

# Research Directions

## Foreword

*In addition to what is presented in the [funded proposal](#) (available upon request), or at least refining it, are the following possible research directions. Please, refer to [the notes about reversibility](#) for a more general introduction as to why reversible computation is of interest.*

## A Reversible Machine for $\lambda$ -calculus

### In a nutshell

As [Ugo Dal Lago](#) observed in a [public message](#), there is no “reversible” machine capable of executing  $\lambda$ -terms. The goal of this direction is to develop such a model of computation, using a recent non-deterministic machine [1], to obtain machines capable of undoing their reductions of  $\lambda$ -terms (and possibly of  $\pi$ -calculus terms) for the first time. Gains include a better understanding of relation to reversible functional programming language, more generality in the machine, and possibly efficient reduction strategies.

### In more details

The question to investigate is: is it possible to “revert” a fundamental model of computation, namely the  $\lambda$ -calculus [2]?

Answering positively this question requires to

1. Define a reversible abstract model of computation  $M$ :
  - a) Define a forward execution, i.e., rules expressing that the model  $M$  in a state  $s$  will reduce the  $\lambda$ -terms  $u$  to  $u'$  and move to state  $s'$  in one forward step  $:(M, s, u) \rightarrow (M, s', u')$ ,
  - b) Define a backward execution, i.e., rules expressing that the model  $M$  in a state  $s$  will reduce the  $\lambda$ -terms  $u$  to  $u'$  and move to state  $s'$  in one backward step  $:(M, s, u) \rightsquigarrow (M, s', u')$ ,
2. Ensure that the model is “correct”:
  - a) By showing that it can correctly execute forward and backward some simple examples,
  - b) By proving properties such as the loop lemma, i.e.,  $(M, s, u) \rightarrow (M, s', u') \leftrightarrow (M, s', u') \rightsquigarrow (M, s, u)$ ,
  - c) By comparing it to existing reversible functional languages [3],
  - d) By exploring if there are additional features and properties revealed by this model that were not previously accessible.

Possible interesting results include:

1. Defining a machine that can implement multiple strategies by simply fidgeting with the backward mechanism,
2. Obtaining efficient machine, as it could explore  $\lambda$ -terms non-deterministically (that is, starting with multiple strategies at the same time, and terminating as soon as one obtained a normal term),
3. Obtaining a simpler model than the original one [1, Section 2], since it will be “manually” refined,
4. Extending the machine to execute other languages in a reversible manner (typically, HOCore [4]).

**What to read?** A good starting point will be [Logan Beatty’s slide](#) that he used to explain our research project during [Dr. Medić’s Visit](#). A good understanding of non-deterministic abstract machines [1, Section 2] is required, and some existing notes are stored in a [private repository](#).

### Possible collaborators:

- Nate Schwartz and Logan Beatty, two undergraduate students that worked on this model,
- [Claudio Antares Mezzina](#),
- [Jorge A. Pérez](#)

# A Reversible Applied $\pi$ -calculus

## In a nutshell

There has been multiple proposals for reversible  $\pi$ -calculus and its semantics [5–9] but, to my knowledge, applied  $\pi$ -calculus [12] has never been reversed. The goal of this direction is to introduce such a model, to compare it to existing reversible  $\pi$ -calculi, and to reason about its application to the audit of (reversible) security protocols.

## In more details

The applied  $\pi$ -calculus is an important starting point to develop audit tools such as ProVerif [13,14], used to verify protocols used in production. Its semantics has recently been refined into a cleaner model [12,15,16], where dependence is easier to track and expected properties (typically, the “diamond” lemmas) proven to hold. The goal of this research project would be to reverse its semantics and to compare it to existing models of reversible, name-passing, computation. In particular, assessing whether the criteria from [8] are met, and whether name extrusion (with disjunctive causality) [12, p. 9] is handled in a nicer way than in [17]

**What to read?** The “modern” take on applied  $\pi$ -calculus is nicely explained in [12], but the reversible semantics of  $\pi$ -calculi are spread through the literature and will be harder to identify. Nevertheless, my understanding is that the early model [7] is still relevant.

### Possible collaborators:

- Ross Horne,
- Semen Yurkov

## A Better Understanding of (H)HPB

- as dependence,
- complexity,
- prove that it maps BS.

## Replication, Recursion and Iteration

[18]

causal consistent encoding

## Choice

### Axiomatic Approach – Verified

### (Parallel) Random-Access Machine

Link with memory cells (guarded and locations).

## References

- [1] M. Biernacka, D. Biernacki, S. Lenglet, A. Schmitt, Non-deterministic abstract machines, in: B. Klin, S. Lasota, A. Muscholl (Eds.), 33rd International Conference on Concurrency Theory, CONCUR 2022, September 12–16, 2022, Warsaw, Poland, Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2022: pp. 7:1–7:24. <https://doi.org/10.4230/LIPICS.CONCUR.2022.7>.
- [2] B.C. Pierce, Types and programming languages, MIT Press, London, England, 2002.
- [3] R. Glück, T. Yokoyama, Reversible computing from a programming language perspective, Theoretical Computer Science 953 (2023) 113429. <https://doi.org/10.1016/J.TCS.2022.06.010>.

