on the almost-sure termination of binary sessions

Ugo Dal Lago Luca Padovani

outline

outline

type systems ensuring lock freedom of sessions

lock freedom (LF) = every communication is eventually completed

Direct approaches

e.g. Kobayashi [2002], Padovani [2014]

− $\mathsf{LF}(x) \wedge \mathsf{LF}(y) \not\Longrightarrow \mathsf{LF}(x \# y)$

Indirect approaches: deadlock freedom + termination ⇒ LF

e.g. Kobayashi and Sangiorgi [2008], Caires and Pfenning [2010], Lindley and Morris [2016]

+ compositional
+ applies to eventual (fair) termination

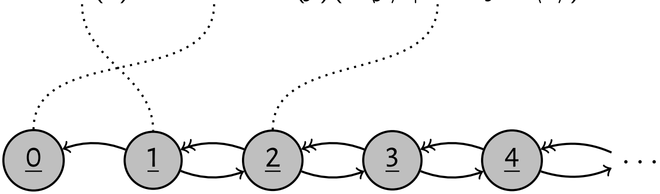
  e.g. **Dagnino and Padovani [2024]**

− termination can be too strong

− fair termination can be too weak

example: non-deterministic random walk

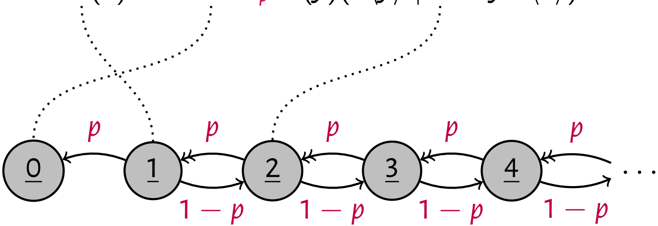
$$N(x) \triangleq \textcolor{blue}{\mathsf{close}}\, x \oplus (y)(N\langle y\rangle \mid \textcolor{blue}{\mathsf{wait}}\, y.N\langle x\rangle)$$



- the reduction $N\langle x\rangle \to$ close x can always occur
- the process is fairly terminating

example: probabilistic random walk

$$P(x) \triangleq \mathsf{close}\, x \;{}_{p}\!\boxplus\; (y)(P\langle y\rangle \mid \mathsf{wait}\, y.P\langle x\rangle)$$



- the reduction $P\langle x\rangle \to \mathsf{close}\, x$ occurs with probability p
- the process is terminating with probability 1 iff $p \geq 0.5$

outline

probabilistic semantics with **multidistributions**

instance of probabilistic term rewriting [Avanzini, Dal Lago, and Yamada, 2020]

$$P(x) \triangleq \mathsf{close}\, x\ {}_{p}\boxplus\ (y)(P\langle y\rangle \mid \mathsf{wait}\, y.P\langle x\rangle)$$

$$\boxed{1 : \underline{1}} \to \boxed{\begin{array}{rl} p & : \underline{0} \\ 1-p & : \underline{2} \end{array}} \twoheadrightarrow \boxed{\begin{array}{rl} p(1-p) & : \underline{1} \\ (1-p)^2 & : \underline{3} \end{array}} \twoheadrightarrow \boxed{\begin{array}{rl} p^2(1-p) & : \underline{0} \\ p(1-p)^2 & : \underline{2} \\ p(1-p)^2 & : \underline{2} \\ (1-p)^3 & : \underline{4} \end{array}} \twoheadrightarrow \cdots$$

We reduce multidistributions of processes

- terminated processes are erased
- multiple equal entries account for non-determinism

measure of a multidistribution

$$|\wr p_i : P_i \int_{i \in I}| \stackrel{\text{def}}{=} \sum_{i \in I} p_i$$

Intuition

- $|M| \geq$ probability that the processes in M are **not terminated**

Basic properties

- $1 \geq |M|$
- if $M \to N$ then $|M| \geq |N|$

almost sure termination (AST)

Definition (reduction sequence of P)
Infinite sequence $M_0 M_1 \cdots$ of multidistributions such that

- $M_0 = \{\!\{ 1 : P \}\!\}$
- $M_n \to M_{n+1}$ for every $n \in \mathbb{N}$

Note
All reduction sequences are infinite because $\{\!\{\}\!\} \to \{\!\{\}\!\}$

Definition (almost-sure termination)
P is AST if $\lim\limits_{n\to\infty} |M_n| = 0$ for every reduction sequence $M_0 M_1 \cdots$ of P

AST $\neq$ finite expected computation

$$P(x) \triangleq \text{close}\, x \;{}_{p}\boxplus\; (y)(P\langle y\rangle \mid \text{wait}\, y.P\langle x\rangle)$$

Property
If $p \geq 0.5$ then $P\langle x\rangle$ is almost surely terminating

Property
If $p = 0.5$ then the expected computation length is ∞

strong almost sure termination (SAST)

Definition (expected derivation length of $\overline{M} = M_0 M_1 \dots$)

$$\mathrm{edl}(\overline{M}) \stackrel{\text{def}}{=} \sum_{n\in\mathbb{N}} |M_n|$$

Definition (expected derivation height)

$$\mathrm{edh}(P) \stackrel{\text{def}}{=} \sup\{\mathrm{edl}(\overline{M}) \mid \overline{M} \text{ is a reduction sequence of } P\}$$

