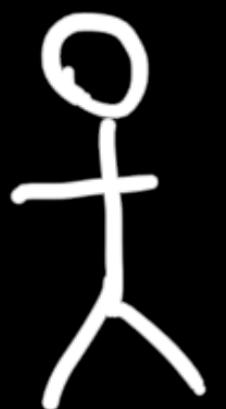


Forwarders as Process Compatibility

Sonia Marin
University of Birmingham
filipendule.github.io

joint work w. M. Carbone & C. Schürmann
IT-University of Copenhagen

Binary Session Types



P_1

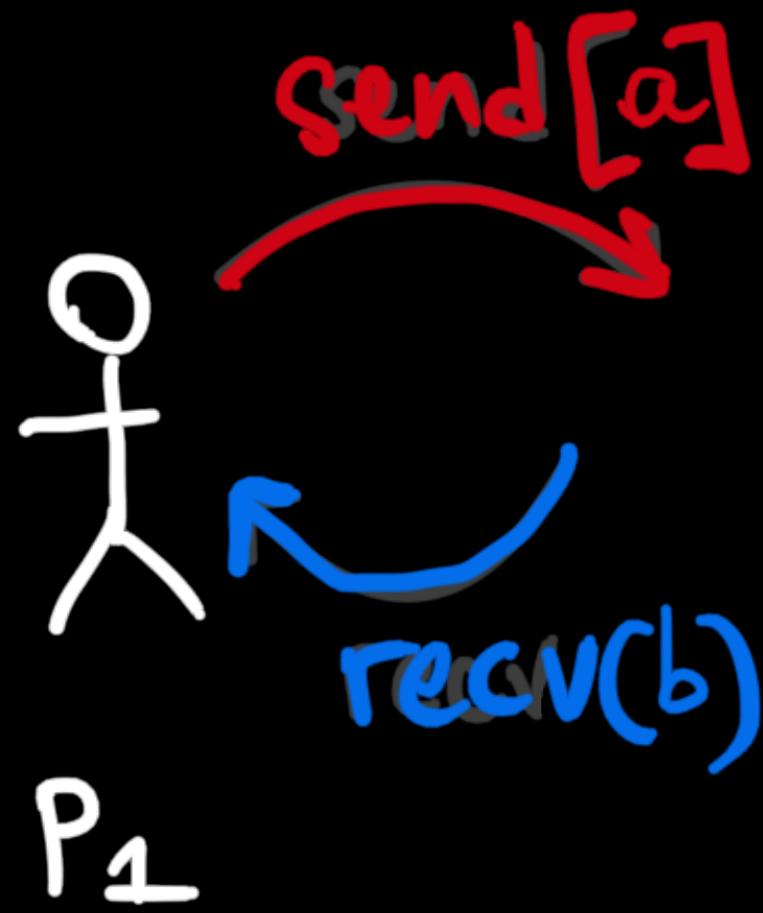


P_2

$P_1 = \text{send}[a]. \text{recv}(b). \text{close}$

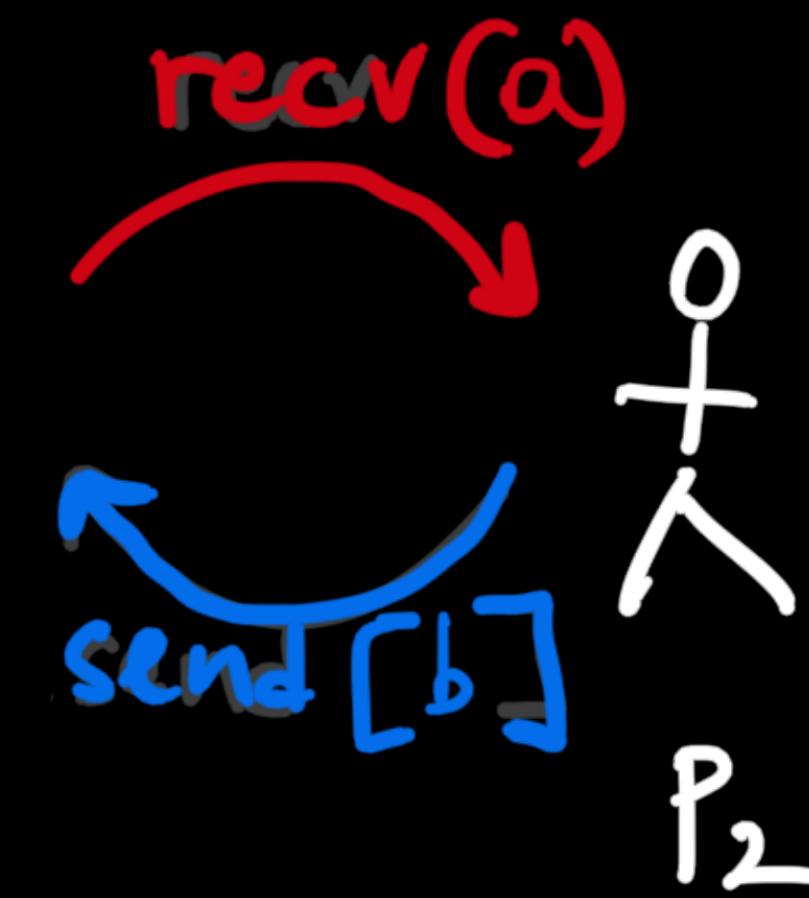
$P_2 = \text{recv}(a). \text{send}[b]. \text{wait}$

Binary Session Types & Linear Logic



$P_1 = \text{send}[a]. \text{recv}(b). \text{close}$

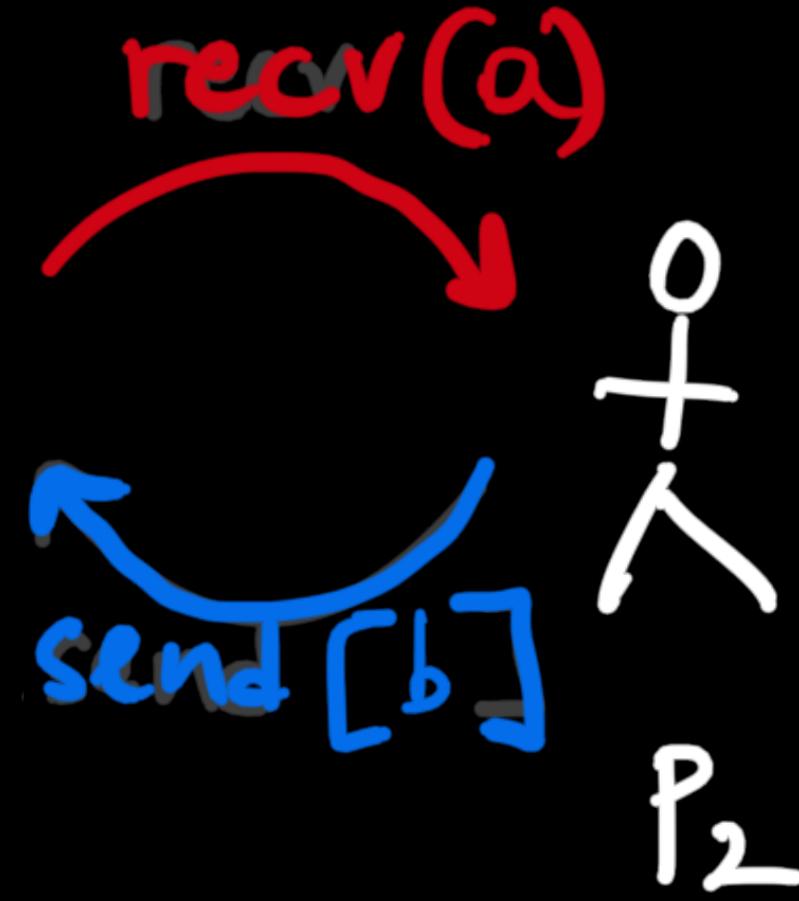
$P_2 = \text{recv}(a) . \text{Send}[b] . \text{Wait}$



$$\vdash x_2 : A \times (B \rightarrow \top)$$
$$\vdash x_1 : A \rightarrow (B \times \bot)$$

