Session Types with Classical and Intuitionistic Logic

Floris Reuvers

Radboud University

January 19, 2026

Table of Contents

Introduction first paper

- Propositions as Sessions
- Philip Wadler
- Distributed systems
- π-calculus (CP)
- Channels
- GV

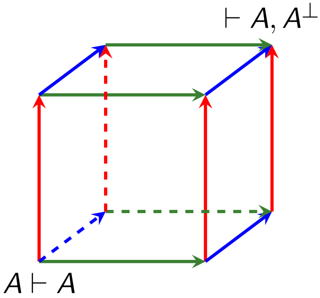
λ-calculus	**π-calculus**
Evaluation of functions	Communication between processes
Data types	Session Types
Traditional Logic	Linear Logic

Introduction second paper

- Session Type Systems based on Linear Logic: Classical versus Intuitionistic
- Bas van den Heuvel and Jorge A. Pérez
- Locality Principle
- πCLL^{CP}, πILL and πULL

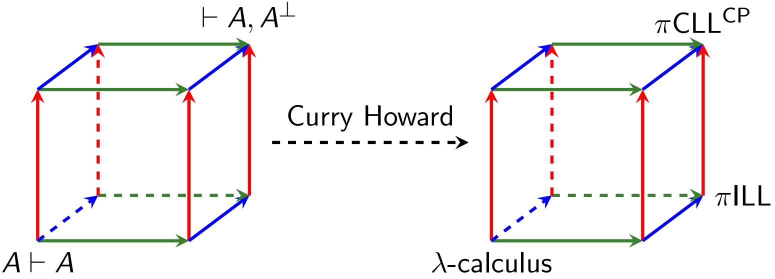
	πCLL^{CP}	πILL	πULL
Locality	No	Yes	Yes and no

Logic Cube



- Intuitionistic $\longrightarrow$ Classical
- Traditional $\longrightarrow$ Linear
- Natural Deduction $\longrightarrow$ Sequent Calculus

Logic Cube: Curry Howard



- Intuitionistic → Classical
- Traditional → Linear
- Natural Deduction → Sequent Calculus

Table of Contents

Intuitionistic Sequent Calculus

Sequents

Sequents are of the following form:

$$A_1, A_2, \ldots, A_n \vdash B \quad \text{and} \quad \vdash (A_1 \wedge A_2 \wedge \cdots \wedge A_n) \rightarrow B$$

Rules

$$\frac{}{A \vdash A}\, I \qquad \frac{\Gamma \vdash A \qquad \Gamma \vdash B}{\Gamma \vdash A \wedge B}\, \wedge\mathrm{R}$$

$$\frac{\Gamma \vdash A \qquad \Sigma, A \vdash B}{\Gamma, \Sigma \vdash B}\, Cut \qquad \frac{\Gamma, A \vdash C}{\Gamma, A \wedge B \vdash C}\, \wedge\mathrm{L}_1$$

Classical Sequent Calculus

Sequents

Judgements are of the following form:

$$A_1, A_2, \ldots, A_n \vdash B_1, B_2, \ldots, B_k$$

Logical Interpretation

This can be expressed as:

$$\vdash (A_1 \wedge A_2 \wedge \cdots \wedge A_n) \rightarrow (B_1 \vee B_2 \vee \cdots \vee B_k)$$

or, equivalently,

$$\vdash \neg A_1 \vee \neg A_2 \vee \cdots \vee \neg A_n \vee B_1 \vee B_2 \vee \cdots \vee B_k$$

Some Rules for CL Sequent Calculus

Rules

$$\frac{}{A \vdash A}\; I \qquad \frac{\Gamma \vdash \Delta, A \qquad A, \Sigma \vdash \Pi}{\Gamma, \Sigma \vdash \Delta, \Pi}\; Cut$$

$$\frac{\Gamma \vdash A, \Delta \qquad \Gamma \vdash B, \Delta}{\Gamma \vdash A \wedge B, \Delta}\; \wedge \mathrm{R} \qquad \frac{\Gamma, A \vdash \Delta}{\Gamma, A \wedge B \vdash \Delta}\; \wedge L_1$$

$$\frac{\Gamma \vdash A, \Delta}{\Gamma, \neg A \vdash \Delta}\; \neg L \qquad \frac{\Gamma, A \vdash \Delta}{\Gamma \vdash \neg A, \Delta}\; \neg R$$

