*"It would be interesting to establish a **termination** result for CSLL. This would prove that the resulting calculus does not generate **livelock**. We expect this proof to be somewhat involved..."*

Qian, Kavvos, and Birkedal [2021]

Attacking the Termination Problem for Client-Server Sessions

Luca Padovani

OPCT 2023

sessions and linear logic

Caires and Pfenning [2010], Wadler [2014], Lindley and Morris [2016]

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

proof = well-typed process

$$\dfrac{\dfrac{\vdots}{\vdash \Gamma, \quad A}}{\vdash \Gamma, \quad A \oplus B}[\oplus] \qquad \dfrac{\dfrac{\vdots}{\vdash \Delta, \quad A^\perp} \qquad \dfrac{\vdots}{\vdash \Delta, \quad B^\perp}}{\vdash \Delta, \quad A^\perp \,\&\, B^\perp}[\&]$$

$$\dfrac{\vdash \Gamma, \quad A \oplus B \qquad \vdash \Delta, \quad A^\perp \,\&\, B^\perp}{\vdash \Gamma, \Delta}[\text{cut}]$$

proof = well-typed process

$$
\dfrac{\dfrac{\vdots}{P \vdash \Gamma, x : A}}{\mathsf{left}\, x.P \vdash \Gamma, x : A \oplus B}\,[\oplus]
\qquad
\dfrac{\dfrac{\vdots}{Q \vdash \Delta, x : A^{\perp}} \quad \dfrac{\vdots}{R \vdash \Delta, x : B^{\perp}}}{\mathsf{case}\, x\{Q, R\} \vdash \Delta, x : A^{\perp} \mathbin{\&} B^{\perp}}\,[\&]
$$

$$
\dfrac{\mathsf{left}\, x.P \vdash \Gamma, x : A \oplus B \qquad \mathsf{case}\, x\{Q, R\} \vdash \Delta, x : A^{\perp} \mathbin{\&} B^{\perp}}{(x)(\mathsf{left}\, x.P \mid \mathsf{case}\, x\{Q, R\}) \vdash \Gamma, \Delta}\,[\mathrm{cut}]
$$

proof = well-typed process

$$\dfrac{\dfrac{\vdots}{P \vdash \Gamma, x : A}}{\mathsf{left}\,x.P \vdash \Gamma, x : A \oplus B}\,[\oplus] \qquad \dfrac{\dfrac{\vdots}{Q \vdash \Delta, x : A^{\perp}} \quad \dfrac{\vdots}{R \vdash \Delta, x : B^{\perp}}}{\mathsf{case}\,x\{Q, R\} \vdash \Delta, x : A^{\perp} \,\&\, B^{\perp}}\,[\&]$$

$$\overline{(x)(\mathsf{left}\,x.P \mid \mathsf{case}\,x\{Q, R\}) \vdash \Gamma, \Delta}\;[\mathrm{cut}]$$

$$\downarrow$$

$$\dfrac{\dfrac{\vdots}{P \vdash \Gamma, x : A} \qquad \dfrac{\vdots}{Q \vdash \Delta, x : A^{\perp}}}{(x)(P \mid Q) \vdash \Gamma, \Delta}\,[\mathrm{cut}]$$

soundness of the logic => soundness of typing

The cut rule is admissible

- each application of the cut rule can be **eliminated** after a suitable number of **cut reductions**
- each open session can be terminated after a suitable number of communications

Consequences

⇒ well-typed processes are **deadlock free**

⇒ well-typed processes **terminate**

⇒ well-typed processes are **livelock free**

$$A ::= \bot \mid \top \mid \mathbf{0} \mid \mathbf{1} \mid A \oplus B \mid A \mathbin{\&} B \mid A \otimes B \mid A \mathbin{\rotatebox[origin=c]{180}{\&}} B \mid ?A \mid !A$$