[Caires & Pfennig | Wadler]

Proofs - as - processes



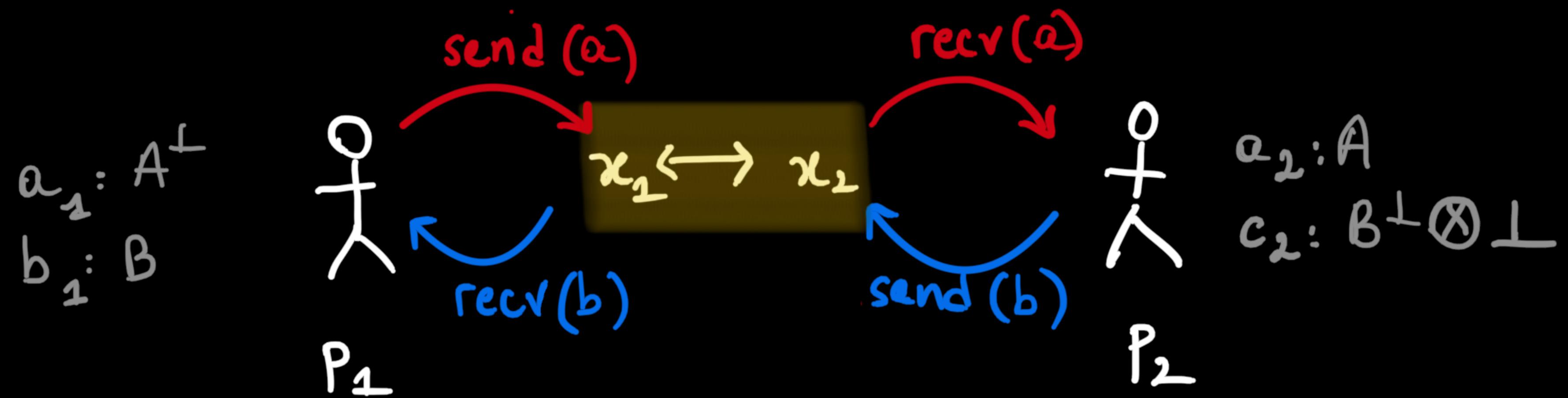
$$Q_2 = a \leftarrow a,$$

$$\frac{\perp}{\frac{Q_2 \vdash a : A^\perp, a_1 : A}{\text{wait. } Q_2 \vdash a : A^\perp, x_1 : \perp, a_1 : A}}$$

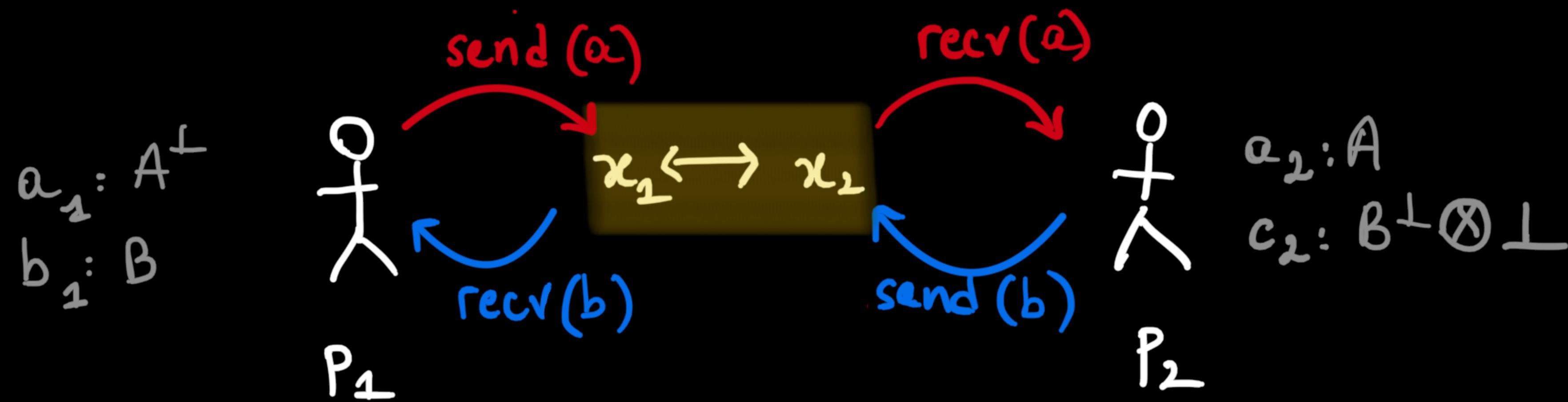
$$\frac{Q_1 = b \leftrightarrow b_1}{Q_1 \vdash b : B, b_1 : B^\perp}$$

$$\frac{\otimes}{\frac{\text{send}[b \triangleright Q_1]. \text{wait}. Q_2 \vdash a : A^\perp, x_1 : B \otimes \perp, a_1 : A, b_1 : B^\perp}{P_1 = \text{recv}(a). \text{send}[b \triangleright Q_1]. \text{wait}. Q_2 \vdash x_1 : A^\perp \wp (B \otimes \perp), a_1 : A, b_1 : B^\perp}}$$

