Comparing Session Type Interpretations of Linear Logic*

Bas van den Heuvel and Jorge A. Pérez

Bernoulli Institute for Math, CS, and AI
University of Groningen

TYPES 2020

*Based on work to appear in the proceedings of PLACES 2020

Introduction

- Curry-Howard for concurrency:
  Linear Logic as Session Type system for the π-calculus
- Two logics, two interpretations:
  Classical and *Intuitionistic* Linear Logic (CLL resp. ILL)
- Logic ensures several correctness properties, so studying the fundamentals of these type systems is essential
- Can explain the many derived works on both interpretations

Introduction

- Urgent open question: how do the interpretations relate?
- Exists informal observation by Caires, Pfenning, Toninho; might be an answer
- Our work formalizes the observation, by developing a common ground for comparison called United Linear Logic (ULL)

Propositions as types

Sequential Curry-Howard (simply-typed λ-calculus)

Intuitionistic logic propositions	$\leftrightarrow$	Types
Natural deduction	$\leftrightarrow$	Type inference
Cut reduction	$\leftrightarrow$	Computation

Concurrent Curry-Howard (typed π-calculus)

Linear logic propositions	$\leftrightarrow$	Session types
Sequent calculus	$\leftrightarrow$	Type inference
Cut reduction	$\leftrightarrow$	Communication

Curry-Howard for concurrency

Session types type channels	———————	dually on opposite sides

Session type	**Linear logic**	**Meaning**	**π-calculus**
`end`	$\mathbf{1}$ and $\perp$	Termination	$x[].\mathbf{0}$ and $x().P$
$!A.B$	$A \otimes B$	Send A, continue as B	$x[y].P$
$?A.B$	$A \parr B$ and $A \multimap B$	Receive A, continue as B	$x(y).P$

Communication:

$$(\nu x)(x[y].P \mid x(z).Q) \to (\nu x)(\nu y)(P \mid Q\{y/z\})$$

Our approach

- Not comparing logics, but type systems: expressivity
- A type system induces a *class of typable processes*:
  We want to compare the classes of CLL- and ILL-typable processes
- Problem: there is no *common ground*
- Our solution: United Linear Logic (ULL),
  based on Girard's Logic of Unity
- ULL subsumes CLL and ILL
- Comparison of classes of processes typable in CLL, ILL and ULL

Towards United Linear Logic

Classical:

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

$$Q \vdash \Delta', x : A^\perp$$

Intuitionistic:

$$\Delta \vdash P :: x : A \qquad \Delta', x : A \vdash Q :: y : B$$

United:

$$\Delta \vdash P :: \Lambda, x : A \qquad \Delta', x : A \vdash Q :: \Lambda', y : B$$

$$\Delta'', x : A^\perp \vdash R :: \Lambda''$$

Towards United Linear Logic

Combination of rules for input $x(y).P$ in ULL:

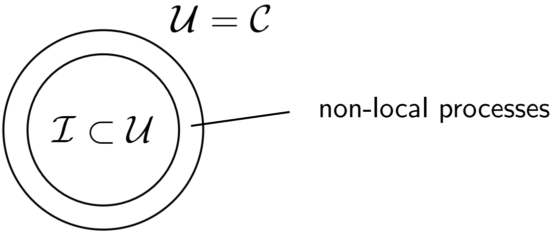
- CLL has one rule: $A \parr B$
- ILL has two: $A \multimap B$ on the right, and $A \otimes B$ on the left
- ULL has three: $A \parr B$ and $A \multimap B$ on the right, and $A \otimes B$ on the left

Other inference rules are similarly combined, making sure that ULL subsumes CLL and ILL

Results

Comparison between three classes of processes:

- $\mathcal{C}$: CLL-typable processes
- $\mathcal{I}$: ILL-typable processes
- $\mathcal{U}$: ULL-typable processes



Results: locality

- Locality (Merro): no input is allowed on received channels
- Well-known principle in process calculi that model programming languages
- ILL enforces locality for shared channels, CLL does not
- Shared names can indefinitely receive channels to perform a service on

$$!x(y).P \to P \mid !x(y).P$$

Typable in ILL and CLL:

$$x(y).y(z).P$$

Not typable in ILL:

$$x(y).!y(z).P$$

Conclusion and Current Work

- Comparison of session type interpretations of CLL and ILL
- ULL: fair ground for comparison
- Result: CLL is strictly more expressive than ILL
  - ULL is two-sided CLL
  - ULL with restriction on the right is ILL
  - ULL and CLL are symmetrical, ILL is not
- Due to asymmetry, ILL enforces locality, CLL does not
- Current work: use ULL to transfer extensions of CLL to ILL and vice versa