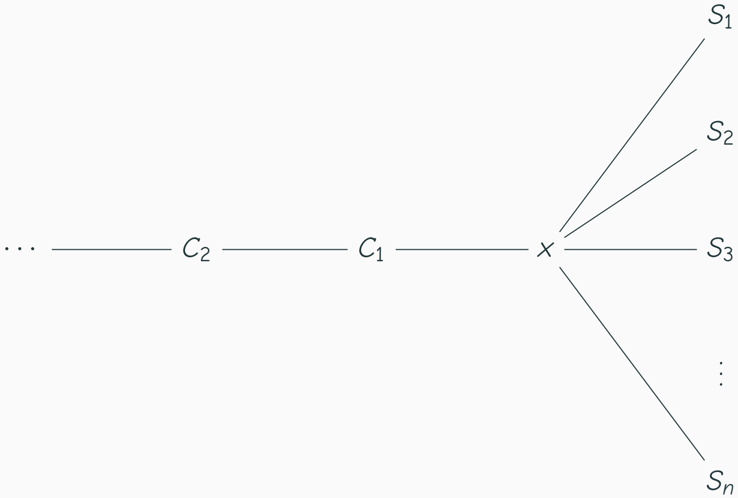
Rules for clients

$$\frac{\vdash \Gamma}{\vdash \Gamma, ?A} \qquad \frac{\vdash \Gamma, A}{\vdash \Gamma, ?A} \qquad \frac{\vdash \Gamma, ?A, ?A}{\vdash \Gamma, ?A}$$

Rule for server

$$\frac{\vdash ?\Gamma, A}{\vdash ?\Gamma, !A}$$

sequential(ized) clients vs unlimited **parallel** instances of server



a problem with this modeling of clients and servers

Lack of accuracy

- availability of **unbounded copies** of the server is unreasonable

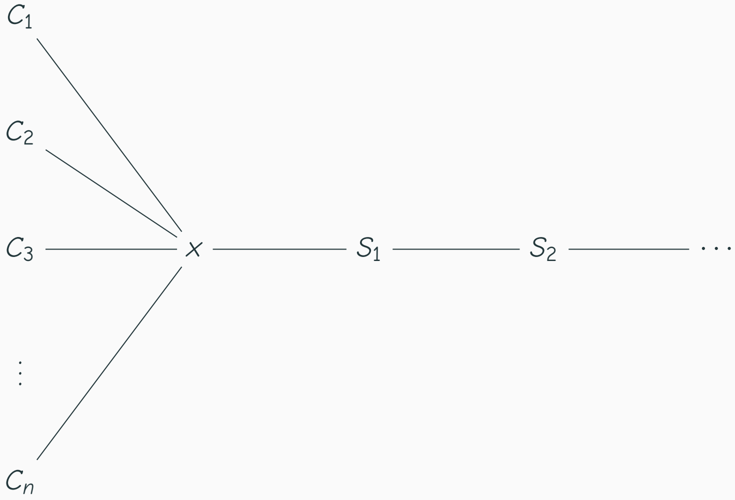
Lack of expressiveness

- unable to model **stateful** servers and **contention**
- examples: auctions, purchase of rare items, ...
- examples: locks, CAS registers, shared objects, ...

exponentials in Client-Server Linear Logic (CSLL)

Qian, Kavvos, and Birkedal [2021]

concurrent clients vs unlimited **sequential** instances of server



a linear logic with co-exponentials

$$A ::= \bot \mid \top \mid \mathbf{0} \mid \mathbf{1} \mid A \oplus B \mid A \,\&\, B \mid A \otimes B \mid A \parr B \mid ¿A \mid ¡A$$

Rules for co-clients

$$\frac{}{\vdash ¿A} \qquad \frac{\vdash \Gamma, A}{\vdash \Gamma, ¿A} \qquad \frac{\vdash \Gamma, ¿A \qquad \vdash \Delta, ¿A}{\vdash \Gamma, \Delta, ¿A}$$

