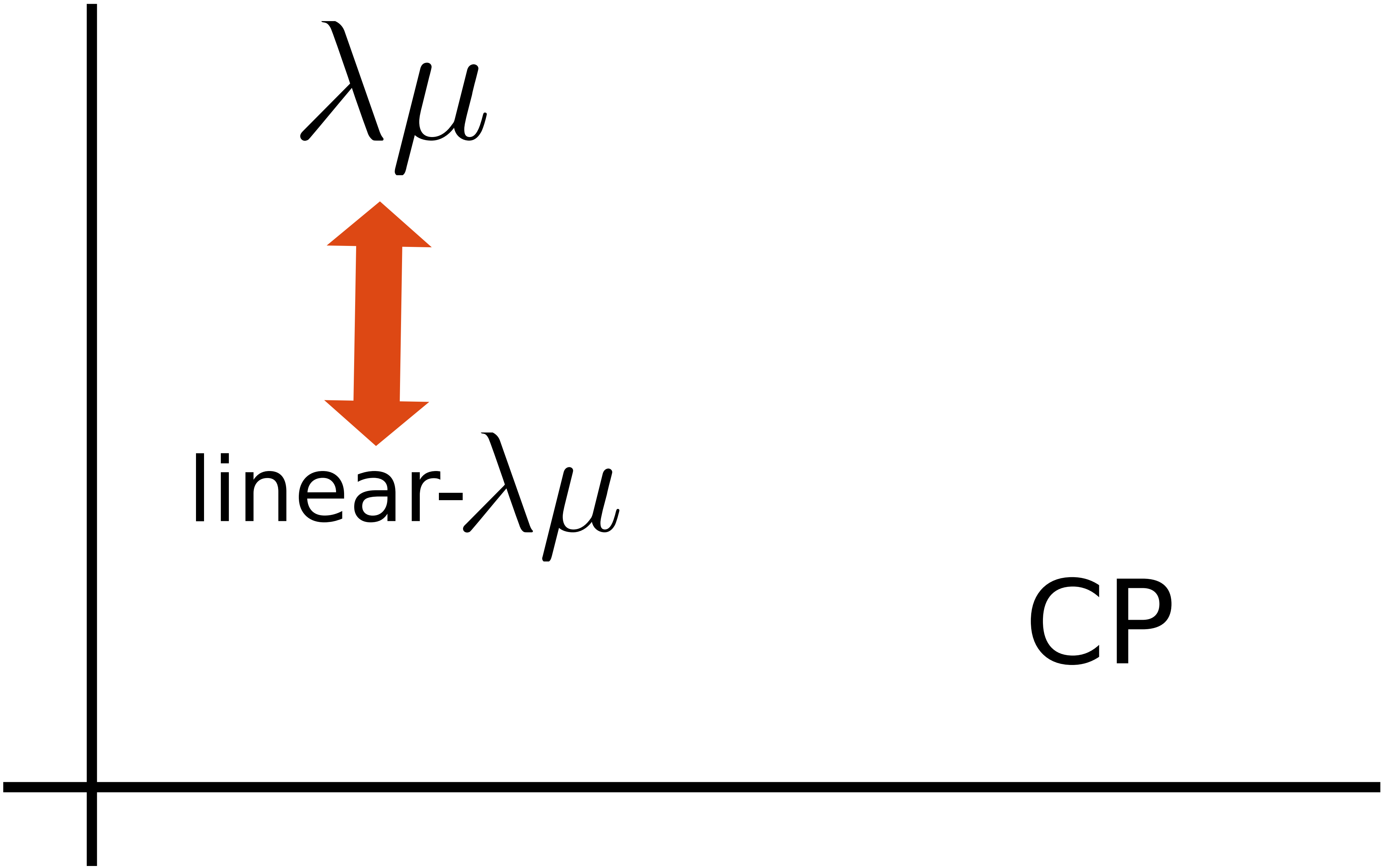
$CP^{\pm}$ is CP (more or less)

