Research Directions

Fundamentals

Reversible computing is gaining a lot of traction as a debugging and forensic tool, but also due to its connections to quantum and adiabatic computing [1]. We also know since the 70's and *Bennett's trick* that it is possible to "revert" any Turing Machine, in a precise technical sense [**?**].

[2]

[3]

A Reversible Machine for λ-calculus

In a nutshell

As Ugo Dal Lago observed in a public message, there is no "reversible" machine capable of executing λ-terms. The goal of this direction is to develop such a model of computation, using a recent non-deterministic machine [3], to obtain machines capable of undoing their reductions of λ-terms (and possibly of π-calculus terms) for the first time.

In more details

The question to investigate is: is it possible to "revert" a fundamental model of computation, namely the λ-calculus [4]?

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 λ-terms u to u' and move to state s' in one forward step :$(M, s, u) \to (M, s', u')$,
   b) Define a backward execution, i.e., rules expressing that the model M in a state s will reduce the λ-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., $(M, s, u) \to (M, s', u') \Leftrightarrow (M, s', u') \rightsquigarrow (M, s, u)$,
   c) By comparing it to existing reversible functional languages [5],
   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 λ-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 [3], since it will be "manually" refined,
4. Extending the machine to execute other languages in a reversible manner (typically, HOCore [6]).

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 [3] 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

References

[1] B. Aman, G. Ciobanu, R. Glück, R. Kaarsgaard, J. Kari, M. Kutrib, I. Lanese, C.A. Mezzina, L. Mikulski, R. Nagarajan, I.C.C. Phillips, G.M. Pinna, L. Prigioniero, I. Ulidowski, G. Vidal, Foundations of reversible computation, in: I. Ulidowski, I. Lanese, U.P. Schultz, C. Ferreira (Eds.), Reversible Computation: Extending Horizons of Computing - Selected Results of the COST Action IC1405, Springer, 2020: pp. 1–40. https://doi.org/10.1007/978-3-030-47361-7_1.

[2] 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.

[3] 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.

[4] B.C. Pierce, Types and programming languages, MIT Press, London, England, 2002.

[5] 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.

[6] 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.