Introduction to Linear Logic

- Only use assumptions exactly *once*
- In traditional logic we can prove:

$$\vdash A \wedge A \rightarrow A \qquad \vdash A \wedge B \rightarrow A \qquad \vdash A \rightarrow A \wedge A$$

- In linear logic, we cannot prove:

$$\nvdash A \otimes A \multimap A \qquad \nvdash A \otimes B \multimap A \qquad \nvdash A \multimap A \otimes A$$

- But we *can* prove:

$$\vdash A \otimes A \multimap A \otimes A \qquad \vdash A \otimes B \multimap B \otimes A$$

Symbols in Linear Logic

Usage	Name	Meaning
$A \multimap B$	Lollipop	Linear implication: "Consume A to produce B"
$A \otimes B$	Times	Multiplicative conjunction: "Both A and B"
$A \parr B$	Par	Multiplicative disjunction: "A or B"
$!A$	"Of course"	Exponential: "A may be used multiple times"
$?A$	"Why-not"	Dual exponential: "A may be produced (proven) multiple times"

Linear Negation

Neutral elements

- $\mathbf{1}$ unit for $\otimes$: absence of any recources
- $\bot$ unit for $\parr$: unconsumable recourses

Definitions

$$A \multimap B := A^{\bot} \parr B$$
$$\mathbf{1} \otimes A \equiv A$$
$$\bot \parr A \equiv A$$

$(\cdot)^{\bot}(nil)$

$$\mathbf{1}^{\bot} := \bot \qquad (A \otimes B)^{\bot} := A^{\bot} \parr B^{\bot} \qquad (!A)^{\bot} := ?A^{\bot}$$
$$\bot^{\bot} := \mathbf{1} \qquad (A \parr B)^{\bot} := A^{\bot} \otimes B^{\bot} \qquad (?A)^{\bot} := !A^{\bot}$$

Sequents in Linear Logic

- Γ: unrestricted context
- Δ: restricted context
- C: single proposition

Intuitionistic Linear Logic

Judgements are given by:

$$\Gamma; \Delta \vdash C$$

Classical Linear Logic

Judgements are given by:

$$\vdash \Gamma; \Delta$$

Intuitionistic Linear Logic (ILL)

Rules

$$\frac{}{\Gamma; A \vdash A}\ \textsc{Id} \qquad \frac{\Gamma; \Delta \vdash A \qquad \Gamma; \Delta', A \vdash C}{\Gamma; \Delta, \Delta' \vdash C}\ \textsc{Cut}$$

$$\frac{\Gamma; \Delta, A, B \vdash C}{\Gamma; \Delta, A \otimes B \vdash C}\ \otimes\mathrm{L} \qquad \frac{\Gamma; \Delta \vdash A \qquad \Gamma; \Delta' \vdash B}{\Gamma; \Delta, \Delta' \vdash A \otimes B}\ \otimes\mathrm{R}$$

Classical Linear Logic (CLL)

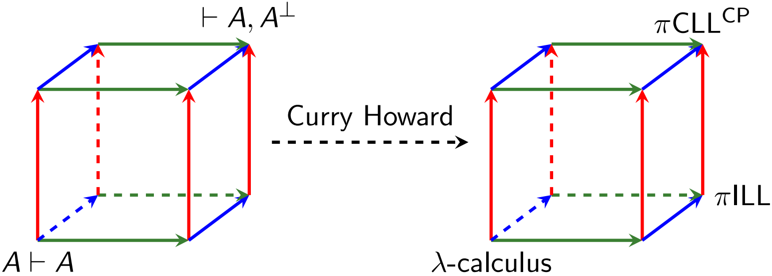
Rules

$$\frac{}{\vdash \Gamma; A, A^{\perp}}\ \textsc{Id} \qquad \frac{\vdash \Gamma; \Delta, A \qquad \vdash \Gamma; \Delta', A^{\perp}}{\vdash \Gamma; \Delta, \Delta'}\ \textsc{Cut}$$

$$\frac{\vdash \Gamma; \Delta, A \qquad \vdash \Gamma; \Delta', B}{\vdash \Gamma; \Delta, \Delta', A \otimes B}\ \otimes$$