Rule for co-servers

$$\frac{\vdash \Gamma \qquad \vdash \Gamma, A \qquad \vdash \Gamma, ¡A, ¡A}{\vdash \Gamma, ¡A}$$

a problem with CSLL

- we have solved the accuracy and expressiveness issues
- ... but now we're dealing with a non-standard linear logic for which no cut elimination result is known
- besides, cut reduction is **not deterministic** nor **confluent**

$$\frac{P \vdash \Gamma, ¿A \qquad Q \vdash \Delta, ¿A}{P :: Q \vdash \Gamma, \Delta, ¿A} \equiv \frac{Q \vdash \Delta, ¿A \qquad P \vdash \Gamma, ¿A}{Q :: P \vdash \Gamma, \Delta, ¿A}$$

- Qian, Kavvos, and Birkedal [2021] prove **deadlock freedom**, leaving termination as an **open problem**
- no termination $\Rightarrow$ no livelock freedom ☹

μMALL$^\infty$

Baelde, Doumane, and Saurin [2016], Doumane [2017]

Linear logic with **fixed points**

$$A ::= \bot \mid \top \mid \mathbf{0} \mid \mathbf{1} \mid A \oplus B \mid A \,\&\, B \mid A \otimes B \mid A \parr B \mid X \mid \mu X.A \mid \nu X.A$$

Infinitary proofs

- fixed points are simply **unfolded**
- proofs may be **infinite**
- **validity condition** on proofs

Properties

- valid proofs enjoy **cut elimination**

co-exponentials as fixed points

$¿A$ = make (**concurrently**) zero or more requests of A

$$¿A \stackrel{\text{def}}{=} \mu X.(\mathbf{1} \oplus A \oplus (X \otimes X))$$

$¡A$ = handle (**sequentially**) zero or more requests of A

$$¡A \stackrel{\text{def}}{=} \nu X.(\bot \,\&\, A \,\&\, (X \parr X))$$

Strategy for proving termination of CSLL (fallacy alert)

1. encode co-exponentials in CSLL into fixed points of μMALL^∞
2. encode well-typed CSLL process into valid μMALL^∞ proof
3. use cut elimination of μMALL^∞ to infer termination of CSLL

a problem with this correspondence between CSLL and μMALL$^{\infty}$

- all μMALL$^{\infty}$ cut reductions correspond to CSLL reductions
- **some** CSLL reductions **don't** correspond to μMALL$^{\infty}$ cut reductions

$$\frac{P \to Q}{P :: R \to Q :: R}$$

- clients may reduce **independently**, even before they connect to the server
- cut elimination of μMALL$^{\infty}$ only entails **weak termination** of CSLL ☹

from weak to fair termination

Theorem (subject reduction)

If P is well typed and $P \rightarrow Q$ then Q is well typed

$$
\begin{array}{ccccccc}
P & \rightarrow & P_1 & \rightarrow & P_2 & \rightarrow & \cdots \\
\text{well typed} & \Rightarrow & \text{well typed} & \Rightarrow & \text{well typed} & \Rightarrow & \cdots
\end{array}
$$

from weak to fair termination

Theorem (subject reduction)

If P is well typed and $P \to Q$ then Q is well typed

Theorem (weak termination)

If P is well typed then P is weakly terminating

$$\begin{array}{ccccccc}
P & \to & P_1 & \to & P_2 & \to & \cdots \\
\\
\text{well typed} & \Rightarrow & \text{well typed} & \Rightarrow & \text{well typed} & \Rightarrow & \cdots \\
\Downarrow & & \Downarrow & & \Downarrow & & \\
\text{weakly term.} & & \text{weakly term.} & & \text{weakly term.} & & \cdots
\end{array}$$

from weak to fair termination

Theorem (subject reduction)

If P is well typed and $P \rightarrow Q$ then Q is well typed

Theorem (weak termination)