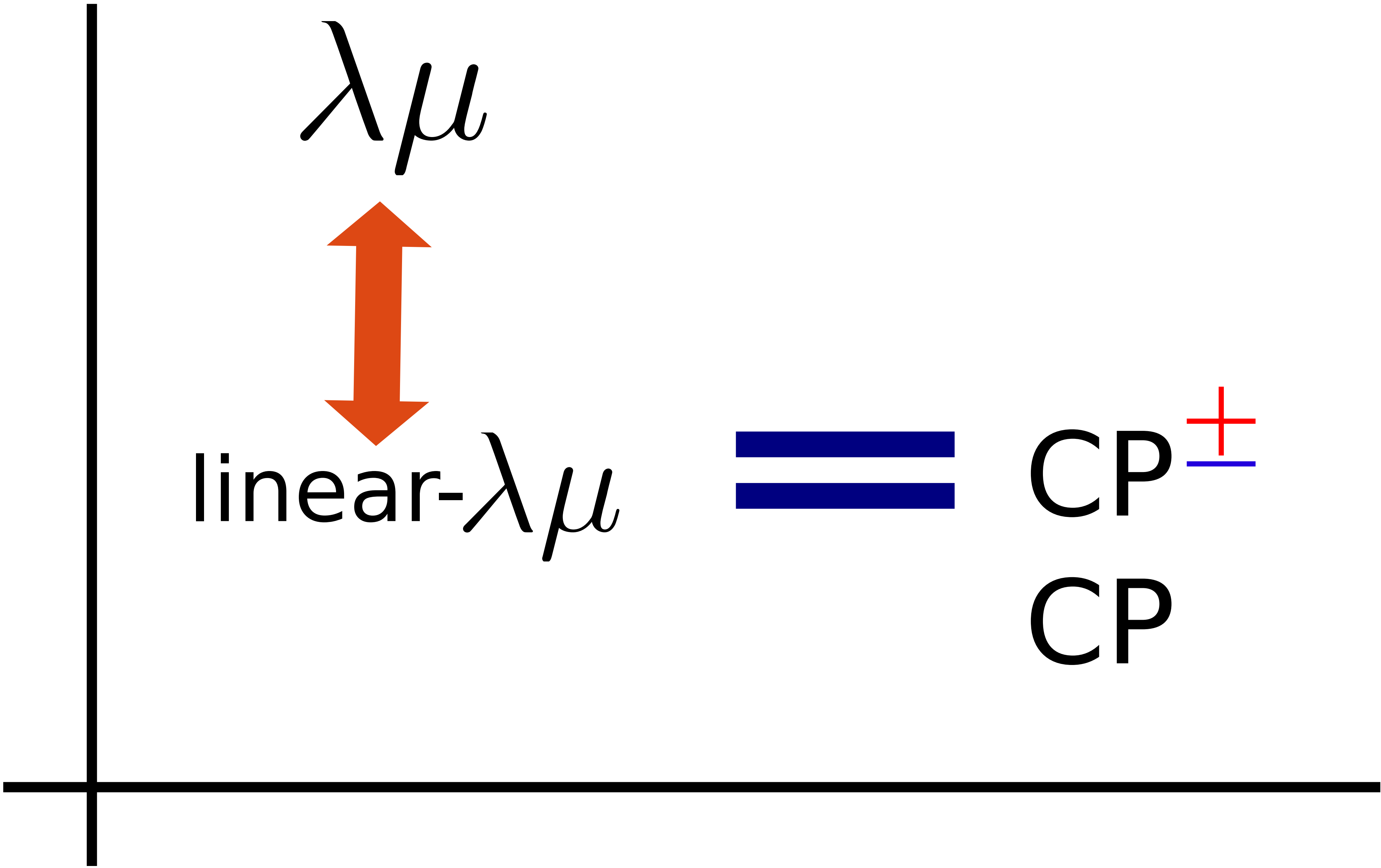
- explicit input and output
- fully dual types

$CP^{\pm}$ is CP
(more or less)

