π-calculus, references and implicit complexity

M2 internship, Daniel Hirschkoff, Plume team, ENS de Lyon

Context

The π-calculus [6] is a core calculus for concurrent and mobile computation based on the notion of channel. Various techniques have been studied to reason about π-calculus processes, notably behavioural equivalences, type systems and modal logics.

The π-calculus is very expressive: it is possible in particular to encode various features found in programming languages, including higher-order computation, concurrent or imperative constructs in the π-calculus..

The internship

The internship will focus on the study of type systems to insure termination in a π-calculus enriched with constructs to handle imperative programming such as references. The π-calculus with references from [4] will serve as a starting point for the internship.

Several works in the literature address termination for π-calculus processes, using enrichment of simple types [3, 2] or type systems based on connections with linear logic [5]. Termination in a concurrent setting can be difficult to guarantee. We are interested not only in ensuring termination of typable processes, but we also want to provide a bound on the time needed to reduce typable processes. Examples of works that follow this approach include [1, 5]. In [5], it is shown that a process calculus featuring probabilistic choices and complexity constraints can be used to model experiments and proofs from the so-called computational model of cryptography. The type system of [5] guarantees a complexity bound on the length of reductions of typable processes.

The goal of the internship is to see how the approach of [5] can be extended to languages with imperative state. Indeed, references can be very useful when modeling cryptographic protocols.

Keywords: π-calculus, type system, termination, implicit complexity.

This internship will take place in the Plume team of the LIP laboratory (ENS de Lyon). It will be jointly supervised by Daniel Hirschkoff (Plume) and Davide Sangiorgi (Università di Bologna and INRIA). A visit in Bologna may be organised to discuss on the progress of the internship.

References

[1] Patrick Baillot, Alexis Ghyselen, and Naoki Kobayashi. Sized types with usages for parallel complexity of pi-calculus processes. In Serge Haddad and Daniele Varacca, editors, *32nd International Conference on Concurrency*

Theory, CONCUR 2021, August 24-27, 2021, Virtual Conference, volume 203 of *LIPIcs*, pages 34:1–34:22. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2021.

[2] Romain Demangeon, Daniel Hirschkoff, and Davide Sangiorgi. Termination in impure concurrent languages. 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 328–342. Springer, 2010.

[3] Yuxin Deng and Davide Sangiorgi. Ensuring termination by typability. *Inf. Comput.*, 204(7):1045–1082, 2006.

[4] Daniel Hirschkoff, Enguerrand Prebet, and Davide Sangiorgi. On the representation of references in the pi-calculus. In Igor Konnov and Laura Kovács, editors, *31st International Conference on Concurrency Theory, CONCUR 2020, September 1-4, 2020, Vienna, Austria (Virtual Conference)*, volume 171 of *LIPIcs*, pages 34:1–34:20. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2020.

[5] Ugo Dal Lago and Giulia Giusti. On session typing, probabilistic polynomial time, and cryptographic experiments. 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 37:1–37:18. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2022.

[6] Davide Sangiorgi and David Walker. *The Pi-Calculus - a theory of mobile processes*. Cambridge University Press, 2001.