Table of Contents

Logic Cube: Curry Howard



- Intuitionistic $\longrightarrow$ Classical
- Traditional $\longrightarrow$ Linear
- Natural Deduction $\longrightarrow$ Sequent Calculus

Curry-Howard correspondence

Traditional Logic	**λ-calculus**
Propositions	Types
Proofs	Programs
Normalization of proofs	Evaluation of programs

Linear Logic	**π-calculus**
Propositions	Session types
Proofs	Processes
Cut elimination	Communication

Introduction to the π-calculus

Session types of the π-calculus

Session types in the π-calculus:

$$A \quad ::= \quad \mathbf{1} \mid \bot \mid A \otimes B \mid A \parr B \mid A \multimap B \mid\ !A \mid\ ?A$$

Duality in channels

- If A is sent, $A^{\perp}$ is received
- If $A^{\perp}$ is sent, $(A^{\perp})^{\perp} = A$ is received

Par and times

Times

$A \otimes B$: Output a channel of session type A, then behave as a channel of session type B

Par

$A \parr B$: Dual of $A \otimes B$, input A and behave as B

Example

If one side of a channel behaves as $A \otimes B$, the other side behaves as $A^\perp \parr B^\perp$

$$(A \otimes B)^\perp = A^\perp \parr B^\perp$$

Server Accept and Client Request

Server Accept

$!A$: Repeatedly provide a service of type A. Accepts (receives) input of type A

Client Request

$?A$: Connect to a service of type A. Request a service (sends) output of type A

Terms of the π-calculus

Terms

$$P, Q \quad ::= \quad \mathbf{0} \mid [x \leftrightarrow y] \mid (\nu x).P \mid P|Q \mid x\langle y\rangle.P \mid x(y).P \mid$$
$$!x(y).P \mid x\langle\rangle.\mathbf{0} \mid x().P$$

Forwarder and Parallel Composition

- Process $[w \leftrightarrow x]$ "fuses" channels x and y
- Input sent along w is sent as output along x
- Input sent along x is sent as output along w

Rule

$$\frac{}{[w \leftrightarrow x] \vdash \Gamma;\ w : A^{\perp},\ x : A} \textsc{Id}$$

Forwarder and Parallel Composition

- Process $[w \leftrightarrow x]$ “fuses” channels x and y
- Input sent along w is sent as output along x
- Input sent along x is sent as output along w

Rule

$$\frac{}{[w \leftrightarrow x] \vdash \Gamma;\ w : A^{\perp},\ x : A}\ \textsc{Id}$$

Parallel composition

- $P|Q$: do processes P and Q concurrently

Channel creation

λ-calculus	**π-calculus**
$\lambda x.M$ x bound in M	$(\nu x).P$ x bound in P

Cut rule

$$\frac{P \vdash \Gamma;\ \Delta,\ x : A \qquad Q \vdash \Gamma;\ \Delta',\ x : A^{\perp}}{(\nu x)(P|Q) \vdash \Gamma;\ \Delta,\ \Delta'}\ \textsc{Cut}$$

Input

- $x(y).P$: receive a channel on channel x, bind it to y and behave as P
- $!x(y).P$: receive on channel x and bind result to y, execute P and repeat.

Rules

$$\frac{P \vdash \Gamma;\ \Delta,\ y : A,\ x : B}{x(y).P \vdash \Gamma;\ \Delta,\ x : A \parr B}\ \parr \qquad \frac{P \vdash \Gamma;\ y : A}{!x(y).P \vdash \Gamma;\ x : !A}\ !$$

Output

- $x\langle y\rangle.P$: send y over channel x and behave as P
- Sending multiple times?