Reduction - as - communication



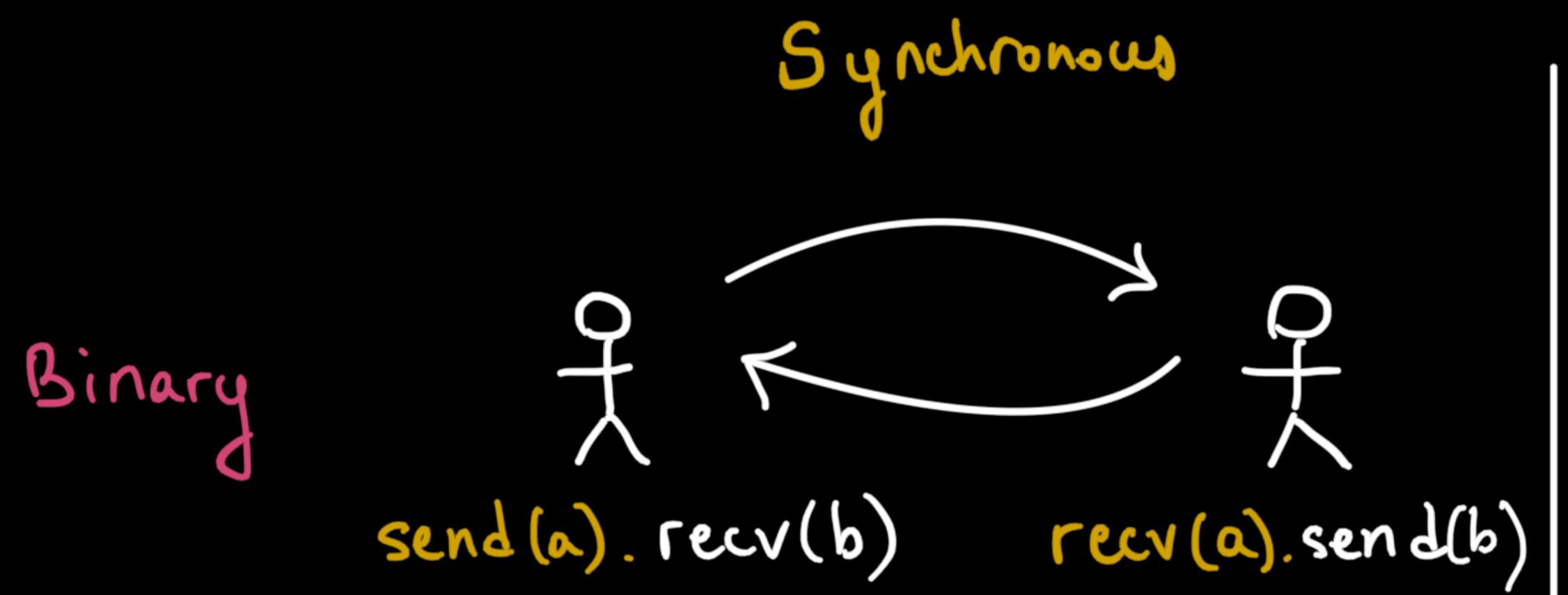
Reduction - as - communication



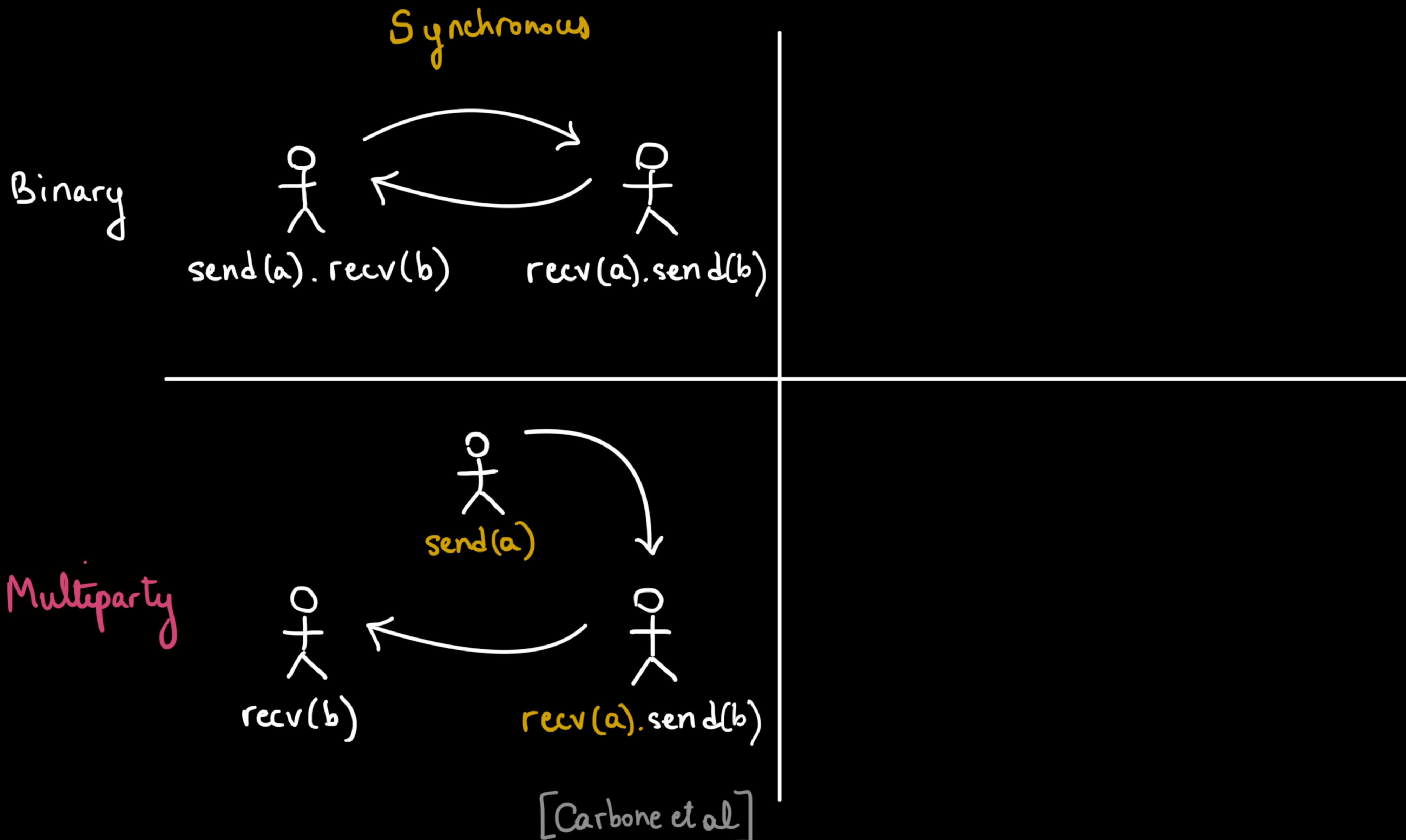
$$\frac{\begin{array}{c} \checkmark \\ P_1 \vdash x_1 : A \otimes B^\perp \& I \\ \hline \end{array} \quad \begin{array}{c} \checkmark \\ P_2 \vdash x_2 : A^\perp \& B \otimes \perp \end{array}}{(\forall x_1 x_2) (P_1 \parallel P_2) \vdash .} \text{CUT}$$

dual types!

