# a logical account of subtyping for session types

Ross Horne, University of Luxembourg    Luca Padovani, University of Camerino

PLACES, 22nd april 2023

# subtyping for session types

Gay and Hole [2005] and others

$$\frac{\forall i \in I : A_i \leqslant B_i \qquad I \subseteq J}{\oplus\{\ell_i : A_i\}_{i \in I} \leqslant \oplus\{\ell_j : B_j\}_{j \in J}} \qquad \frac{\forall j \in J : A_j \leqslant B_j \qquad J \subseteq I}{\&\{\ell_i : A_i\}_{i \in I} \leqslant \&\{\ell_j : B_j\}_{j \in J}}$$

- if $A \leqslant B$, a process that behaves as $A$ can be used where a process that behaves as $B$ is expected
- direction of $\leqslant$ may vary depending on viewpoint [Gay, 2016]
- $\&$ and $\oplus$ are $n$-**ary operators**

# propositions as protocols

Caires et al. [2016], Wadler [2014], Lindley and Morris [2016] and others

| | | |
|---|---|---|
| **linear logic propositions** | $\Longleftrightarrow$ | **session types** |
| **linear logic proofs** | $\Longleftrightarrow$ | **well-typed processes** |
| **cut reduction** | $\Longleftrightarrow$ | **communication** |

1. Can we define a non-trivial subtyping in this logical setting, where $\&$ **and** $\oplus$ **have fixed arity**?
2. If so, is there anything "special" about subtyping in this setting that has not occurred elsewhere?

# propositions as protocols

Caires et al. [2016], Wadler [2014], Lindley and Morris [2016] and others

| | | |
|---|---|---|
| **linear logic propositions** | $\Longleftrightarrow$ | **session types** |
| **linear logic proofs** | $\Longleftrightarrow$ | **well-typed processes** |
| **cut reduction** | $\Longleftrightarrow$ | **communication** |

1. Can we define a non-trivial subtyping in this logical setting, where $\&$ **and** $\oplus$ **have fixed arity**? **Yes**
2. If so, is there anything “special” about subtyping in this setting that has not occurred elsewhere? **Yes**

# types and typing rules (finite case)

$$A, B ::= \mathbf{0} \mid \top \mid \overbrace{\mathbf{1} \mid \bot}^{\text{termination}} \mid \overbrace{A \oplus B \mid A \,\&\, B}^{\text{selection}} \mid \overbrace{A \otimes B \mid A \parr B}^{\text{delegation}}$$

# types and typing rules (finite case)

$$A, B ::= \overbrace{\mathbf{0} \mid \top}^{???} \mid \overbrace{\mathbf{1} \mid \bot}^{\text{termination}} \mid \overbrace{A \oplus B \mid A \mathbin{\&} B}^{\text{selection}} \mid \overbrace{A \otimes B \mid A \mathbin{⅋} B}^{\text{delegation}}$$

# types and typing rules (finite case)

$$A, B ::= \overbrace{\mathbf{0} \mid \top}^{???} \mid \overbrace{\mathbf{1} \mid \bot}^{\text{termination}} \mid \overbrace{A \oplus B \mid A \mathbin{\&} B}^{\text{selection}} \mid \overbrace{A \otimes B \mid A \parr B}^{\text{delegation}}$$