Rules

$$\frac{P \vdash \Gamma,\ y : A \qquad Q \vdash \Delta,\ x : B}{\nu y.x\langle y\rangle.(P|Q) \vdash \Gamma,\ \Delta,\ x : A \otimes B} \otimes$$

$$\frac{P \vdash \Gamma,\ u : A;\ \Delta}{P\{x/u\} \vdash \Gamma;\ \Delta,\ x : ?A} ?$$

Closing a channel

$\mathbf{0}$

- $\mathbf{0}$: process construct for inaction
- Similar to *nil* for lists or O for nat

Closing x

$$x\langle\rangle.\mathbf{0} \mid x().Q \rightarrow Q$$

Rules

$$\frac{P \vdash \Gamma; \Delta}{x().P \vdash \Gamma; \Delta,\ x : \bot}\ \bot \qquad \frac{}{x\langle\rangle.\mathbf{0} \vdash \Gamma;\ x : \mathbf{1}}\ \mathbf{1}$$

Findings first paper

CP

- Wadler uses $\vdash !\Gamma, \Delta$ instead of $\vdash \Gamma; \Delta$
- Judgements in CP look like $P \vdash !\Gamma, \Delta$

GV

- Based on the language by Gay and Vasconcelos
- A session-typed functional language
- Translation into CP

Table of Contents

Sequents in πCLL^{CP}, πILL and πULL

Sequents

- Unrestricted context: Γ
- Restricted (Linear contexts): Δ and Λ

πCLL^{CP}

$P \vdash \Gamma \,;\, \Delta$

πILL

$\Gamma \,;\, \Delta \vdash P :: x : A$

πULL

$\Gamma \,;\, \Delta \vdash P :: \Lambda$

Differences in the identity rule

ID πCLL$^{\mathsf{CP}}$

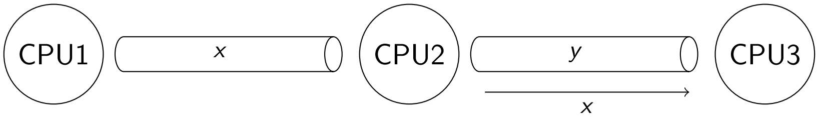
$$\frac{}{[x \leftrightarrow y] \vdash \Gamma \; ; \; x : A, \; y : A^{\perp}} \text{ID}$$

ID π-ILL

$$\frac{}{\Gamma \; ; \; x : A \vdash [x \leftrightarrow y] :: \; y : A} \text{ID}$$

Locality Principle

- "*received channels cannot be used for further reception, i.e., only the output capability of channels can be sent*"
- "*received channels cannot be used to provide a service*"



Question

What should happen when CPU1 sends a message on channel x?

Term that violates locality

Let $C = (\nu x)(x(y).!y(z).P_x|(\nu q)x\langle q\rangle.(Q_q|R_x))$

Prooftree $\pi\mathrm{CLL}^{\mathrm{CP}}$

$$(\nu x)(x(y).!y(z).P_x|(\nu q)x\langle q\rangle.(Q_q|R_x)) \vdash \cdot\ ;\ \cdot$$

Term that violates locality

Let $C = (\nu x)(x(y).!y(z).P_x | (\nu q)x\langle q\rangle.(Q_q | R_x))$

Prooftree πCLL^{CP}

$$\dfrac{x(y).!y(z).P_x \vdash \cdot\ ;\ x : (!A) ⅋ (?B) \qquad t_1}{(\nu x)(x(y).!y(z).P_x | (\nu q)x\langle q\rangle.(Q_q | R_x)) \vdash \cdot\ ;\ \cdot}\ \textsc{Cut}$$

Term that violates locality

Let $C = (\nu x)(x(y).!y(z).P_x | (\nu q)x\langle q\rangle.(Q_q | R_x))$

Prooftree πCLL^{CP}

$$\dfrac{\dfrac{!y(z).P_x \vdash \cdot\ ;\ y : !A, x : ?B}{x(y).!y(z).P_x \vdash \cdot\ ;\ x : (!A) \parr (?B)}\ \parr \qquad t_1}{(\nu x)(x(y).!y(z).P_x | (\nu q)x\langle q\rangle.(Q_q | R_x)) \vdash \cdot\ ;\ \cdot}\ \textsc{Cut}$$