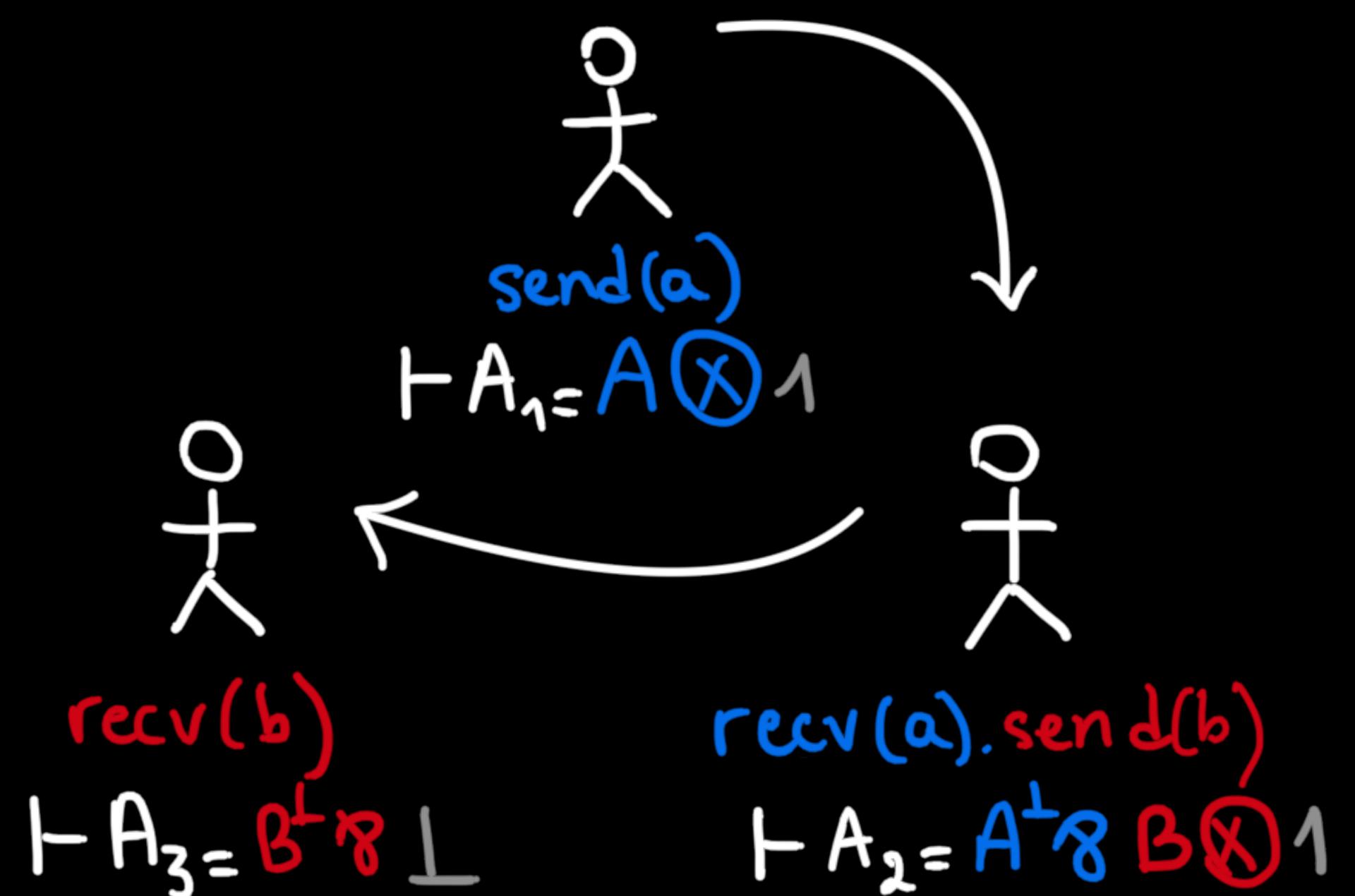
Generalizing CUT



Generalizing CUT



Coherence

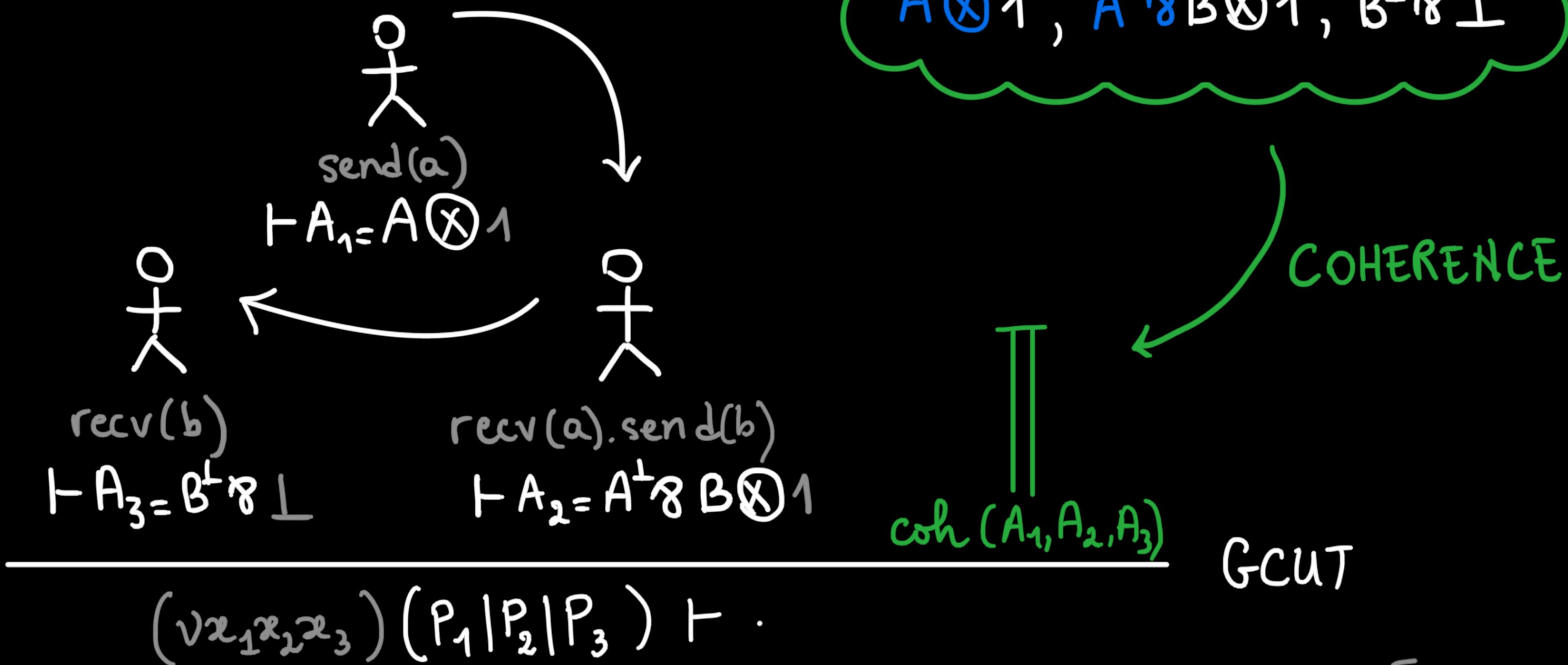


Cohherence

$$\frac{\begin{array}{c} \text{Diagram showing three participants: Alice, Bob, and Charlie. Alice sends message } a \text{ to Bob. Bob receives } a \text{ and sends } b \text{ to Charlie. Charlie receives } b. \\ \text{Alice's state: } \vdash A_1 = A \otimes 1 \\ \text{Bob's state: } \vdash A_2 = A^\perp \otimes B \otimes 1 \\ \text{Charlie's state: } \vdash A_3 = B^\perp \otimes \perp \end{array}}{(\forall x_1 x_2 x_3) (P_1 | P_2 | P_3) \vdash \cdot}$$