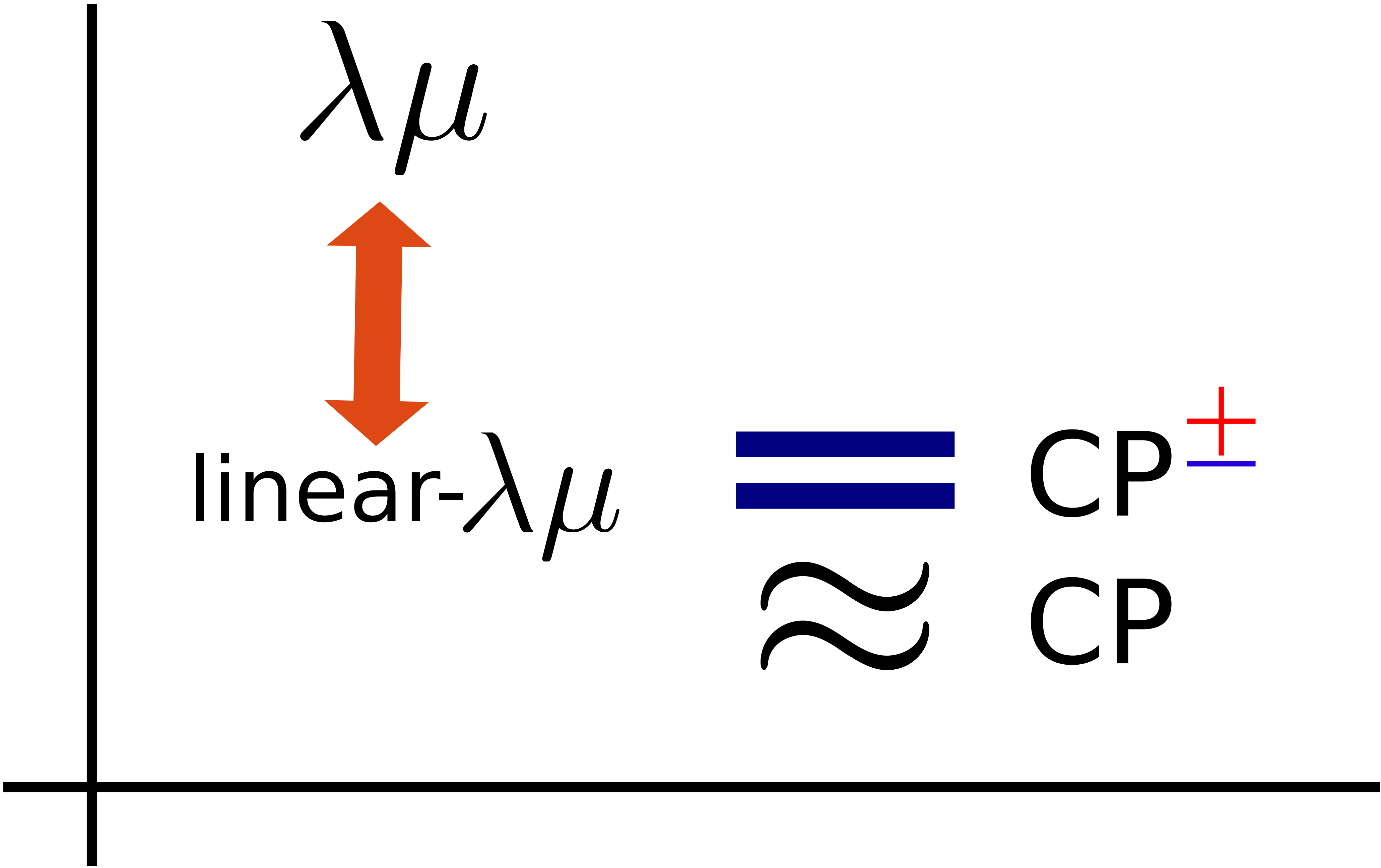
- explicit input and output
- fully dual types
- can always switch