**no rule for 0**

$$\frac{}{\textsf{fail}\ x \vdash \Gamma, x : \top}$$

# subtyping (finite case)

reflexivity

$$\frac{}{\kappa \leqslant \kappa}$$

bottom

$$\frac{}{\mathbf{0} \leqslant A}$$

top

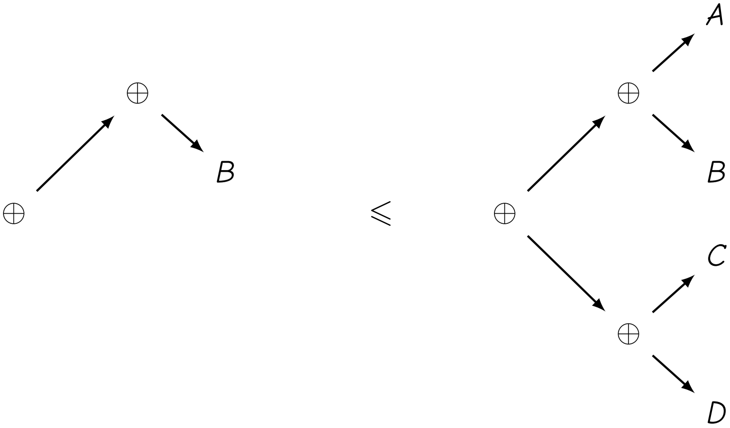
$$\frac{}{A \leqslant \top}$$

precongruence

$$\frac{A \leqslant A' \qquad B \leqslant B'}{A \star B \leqslant A' \star B'}$$

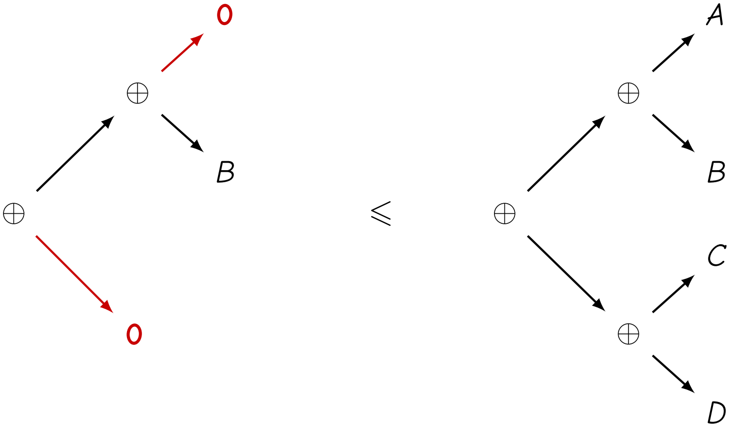
**that's it!**

# example of subtyping with $n$-ary $\oplus$



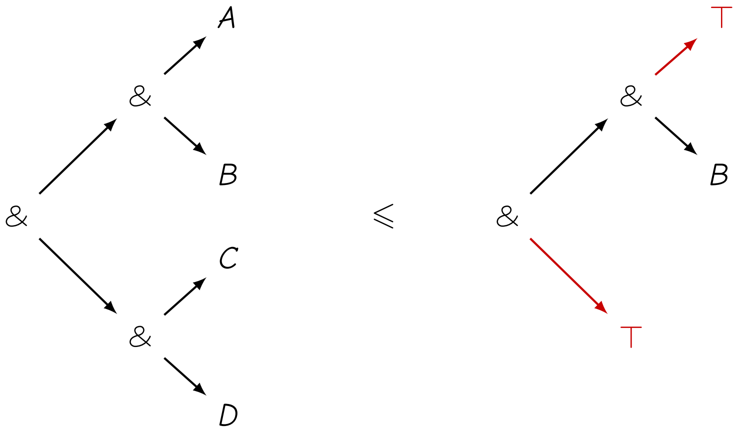

$$\oplus\{\mathsf{left} : \oplus\{\mathsf{right} : B\}\} \leqslant \oplus \begin{Bmatrix} \mathsf{left} : \oplus\{\mathsf{left} : A, \mathsf{right} : B\}, \\ \mathsf{right} : \oplus\{\mathsf{left} : C, \mathsf{right} : D\} \end{Bmatrix}$$

# example of subtyping with **binary** $\oplus$




$$(\mathbf{0} \oplus B) \oplus \mathbf{0} \leqslant (A \oplus B) \oplus (C \oplus D)$$

# example of subtyping with binary &




$$(A \mathbin{\&} B) \mathbin{\&} (C \mathbin{\&} D) \leqslant (\top \mathbin{\&} B) \mathbin{\&} \top$$

# compatibility = cut + subtyping

$$\frac{P \vdash \Gamma, x : A \qquad Q \vdash \Delta, x : B}{(x)(P \mid Q) \vdash \Gamma, \Delta} \; A \leqslant B^{\perp}$$

**We are changing a key proof rule, how do we know it's sound?**

# a coercion semantics of subtyping

If $\pi :: A \leqslant B$, then $[\![\pi]\!]_{x,y}$ is a (cut-free) **process** that "translates" protocol $A$ (from $x$) into protocol $B$ (on $y$)

$$\left[\!\!\left[\frac{}{\mathbf{1} \leqslant \mathbf{1}}\right]\!\!\right]_{x,y} = x().y[] \qquad \left[\!\!\left[\frac{}{\mathbf{0} \leqslant A}\right]\!\!\right]_{x,y} = \mathsf{fail}\ x \qquad \left[\!\!\left[\frac{}{A \leqslant \top}\right]\!\!\right]_{x,y} = \mathsf{fail}\ y$$

$$\left[\!\!\left[\frac{\pi_1 :: A \leqslant A' \qquad \pi_2 :: B \leqslant B'}{A \oplus B \leqslant A' \oplus B'}\right]\!\!\right]_{x,y} = \mathsf{case}\ x \left\{ \begin{array}{l} y[\mathsf{left}].\ [\![\pi_1]\!]_{x,y} \\ y[\mathsf{right}].\ [\![\pi_2]\!]_{x,y} \end{array} \right\}$$