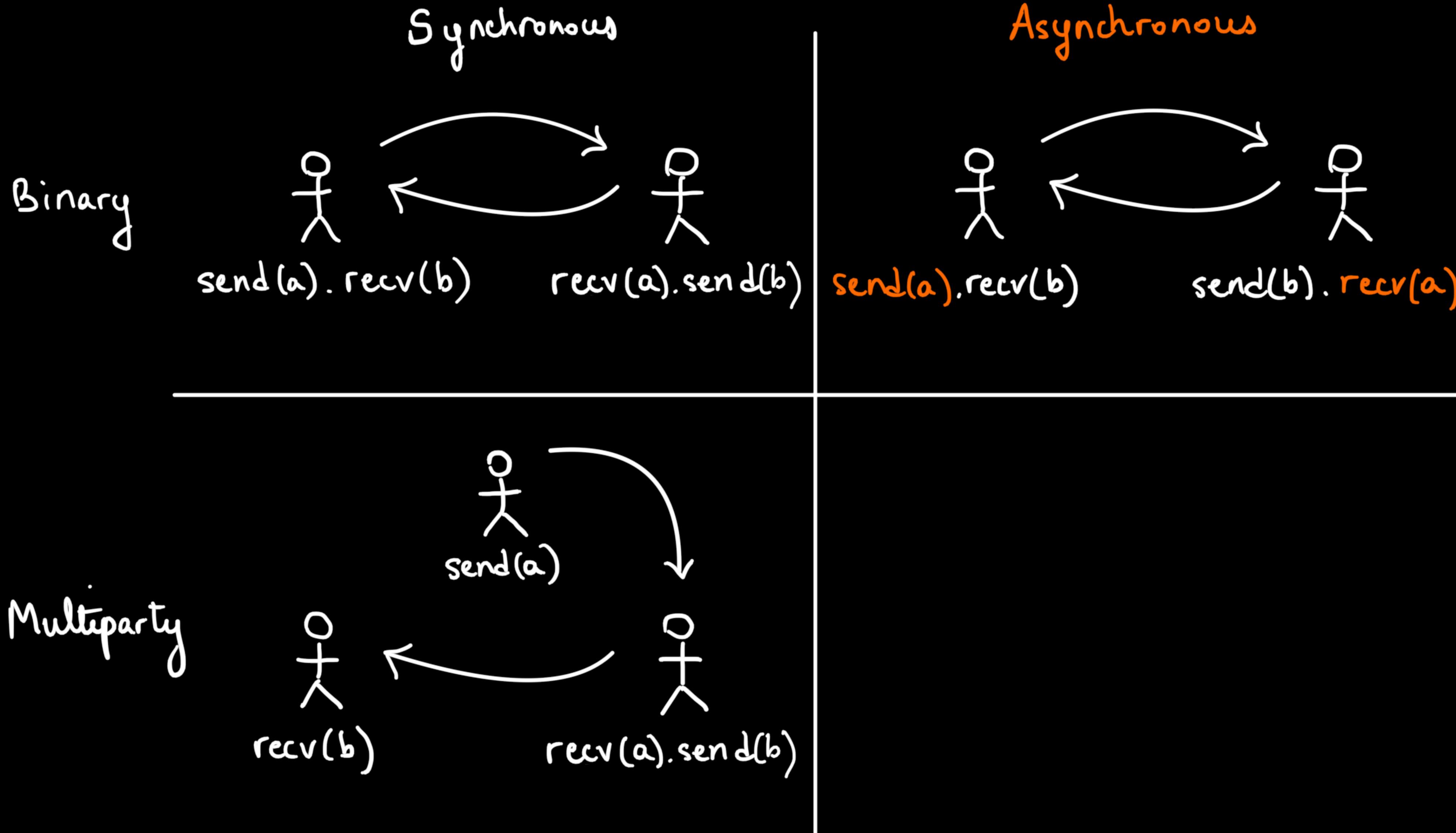
Π $\text{coh}(A_1, A_2, A_3)$ GCUT

Coherence



[Carbone et al]

Generalizing CUT



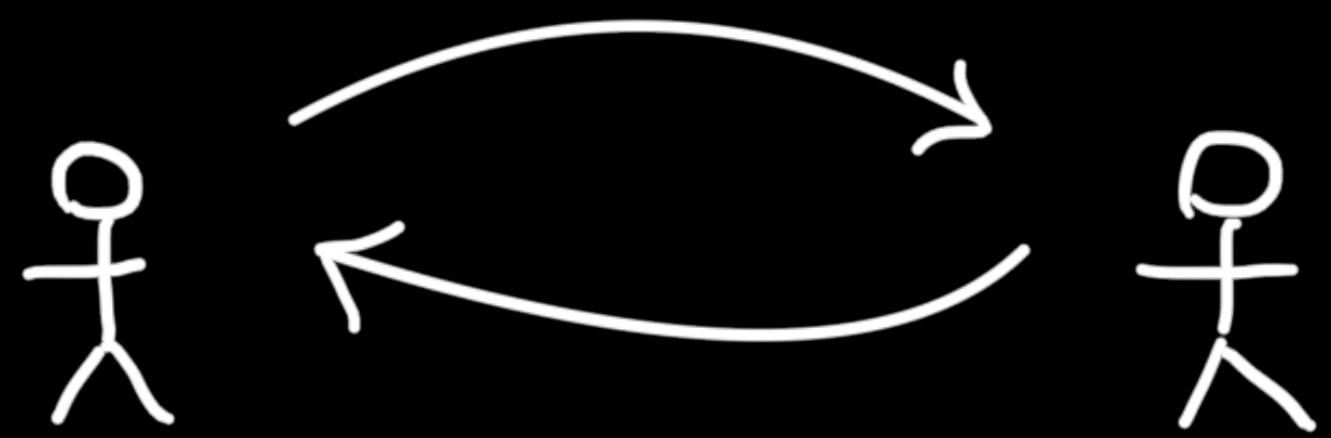
Griss-cross



$\text{send}(a), \text{recv}(b)$
 $\vdash A \otimes B^\perp \& 1$

$\text{send}(b), \text{recv}(a)$
 $\vdash B \otimes A^\perp \& \perp$

Griss-cross



send(a).recv(b)

$\vdash A_1 = A \otimes B^\perp \wp 1$

send(b).recv(a)

$\vdash A_2 = B \otimes A^\perp \wp \perp$

T

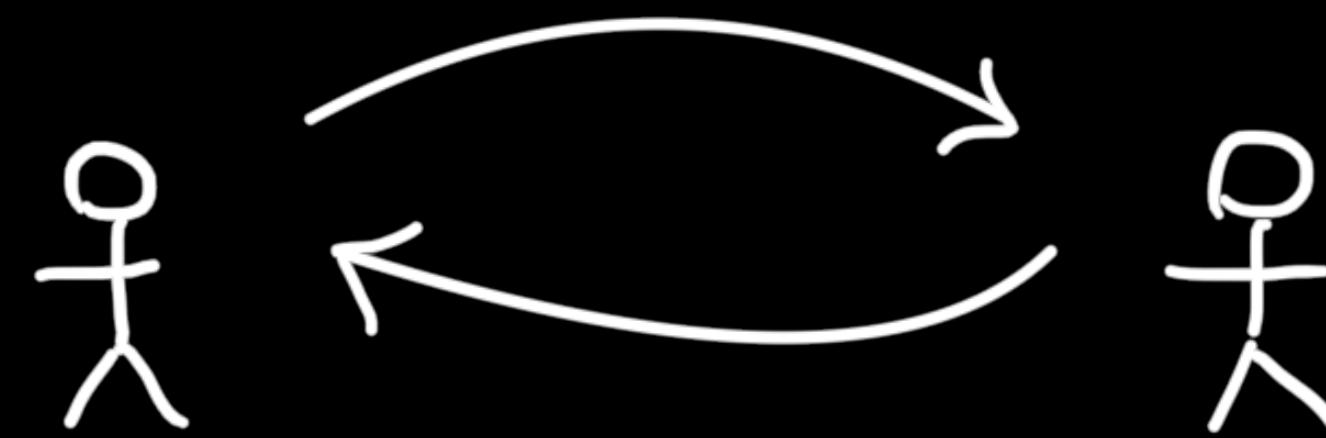
comp(A₁, A₂)

$(\forall x_1 x_2) (P_1 | P_2) \vdash .$

Griss-cross

$$\frac{\perp, 1}{\boxed{[A^\perp] B \otimes \perp, [B^\perp] A \otimes 1}} = \boxed{A^\perp \otimes B \otimes \perp, B^\perp \otimes A \otimes 1}$$

via BUFFERING



send(a).recv(b)

$$\vdash A_1 = A \otimes B^\perp \otimes 1$$

send(b).recv(a)