λμ
CP

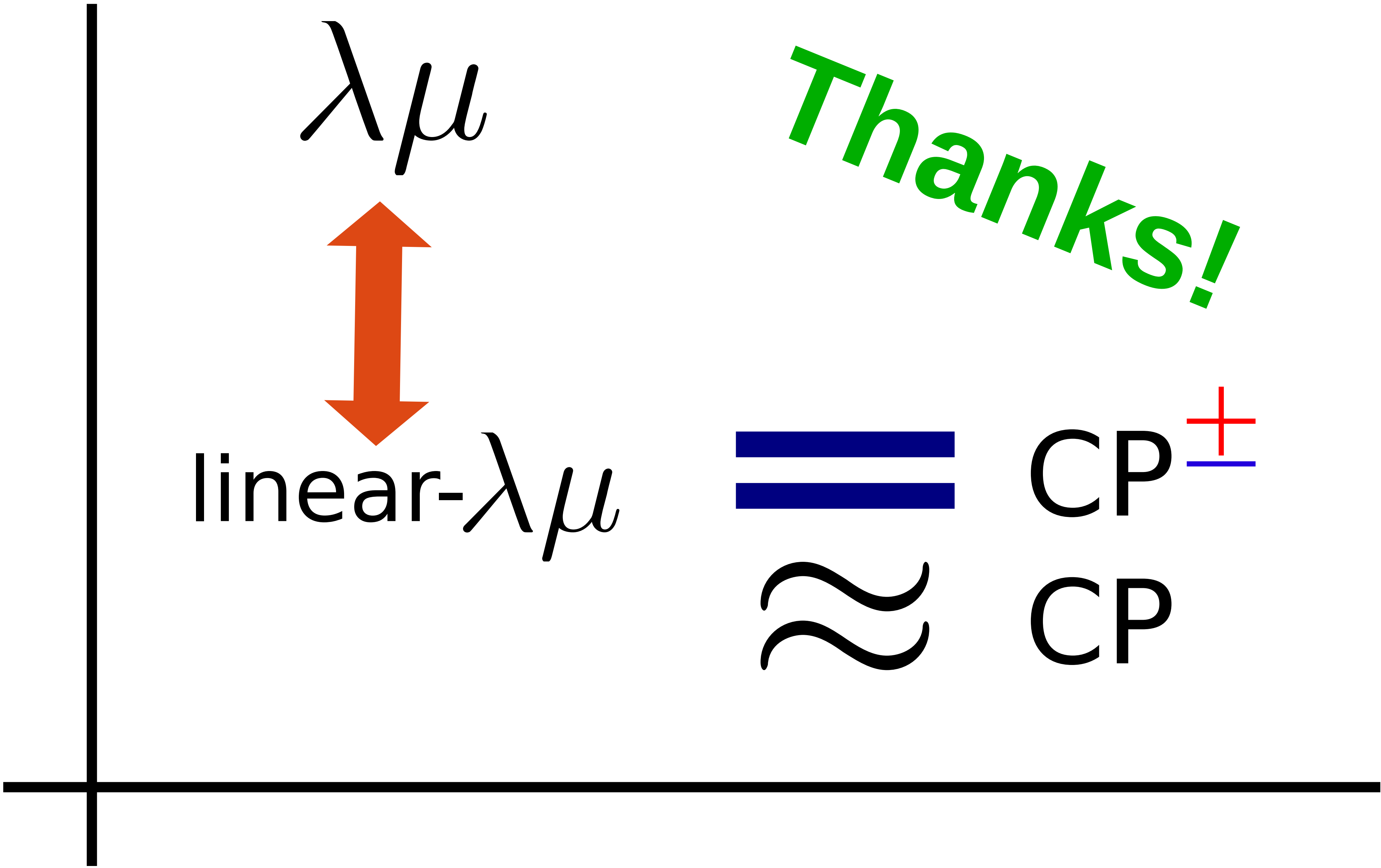

$\lambda\mu$
linear-$\lambda\mu$
CP



λμ
linear-λμ
CP±
CP



λμ
linear-λμ
CP±
CP



λμ
Thanks!
linear-λμ
=
CP±
≈
CP

References

- Phil Wadler, Propositions are sessions, ICFP 2012.
- Luis Caires and Frank Pfenning, Session types as intuitionistic linear propositions, CONCUR 2010.
- Michel Parigot, Lambda-mu calculus: an algorithmic interpretation of classical natural deduction, Logic Programming and Automated Reasoning 1992.
- Paul-Andre Mellies and Nicolas Tabareau, Resource modalities in tensor logic, Annals of Pure and Applied Logic 2010.
- Frank Pfenning and Dennis Griffith, Polarized substructural session types, Foundations of Software Science and Computation Structures 2015.