Term that violates locality

Let $C = (\nu x)(x(y).!y(z).P_x | (\nu q)x\langle q\rangle.(Q_q | R_x))$

Prooftree πCLL^{CP}

$$\dfrac{\dfrac{\dfrac{!y(z).P'_x \vdash u : B \; ; \; y : !A}{!y(z).P_x \vdash \cdot \; ; \; y : !A, x : ?B}\;?}{x(y).!y(z).P_x \vdash \cdot \; ; \; x : (!A) \parr (?B)}\;\parr \qquad t_1}{(\nu x)(x(y).!y(z).P_x | (\nu q)x\langle q\rangle.(Q_q | R_x)) \vdash \cdot \; ; \; \cdot}\;\textsc{Cut}$$

$P_x = P'_x\{x/u\}$

Term that violates locality

Let $C = (\nu x)(x(y).!y(z).P_x | (\nu q)x\langle q\rangle.(Q_q | R_x))$

Prooftree $\pi\text{CLL}^{\text{CP}}$

$$\dfrac{\dfrac{\dfrac{\dfrac{P'_x \vdash u : B \,;\, z : A}{!y(z).P'_x \vdash u : B \,;\, y : !A}\,!}{!y(z).P_x \vdash \cdot \,;\, y : !A, x : ?B}\,?}{x(y).!y(z).P_x \vdash \cdot \,;\, x : (!A) \parr (?B)}\,\parr \qquad t_1}{(\nu x)(x(y).!y(z).P_x | (\nu q)x\langle q\rangle.(Q_q | R_x)) \vdash \cdot \,;\, \cdot}\,\textsc{Cut}$$

$P_x = P'_x\{x/u\}$

Term that violates locality

Let $C = (\nu x)(x(y).!y(z).P_x | (\nu q)x\langle q\rangle.(Q_q | R_x))$

Prooftree $\pi\mathrm{CLL}^{\mathrm{CP}}$

$$
\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{t_2}{P'_x \vdash u : B \; ; \; z : \; A}\;!}{!y(z).P'_x \vdash u : B \; ; \; y : !A}\;?}{!y(z).P_x \vdash \cdot \; ; \; y : !A, x : ?B}\;⅋}{x(y).!y(z).P_x \vdash \cdot \; ; \; x : (!A) ⅋ (?B)} \quad t_1}{(\nu x)(x(y).!y(z).P_x | (\nu q)x\langle q\rangle.(Q_q | R_x)) \vdash \cdot \; ; \; \cdot}\;\textsc{Cut}
$$

$P_x = P'_x\{x/u\}$

C is not Typeable in πILL

Attempt 1 with $\parr$R rule

Prooftree

$$\cdot\ ;\ \cdot \vdash (\nu x)(x(y).!y(z).P_x | (\nu q)x\langle q\rangle.(Q_q | R_x)) :: w : C$$

C is not Typeable in πILL

Attempt 1 with $\parr$R rule

Prooftree

$$\dfrac{\cdot\,;\,\cdot \vdash x(y).!y(z).P_x :: x : A \parr B \qquad t_1}{\cdot\,;\,\cdot \vdash (\nu x)(x(y).!y(z).P_x | (\nu q)x\langle q\rangle.(Q_q | R_x)) :: w : C}\ \textsc{Cut}$$

C is not Typeable in πILL

Attempt 1 with $⅋\mathrm{R}$ rule

Prooftree

$$\dfrac{\dfrac{\cdot\ ;\ y : A^{\perp} \vdash\ !y(z).P_x :: x : B}{\cdot\ ;\ \cdot \vdash x(y).!y(z).P_x :: x : A ⅋ B}\ ⅋\mathrm{R} \qquad t_1}{\cdot\ ;\ \cdot \vdash (\nu x)(x(y).!y(z).P_x | (\nu q)x\langle q\rangle.(Q_q | R_x)) :: w : C}\ \textsc{Cut}$$