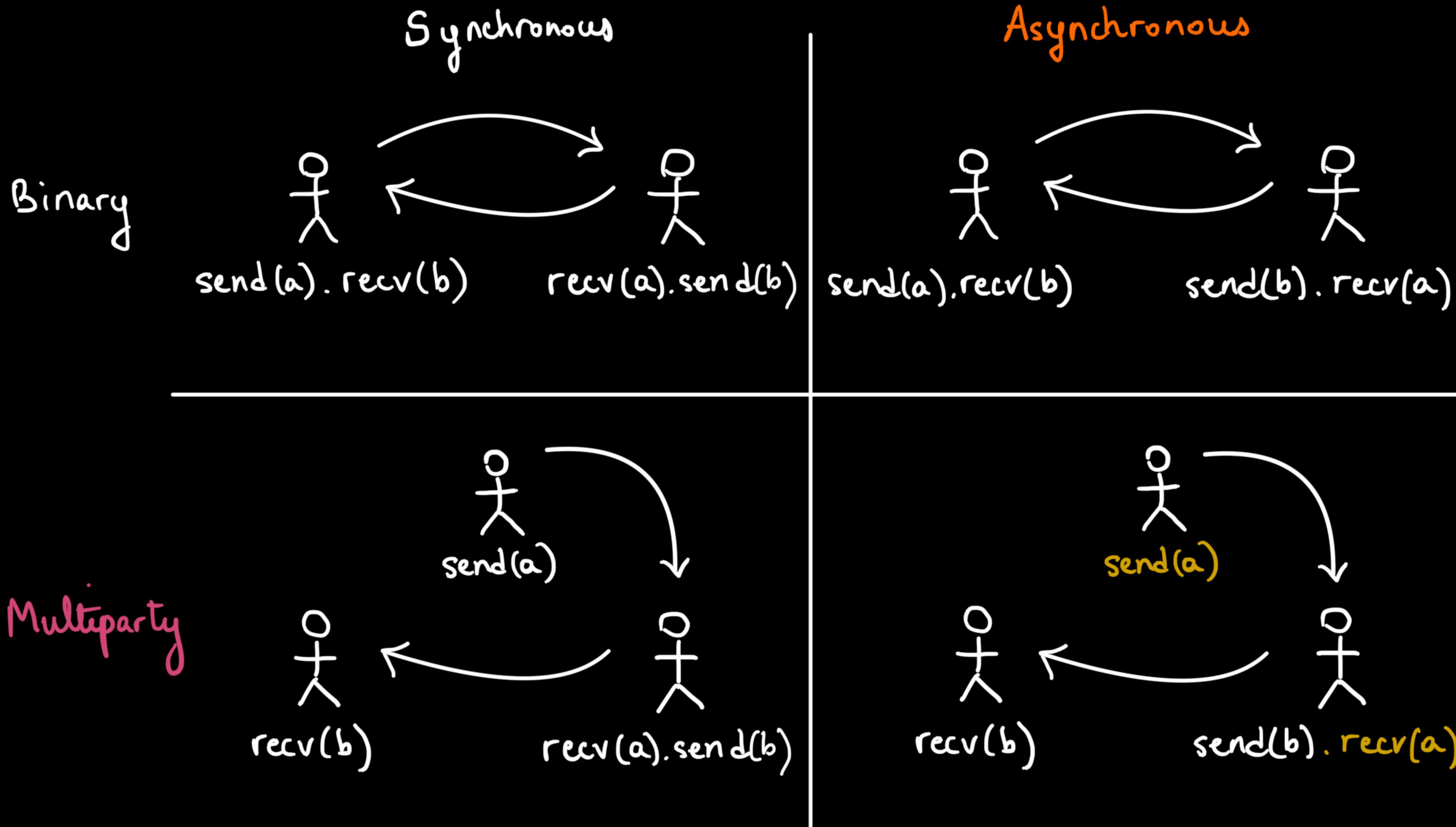
$$\vdash A_2 = B \otimes A^\perp \otimes \perp$$