If P is well typed then P is weakly terminating

$$
\begin{array}{ccccccc}
P & \rightarrow & P_1 & \rightarrow & P_2 & \rightarrow & \cdots \\
\\
\text{well typed} & \Rightarrow & \text{well typed} & \Rightarrow & \text{well typed} & \Rightarrow & \cdots \\
\Downarrow & & \Downarrow & & \Downarrow & & \\
\text{weakly term.} & & \text{weakly term.} & & \text{weakly term.} & & \cdots
\end{array}
$$

Theorem (Ciccone and Padovani [2022a])

$P \rightarrow^* Q$ implies Q weakly term. $\iff$ P fairly terminating

deadlock freedom + **fair** termination $\Rightarrow$ livelock freedom

concluding remarks

Properties of CSLL

- does it terminate? almost certainly yes, but still **open problem**
- does it enjoy livelock freedom? **yes**

concluding remarks

Properties of CSLL

- does it terminate? almost certainly yes, but still **open problem**
- does it enjoy livelock freedom? **yes**

Building on the expressiveness of μMALL$^\infty$

- binary sessions [Ciccone and Padovani, 2022b]
- client-server sessions [Padovani, 2023]
- **concurrent objects and actors?**
  [Crafa and Padovani, 2017, de'Liguoro and Padovani, 2018]

Properties of CSLL

- does it terminate? almost certainly yes, but still **open problem**
- does it enjoy livelock freedom? **yes**

Building on the expressiveness of μMALL$^\infty$

- binary sessions [Ciccone and Padovani, 2022b]
- client-server sessions [Padovani, 2023]
- **concurrent objects and actors?** [Crafa and Padovani, 2017, de'Liguoro and Padovani, 2018]

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 and Frank Pfenning. Session types as intuitionistic linear propositions. In Paul Gastin and François Laroussinie, editors, *CONCUR 2010 - Concurrency Theory, 21th International Conference, CONCUR 2010, Paris, France, August 31-September 3, 2010. Proceedings*, volume 6269 of *Lecture Notes in Computer Science*, pages 222–236. Springer, 2010.

Luca Ciccone and Luca Padovani. Fair termination of binary sessions. *Proc. ACM Program. Lang.*, 6(POPL):1–30, 2022a.

Luca Ciccone and Luca Padovani. An infinitary proof theory of linear logic ensuring fair termination in the linear π-calculus. In Bartek Klin, Slawomir Lasota, and Anca Muscholl, editors, *33rd International Conference on Concurrency Theory, CONCUR 2022, September 12-16, 2022, Warsaw, Poland*, volume 243 of *LIPIcs*, pages 36:1–36:18. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2022b.

Silvia Crafa and Luca Padovani. The chemical approach to typestate-oriented programming. *ACM Trans. Program. Lang. Syst.*, 39 (3):13:1–13:45, 2017.

Ugo de'Liguoro and Luca Padovani. Mailbox types for unordered interactions. In Todd D. Millstein, editor, *32nd European Conference on Object-Oriented Programming, ECOOP 2018, July 16-21, 2018, Amsterdam, The Netherlands*, volume 109 of *LIPIcs*, pages 15:1–15:28. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2018.

Amina Doumane. *On the infinitary proof theory of logics with fixed points. (Théorie de la démonstration infinitaire pour les logiques à points fixes)*. PhD thesis, Paris Diderot University, France, 2017. URL `https://tel.archives-ouvertes.fr/tel-01676953`.

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.

Luca Padovani. On the fair termination of client-server sessions. In Delia Kesner and Pierre-Marie Pédrot, editors, *28th International Conference on Types for Proofs and Programs (TYPES 2022)*, LIPIcs. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2023.

Zesen Qian, G. A. Kavvos, and Lars Birkedal. Client-server sessions in linear logic. *Proc. ACM Program. Lang.*, 5(ICFP):1–31, 2021.

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