- [4] P. Maksimovic, A. Schmitt, HOCore in coq, in: C. Urban, X. Zhang (Eds.), *Interactive Theorem Proving - 6th International Conference, ITP 2015, Nanjing, China, August 24-27, 2015, Proceedings*, Springer, 2015: pp. 278–293. [https://doi.org/10.1007/978-3-319-22102-1\\_19](https://doi.org/10.1007/978-3-319-22102-1_19).
- [5] F. Tiezzi, N. Yoshida, Reversible session-based pi-calculus, *The Journal of Logic and Algebraic Programming* 84 (2015) 684–707. <https://doi.org/10.1016/j.jlamp.2015.03.004>.
- [6] E. Graversen, I.C.C. Phillips, N. Yoshida, Event structures for the reversible early internal  $\pi$ -calculus, *Journal of Logical and Algebraic Methods in Programming* 124 (2022) 100720. <https://doi.org/10.1016/j.jlamp.2021.100720>.
- [7] I. Cristescu, Operational and denotational semantics for the reversible  $\pi$ -calculus, PhD thesis, Université Paris Diderot – Paris 7–Sorbonne Paris Cité, 2015. <http://scholar.harvard.edu/files/cristescu/files/these.pdf>.
- [8] I. Cristescu, J. Krivine, D. Varacca, A compositional semantics for the reversible p-calculus, in: *LICS, IEEE Computer Society*, 2013: pp. 388–397. <https://doi.org/10.1109/LICS.2013.45>.
- [9] I. Lanese, C.A. Mezzina, J.-B. Stefani, Reversibility in the higher-order  $\pi$ -calculus, *Theoretical Computer Science* 625 (2016) 25–84. <https://doi.org/10.1016/j.tcs.2016.02.019>.
- [10] M.D. Ryan, B. Smyth, Applied pi calculus, in: V. Cortier, S. Kremer (Eds.), *Formal Models and Techniques for Analyzing Security Protocols*, IOS Press, 2011: pp. 112–142. <https://doi.org/10.3233/978-1-60750-714-7-112>.
- [11] M. Abadi, B. Blanchet, C. Fournet, The applied pi calculus: Mobile values, new names, and secure communication, *Journal of the ACM* 65 (2018) 1:1–1:41. <https://doi.org/10.1145/3127586>.
- [12] C. Aubert, R. Horne, C. Johansen, Diamonds for security: A non-interleaving operational semantics for the applied pi-calculus, in: B. Klin, S. Lasota, A. Muscholl (Eds.), *33rd International Conference on Concurrency Theory, Schloss Dagstuhl–Leibniz-Zentrum für Informatik*, 2022: pp. 30:1–30:26. <https://doi.org/10.4230/LIPIcs.CONCUR.2022.30>.
- [13] B. Blanchet, Modeling and verifying security protocols with the applied pi calculus and ProVerif, *Foundations and Trends in Privacy and Security* 1 (2016) 1–135. <https://doi.org/10.1561/33000000004>.
- [14] B. Blanchet, V. Cheval, ProVerif: Cryptographic protocol verifier in the formal model, (2021). <https://bblanche.gitlabpages.inria.fr/proverif/>.
- [15] C. Aubert, R. Horne, S. Mauw, C. Johansen, Unlinkability and history preserving bisimilarity, (n.d.). <https://aubert.perso.math.cnrs.fr/recherche/partages/temp/ COSE-D-23-02121.pdf>.
- [16] C. Aubert, R. Horne, C. Johansen, Bisimulations respecting duration and causality for the non-interleaving applied  $\pi$ -calculus, in: V. Castiglioni, C.A. Mezzina (Eds.), *Proceedings Combined 29th International Workshop on Expressiveness in Concurrency and 19th Workshop on Structural Operational Semantics*, Warsaw, Poland, 12th September 2022, Open Publishing Association, 2022: pp. 3–22. <https://doi.org/10.4204/EPTCS.368.1>.
- [17] S. Crafa, D. Varacca, N. Yoshida, Event structure semantics of parallel extrusion in the pi-calculus, in: L. Birkedal (Ed.), *Foundations of Software Science and Computational Structures - 15th International Conference, FOSSACS 2012, Held as Part of the European Joint Conferences on Theory and Practice of Software, ETAPS 2012, Tallinn, Estonia, March 24 - April 1, 2012. Proceedings*, Springer, 2012: pp. 225–239. [https://doi.org/10.1007/978-3-642-28729-9\\_15](https://doi.org/10.1007/978-3-642-28729-9_15).
- [18] C. Aubert, Replications in reversible concurrent calculi, in: M. Kutrib, U. Meyer (Eds.), *Reversible Computation - 15th Conference on Reversible Computation, RC 2023, Giessen, Germany, July 18-19, Proceedings*, Springer, 2023: pp. 15–23. [https://doi.org/10.1007/978-3-031-38100-3\\_2](https://doi.org/10.1007/978-3-031-38100-3_2).