T

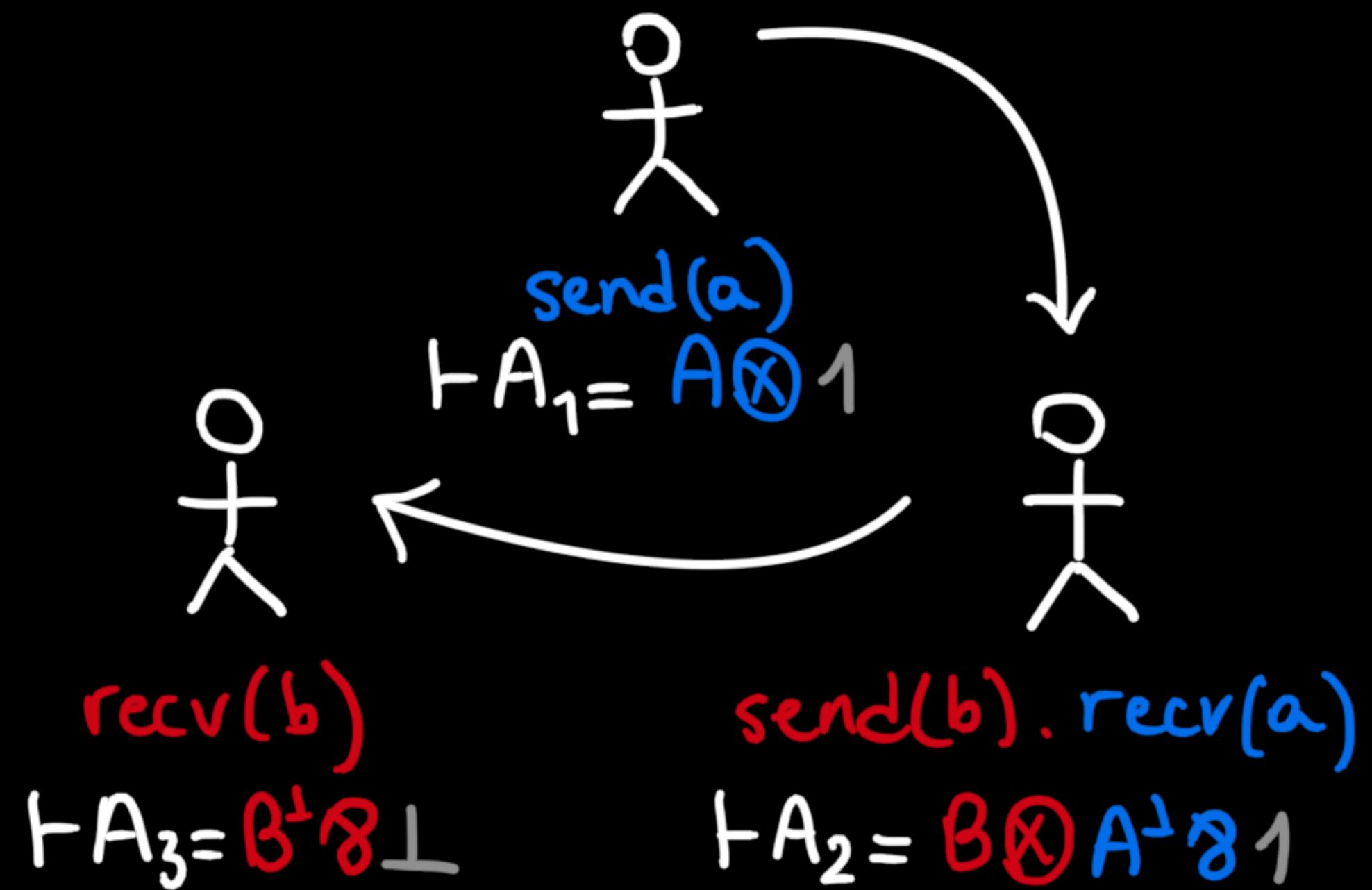
$$\text{comp}(A_1^\perp, A_2^\perp)$$

$$\frac{}{(v x_1 x_2) (P_1 | P_2) \vdash .}$$

Generalizing CUT



Forwarders



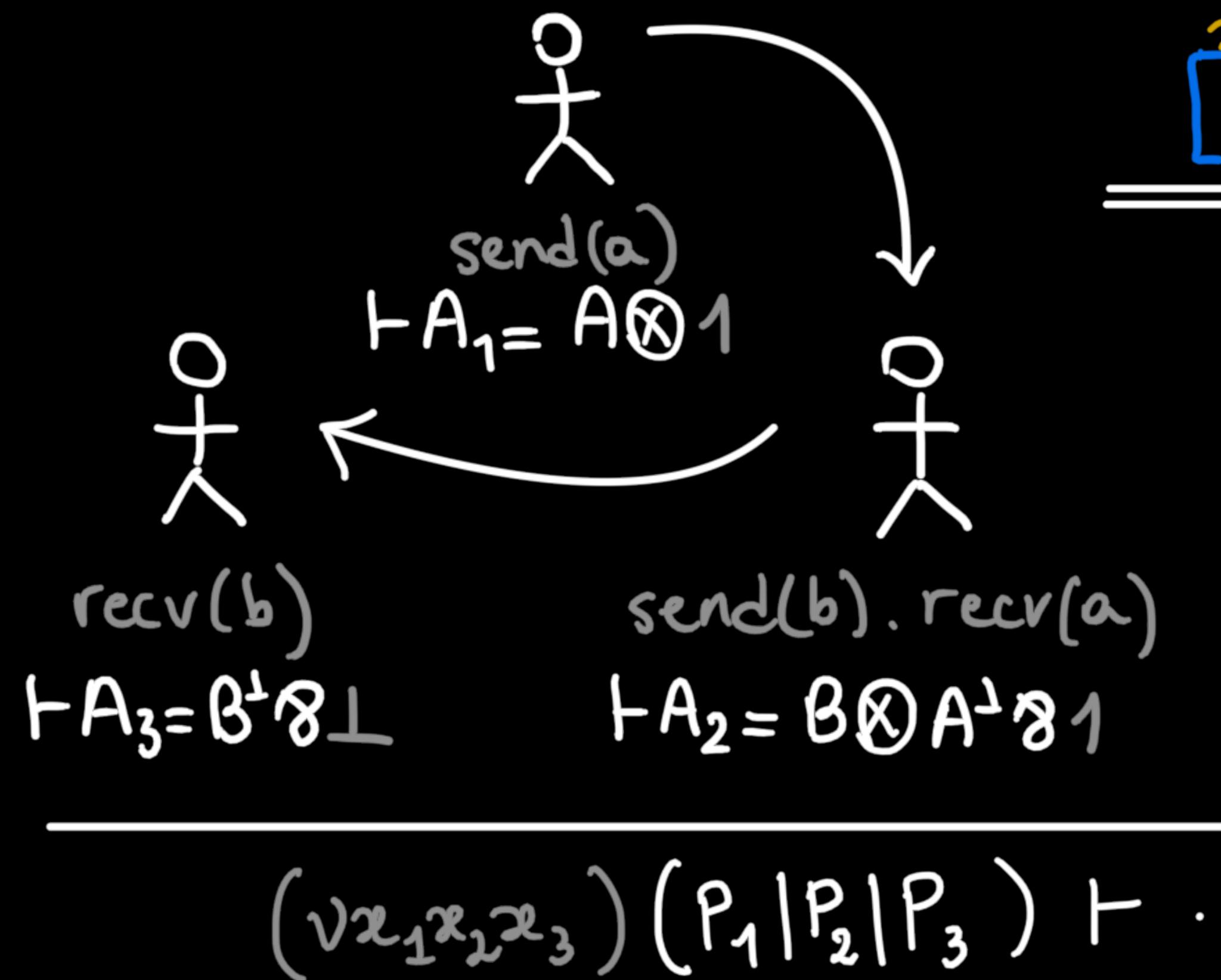
Forwards

$$\frac{\begin{array}{c} \text{○} \xrightarrow{\text{send}(a)} \text{○} \\ \vdash A_1 = A \otimes 1 \\ \text{○} \xleftarrow{\text{recv}(b)} \text{○} \\ \vdash A_3 = B^\perp \otimes \perp \\ \text{○} \xrightarrow{\text{send}(b) . \text{recv}(a)} \text{○} \\ \vdash A_2 = B \otimes A^\perp \otimes 1 \end{array}}{\text{fwd } (A_1, A_2, A_3)}$$

\prod

$$(\forall x_1 x_2 x_3) (P_1 | P_2 | P_3) \vdash .$$

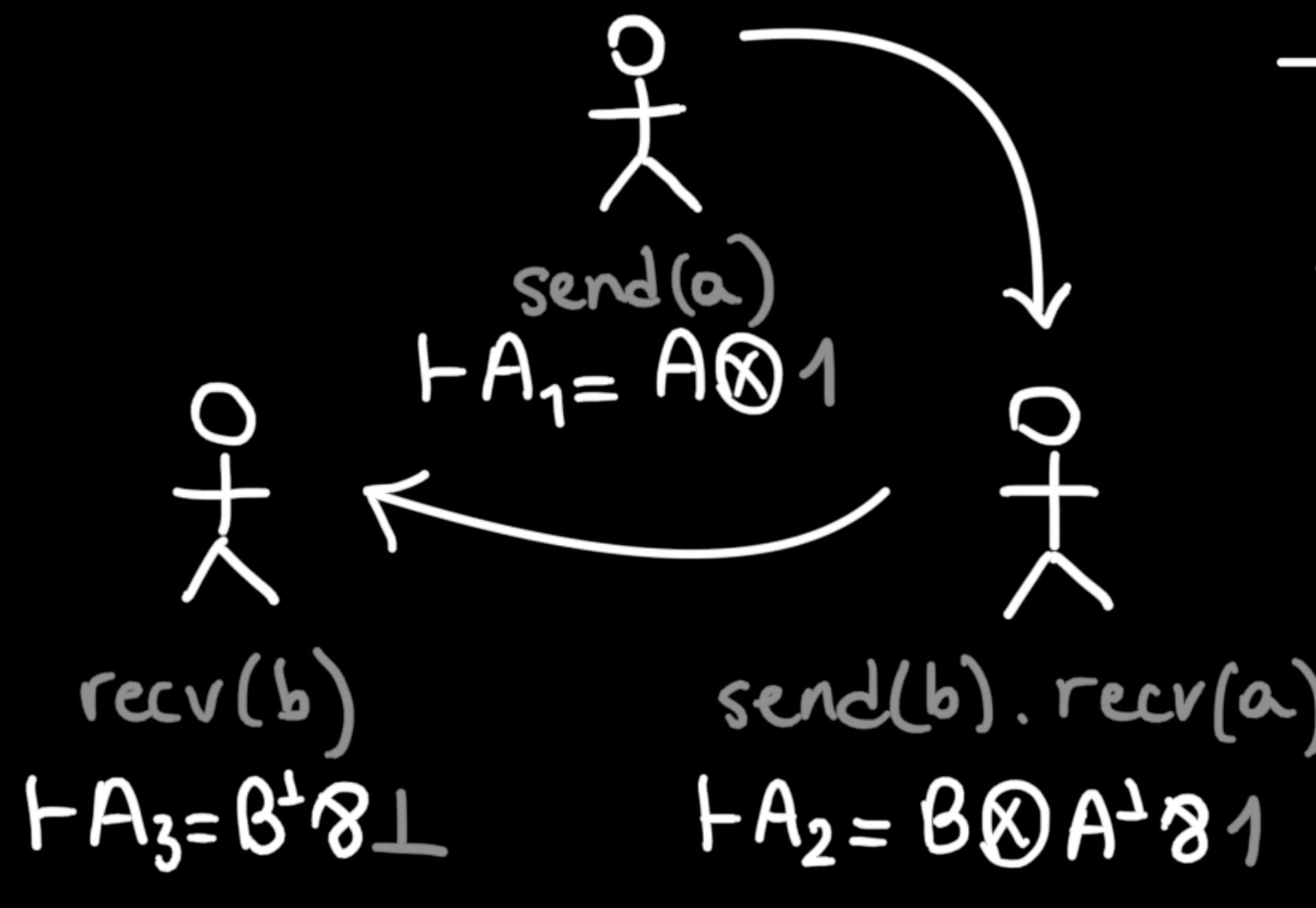
Forwards



$$\frac{x_2: [A^\perp] \quad x_1: \perp, x_3: [B^\perp]}{x_1: A^\perp \wp \perp, x_2: B^\perp \wp A \otimes \perp, x_3: B \otimes 1}$$