**Theorem**

*If* $\pi :: A \leqslant B$ *then* $[\![\pi]\!]_{x,y} \vdash x : A^{\perp}, y : B$

# from compatibility back to cut

$$\frac{P \vdash \Gamma, x : A \qquad Q \vdash \Delta, x : B}{(x)(P \mid Q) \vdash \Gamma, \Delta}\ \pi :: A \leqslant B^{\perp}$$

$$\Downarrow$$

$$\frac{\dfrac{P\{y/x\} \vdash \Gamma, y : A \qquad [\![\pi]\!]_{y,x} \vdash y : A^{\perp}, x : B^{\perp}}{(y)(P\{y/x\} \mid [\![\pi]\!]_{y,x}) \vdash \Gamma, x : B^{\perp}} \qquad Q \vdash \Delta, x : B}{(x)((y)(P\{y/x\} \mid [\![\pi]\!]_{y,x}) \mid Q) \vdash \Gamma}$$

# types and typing rules (general case)

We use $\mu$MALL$^\infty$ [Baelde et al., 2016], linear logic with fixed points

$$A, B ::= \underbrace{\mathbf{0} \mid \top}_{\text{subtyping}} \mid \overbrace{\mathbf{1} \mid \bot}^{\text{termination}} \mid \overbrace{A \oplus B \mid A \,\&\, B}^{\text{selection}} \mid \overbrace{A \otimes B \mid A \parr B}^{\text{delegation}} \mid \overbrace{\mu X.A \mid \nu X.A}^{\text{recursion}}$$

In $\mu$MALL$^\infty$ fixed points are simply **unfolded**

$$\frac{P \vdash \Gamma, x : A\{\sigma X.A/X\}}{P \vdash \Gamma, x : \sigma X.A}\ \sigma \in \{\mu, \nu\}$$

- typing derivations may be infinite

# types and typing rules (general case)

We use $\mu$MALL$^\infty$ [Baelde et al., 2016], linear logic with fixed points

$$A, B ::= \underbrace{\mathbf{0} \mid \top}_{\text{subtyping}} \mid \overbrace{\mathbf{1} \mid \bot}^{\text{termination}} \mid \overbrace{A \oplus B \mid A \& B}^{\text{selection}} \mid \overbrace{A \otimes B \mid A \parr B}^{\text{delegation}} \mid \overbrace{\mu X.A \mid \nu X.A}^{\text{recursion}}$$

In $\mu$MALL$^\infty$ fixed points are simply **unfolded**

$$\frac{P \vdash \Gamma, x : A\{\sigma X.A/X\}}{P \vdash \Gamma, x : \sigma X.A} \; \sigma \in \{\mu, \nu\}$$

- typing derivations may be infinite, but **not all are valid**!
- in a nutshell: every infinite branch of a proof must unfold a greatest fixed point infinitely many times
- this is a *rough simplification*, see paper and $\mu$MALL$^\infty$

# subtyping (general case)

reflexivity

$$\frac{}{\kappa \leqslant \kappa}$$

bottom

$$\frac{}{\mathbf{0} \leqslant A}$$

top

$$\frac{}{A \leqslant \top}$$

precongruence

$$\frac{A \leqslant A' \qquad B \leqslant B'}{A \star B \leqslant A' \star B'}$$

$$\frac{A\{\sigma X.A/X\} \leqslant B}{\sigma X.A \leqslant B}$$

unfold on the left

$$\frac{A \leqslant B\{\sigma X.B/X\}}{A \leqslant \sigma X.B}$$

unfold on the right

- subtyping derivations may be infinite
- without restrictions, the two fixed points would be equivalent

# the validity condition for subtyping derivations*

*again, this is a rough simplification

A subtyping derivation is **valid** provided that every infinite branch of the derivation contains either

- infinitely many unfoldings of a $\mu$ to the **left** of $\leqslant$, or
- infinitely many unfoldings of a $\nu$ to the **right** of $\leqslant$

## Intuition

“Small” protocols (least fixed point) can be subsumed by “large” protocols (greatest fixed point), but **not the other way around**