C is not Typeable in πILL

Attempt 1 with $⅋\mathrm{R}$ rule

Prooftree

$$\dfrac{\dfrac{\dfrac{}{\cdot\,;\, y : A^{\perp} \vdash\, !y(z).P_x :: x : B}}{\cdot\,;\,\cdot \vdash x(y).!y(z).P_x :: x : A ⅋ B}\,⅋\mathrm{R} \qquad t_1}{\cdot\,;\,\cdot \vdash (\nu x)(x(y).!y(z).P_x | (\nu q)x\langle q\rangle.(Q_q | R_x)) :: w : C}\,\textsc{Cut}$$

- Channel y ends up on the left of the turnstile
- No rule to define a service on y

C is not Typeable in πILL

Attempt 2 with $\otimes$L rule

Prooftree

$$\cdot\, ; \cdot \vdash (\nu x)((\nu q)x\langle q\rangle.(Q_q|R_x)|x(y).!y(z).P_x) :: w : C$$

C is not Typeable in πILL

Attempt 2 with $\otimes$L rule

Prooftree

$$\dfrac{t_1 \qquad \cdot\,;\, x : A^\perp \otimes B^\perp \vdash x(y).!y(z).P_x :: w : C}{\cdot\,;\,\cdot \vdash (\nu x)((\nu q)x\langle q\rangle.(Q_q|R_x)|x(y).!y(z).P_x) :: w : C}\ \textsc{Cut}$$

C is not Typeable in πILL

Attempt 2 with $\otimes$L rule

Prooftree

$$\dfrac{t_1 \qquad \dfrac{\cdot\ ;\ y:\ A^{\perp}, x : B^{\perp} \vdash\ !y(z).P_x\ ::\ w : C}{\cdot\ ;\ x : A^{\perp} \otimes B^{\perp} \vdash x(y).!y(z).P_x\ ::\ w : C}\ \otimes\mathrm{L}}{\cdot\ ;\ \cdot \vdash (\nu x)((\nu q)x\langle q\rangle.(Q_q|R_x)|x(y).!y(z).P_x)\ ::\ w : C}\ \textsc{Cut}$$

C is not Typeable in πILL

Attempt 2 with $\otimes$L rule

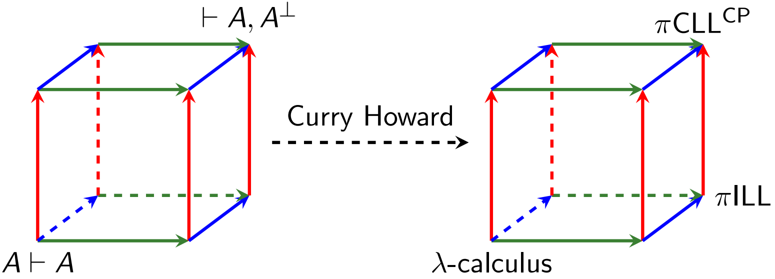
Prooftree

$$\dfrac{t_1 \qquad \dfrac{\dfrac{}{\cdot\ ;\ y : A^{\perp}, x : B^{\perp} \vdash\ !y(z).P_x :: w : C}}{\cdot\ ;\ x : A^{\perp} \otimes B^{\perp} \vdash x(y).!y(z).P_x :: w : C}\ \otimes\mathrm{L}}{\cdot\ ;\ \cdot \vdash (\nu x)((\nu q)x\langle q\rangle.(Q_q|R_x)|x(y).!y(z).P_x) :: w : C}\ \textsc{Cut}$$

- Again, channel y ends up on the left of the turnstile
- No rule to define a service on y

Table of Contents

Logic Cube: Curry Howard



- Intuitionistic $\longrightarrow$ Classical
- Traditional $\longrightarrow$ Linear
- Natural Deduction $\longrightarrow$ Sequent Calculus

Questions

Are there any questions?