x_2 \prod x_3
 \uparrow \uparrow
 $fwd(A_1^\perp, A_2^\perp, A_3)$

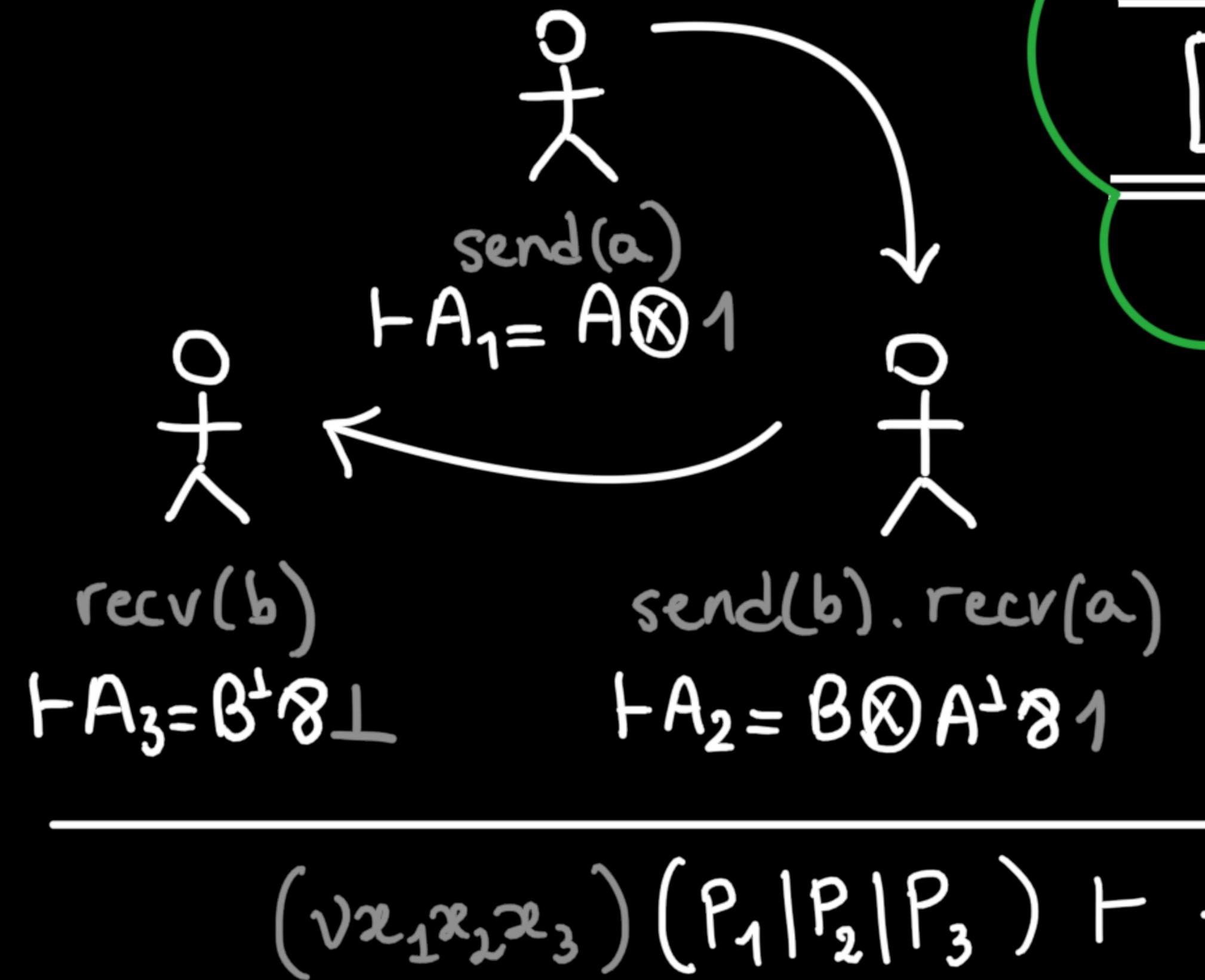
Forwards



$$\frac{\frac{x_1 : \perp, [B^\perp] x_2 : \perp \quad x_3 : B \otimes 1}{x_2 : A^\perp x_1 : [B^\perp] x_2 : A \otimes \perp, x_3 : B \otimes 1} \quad \text{fwd } (A_1^\perp, A_2^\perp, A_3^\perp)}{x_1 : A^\perp \wp \perp, x_2 : B^\perp \wp A \otimes \perp, x_3 : B \otimes 1}$$

$(\forall x_1 x_2 x_3) (P_1 | P_2 | P_3) \vdash .$

Forwarders



$$\begin{array}{c}
 \boxed{\dfrac{x_1 : \perp^{\overbrace{x_3}}, \quad x_2 : \perp^{\overbrace{x_3}}, \quad x_3 : 1^{\overbrace{x_1 x_2}}}{x_1 : \perp, [B^\perp] \quad x_2 : \perp, \quad x_3 : B \otimes^{\overbrace{x_2}} 1}} \\
 \dfrac{\boxed{\dfrac{x_2 : \perp^{\overbrace{x_1}}}{[A^\perp] \quad x_1 : \perp, [B^\perp]}} \quad x_2 : A \otimes^{\overbrace{x_1}} \perp, \quad x_3 : B \otimes 1}{x_1 : A^\perp \wp_{\overbrace{x_2}} \perp, \quad x_2 : B^\perp \wp_{\overbrace{x_3}} A \otimes \perp, \quad x_3 : B \otimes 1}}
 \end{array}$$

ASYNCHRONOUS FORWARDERS

\prod

$fwd(A_1^\perp, A_2^\perp, A_3^\perp)$

Multiparty compatibility - logically

Asynchronous forwarders characterize precisely

the multiparty compatible types.



semantic notion from [Denielou & Yoshida]
refined by several authors
e.g. [Scalas et al.]

Multiparty compatibility - logically

Asynchronous forwarders characterize precisely
the multiparty compatible types.

and still satisfy cut-elimination
i.e. $(\forall x_1 x_2)(F_1 \parallel F_2)$ is an async. forwarder

♪ Chank you!

The image features a central, handwritten-style text "Chank you!" in white. It is surrounded by a variety of colorful musical symbols and lines. At the top left is a yellow treble clef. To the right of the main text is a blue dynamic symbol labeled "bbd". Below the main text, there's a purple eighth-note-like symbol with a wavy line. To the left of the text, there's an orange note with a blue grace note above it. A pink wavy line extends from the bottom left towards the center. A blue fermata symbol is located at the bottom left. A yellow wavy line extends from the bottom right towards the center. An orange sixteenth-note-like symbol is positioned between the two wavy lines. A purple sharp sign symbol is placed to the right of the main text. The entire composition is set against a black background.