## More formally

A (valid) subtyping derivation must result into a **well-typed coercion**

# an example of invalid subtyping

$$\dfrac{\dfrac{\dfrac{}{\mathbf{1} \leqslant \mathbf{1}} \qquad \dfrac{\vdots}{\nu X.(\mathbf{1} \oplus X) \leqslant \mu X.(\mathbf{1} \oplus X)}}{\mathbf{1} \oplus \nu X.(\mathbf{1} \oplus X) \leqslant \mathbf{1} \oplus \mu X.(\mathbf{1} \oplus X)}}{\nu X.(\mathbf{1} \oplus X) \leqslant \mu X.(\mathbf{1} \oplus X)}$$

- there is only one infinite branch
- infinitely many unfoldings of a $\nu$ to the left of $\leqslant$, and
- infinitely many unfoldings of a $\mu$ to the right of $\leqslant$

# invalid subtyping: what can possibly go wrong?

$$P(x) \triangleq x[\mathsf{right}].P\langle x\rangle \qquad Q(x,y) \triangleq \mathsf{case}\ x\{x().y[], Q\langle x,y\rangle\}$$

$$A \stackrel{\text{def}}{=} \nu X.(1 \oplus X) \qquad B \stackrel{\text{def}}{=} \nu X.(\bot \mathbin{\&} X)$$

$$\dfrac{\dfrac{\vdots}{P\langle x\rangle \vdash x : A} \qquad \dfrac{\vdots}{Q\langle x,y\rangle \vdash x : B, y : \mathbf{1}}}{(x)(P\langle x\rangle \mid Q\langle x,y\rangle) \vdash y : \mathbf{1}}\ A \leqslant B^{\bot}$$

## Moral

- the validity condition makes $\leqslant$ termination preserving

# concluding remarks

## Results

- **termination-preserving** subtyping when $\oplus/\&$ have **fixed arity**
- the axioms $\mathbf{0} \leqslant A$ and $A \leqslant \top$ are all we need to express differences in the branching structure of session types

## Future work

- investigate principal typing
- consider more interesting coercions
  *e.g.* to capture asynchronous subtyping [Ghilezan et al., 2022]

# concluding remarks

## Results

- **termination-preserving** subtyping when $\oplus/\&$ have **fixed arity**
- the axioms $\mathbf{0} \leqslant A$ and $A \leqslant \top$ are all we need to express differences in the branching structure of session types

## Future work

- investigate principal typing
- consider more interesting coercions
  *e.g.* to capture asynchronous subtyping [Ghilezan et al., 2022]

**thank you**

# references

David Baelde, Amina Doumane, and Alexis Saurin. Infinitary proof theory: the multiplicative additive case. In Jean-Marc Talbot and Laurent Regnier, editors, *25th EACSL Annual Conference on Computer Science Logic, CSL 2016, August 29 - September 1, 2016, Marseille, France*, volume 62 of *LIPIcs*, pages 42:1–42:17. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2016.

Luís Caires, Frank Pfenning, and Bernardo Toninho. Linear logic propositions as session types. *Math. Struct. Comput. Sci.*, 26(3): 367–423, 2016.

Simon J. Gay. Subtyping supports safe session substitution. In Sam Lindley, Conor McBride, Philip W. Trinder, and Donald Sannella, editors, *A List of Successes That Can Change the World - Essays Dedicated to Philip Wadler on the Occasion of His 60th Birthday*, volume 9600 of *Lecture Notes in Computer Science*, pages 95–108. Springer, 2016.

# references (cont.)


Simon J. Gay and Malcolm Hole. Subtyping for session types in the pi calculus. *Acta Informatica*, 42(2-3):191–225, 2005.

Silvia Ghilezan, Jovanka Pantović, Ivan Prokić, Alceste Scalas, and Nobuko Yoshida. Precise subtyping for asynchronous multiparty sessions. *ACM Trans. Comput. Logic*, oct 2022. ISSN 1529-3785. Just Accepted.

Sam Lindley and J. Garrett Morris. Talking bananas: structural recursion for session types. In Jacques Garrigue, Gabriele Keller, and Eijiro Sumii, editors, *Proceedings of the 21st ACM SIGPLAN International Conference on Functional Programming, ICFP 2016, Nara, Japan, September 18-22, 2016*, pages 434–447. ACM, 2016.

Philip Wadler. Propositions as sessions. *J. Funct. Program.*, 24(2-3): 384–418, 2014.