Definition
P is strongly almost-surely terminating (SAST) if $\mathrm{edh}(P) < \infty$

$$\text{SAST} \subsetneq \text{AST}$$

outline

types

$$
\begin{aligned}
A, B := \mathbf{1} \mid A \otimes B \mid \oplus\{\mathsf{a}_i^{\ell_i} : A_i\}_{i \in I} \mid ?^{\ell} A \\
\bot \mid A \parr B \mid \&\{\mathsf{a}_i^{\ell_i} : A_i\}_{i \in I} \mid !^{\ell} A
\end{aligned}
$$

Are

- based on classical linear logic
- possibly infinite

Have

- *n*-ary additives
- height annotations $\ell \in \mathbb{R}_{\geq 0}$

typing judgments and typing rules

Judgments

$$P \vdash^{\ell} \Gamma$$

- P is well typed in Γ
- $\ell \geq \mathrm{edh}(P)$

$$M \vdash^{\ell} \Gamma$$

- $M = \wr p_i : P_i \int_{i \in I}$ is well typed in Γ
- $\ell \geq \sum_{i \in I} p_i \mathrm{edh}(P_i)$

Typing rules are the same as in Francesco's talk, except…

- annotations are in $\mathbb{R}_{\geq 0}$
- the choice rule computes an expected height

$$\frac{P \vdash^{\ell} \Gamma \qquad Q \vdash^{\ell'} \Gamma}{P \;{}_{p}\!\boxplus\; Q \vdash^{1+p\ell+(1-p)\ell'} \Gamma}$$

properties of well-typed processes

Theorem (deadlock freedom)
If $P \vdash^{\ell} x : \mathbf{1}$, then P is deadlock free

Theorem (subject reduction)
If $M \vdash^{\ell} \Gamma$ and $M \to N$, then $N \vdash^{\ell'} \Gamma$ and $\ell \geq |M| + \ell'$

Theorem (soundness)
If $P \vdash^{\ell} \Gamma$, then P is SAST

example

$$P(x) \triangleq \mathsf{close}\,x \;{}_{p}\!\boxplus\; (y)(P\langle y\rangle \mid \mathsf{wait}\,y.P\langle x\rangle)$$

$$\dfrac{\dfrac{}{\mathsf{close}\,x \vdash^{1} x : \mathbf{1}}\,\mathbf{1} \qquad \dfrac{\dfrac{\vdots}{P\langle y\rangle \vdash^{\ell} y : \mathbf{1}} \qquad \dfrac{\dfrac{\vdots}{P\langle x\rangle \vdash^{\ell} x : \mathbf{1}}}{\mathsf{wait}\,y.P\langle x\rangle \vdash^{\ell} x : \mathbf{1}, y : \bot}\,\bot}{(y)(P\langle y\rangle \mid \mathsf{wait}\,y.P\langle x\rangle) \vdash^{2\ell} x : \mathbf{1}}\,\text{cut}}{P\langle x\rangle \vdash^{1+p+2(1-p)\ell} x : \mathbf{1}}\,\text{choice}$$

$$\ell = \frac{1+p}{2p-1} \qquad \Rightarrow \qquad \text{if } p > 0.5 \text{ then } P\langle x\rangle \text{ is SAST}$$

outline

conclusion

Summary

- type system for sessions with probabilistic choices
- SAST = termination with probability $1 \wedge$ finite expected computation
- termination $\subsetneq$ SAST $\subsetneq$ fair termination
- deadlock freedom $\wedge$ SAST $\Rightarrow$ lock freedom

In the paper

- ✔ more examples (buyer and seller, lottery, clients and server, …)
- ✔ weaker typing rule for choices that ensures AST instead of SAST

conclusion

Summary

- type system for sessions with probabilistic choices
- SAST = termination with probability $1 \wedge$ finite expected computation
- termination $\subsetneq$ SAST $\subsetneq$ fair termination
- deadlock freedom $\wedge$ SAST $\Rightarrow$ lock freedom

In the paper

- ✔ more examples (buyer and seller, lottery, clients and server, …)
- ✔ weaker typing rule for choices that ensures AST instead of SAST

thank you!

references

Martin Avanzini, Ugo Dal Lago, and Akihisa Yamada. On probabilistic term rewriting. *Sci. Comput. Program.*, 185, 2020.

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.

Francesco Dagnino and Luca Padovani. sMALL CaPS: An Infinitary Linear Logic for a Calculus of Pure Sessions. In *Proceedings of the 26th International Symposium on Principles and Practice of Declarative Programming (PPDP'24)*, 2024.

Naoki Kobayashi. A type system for lock-free processes. *Inf. Comput.*, 177(2): 122–159, 2002.

references (cont.)

Naoki Kobayashi and Davide Sangiorgi. A hybrid type system for lock-freedom of mobile processes. In Aarti Gupta and Sharad Malik, editors, *Computer Aided Verification, 20th International Conference, CAV 2008, Princeton, NJ, USA, July 7-14, 2008, Proceedings*, volume 5123 of *Lecture Notes in Computer Science*, pages 80–93. Springer, 2008.

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. Deadlock and lock freedom in the linear π-calculus. In Thomas A. Henzinger and Dale Miller, editors, *Joint Meeting of the Twenty-Third EACSL Annual Conference on Computer Science Logic (CSL) and the Twenty-Ninth Annual ACM/IEEE Symposium on Logic in Computer Science (LICS), CSL-LICS '14, Vienna, Austria, July 14 - 18, 2014*, pages 72:1–72:10. ACM, 2014.