Internship Proposal
Implementing Reversible Process Algebras

Keywords
Process Algebra, Concurrency, Reversible Calculi, Process Semantics

Location
Augusta University, GA, USA

Advisor
Clément Aubert, School of Computer and Cyber Sciences, Augusta University

Preferred Skills
Interest in formal languages and specification, curiosity for distributed computation,
abstract and logic reasoning, programming skills.

Context. Process algebras (π-calculus, CCS, Ambient calculus, etc.) are an abstraction of concurrent systems useful to study, specify and verify distributed behaviors. Implementing process calculi serves three overlapping goals:

- It allows to machine-check theorems and definitions [1,2] using proof assistants such as Coq [3], resulting sometimes in simplications [2] or the finding of regrettable imprecisions and errors [4].
- Using it as an actual programming language, it enables the implementation of toy programs [5] that exemplifies the purpose and expressivity of the calculus.
- It can also be used as a specification language: typically, the Proverif tool [6], which implements the applied π-calculus [7], has been used to certify and model security protocols in a variety of areas [8].

The CCS language undergoes two different efforts making it amenable to represent computation that can move backward and forward: Reversible CCS (RCCS) [9] and CCS with keys (CCSK) [10] were both developed with the goal of becoming the extension to CCS providing a better understanding of the mechanisms underlying reversible concurrent computation—they actually turned out to be the two faces of the same coin [11]. Reversible computation in general has received a lot of attention from different communities [12], and the study of reversible process calculi has made important progresses in the recent years [12, Sect. 6]. However, aside from SimCCSK [13]—which is not publicly available and not maintained since 2008—no implementation of concurrent, reversible CCS exists.

Goals. The student will work toward an implementation of either CCSK, RCCS, or a declension of them [14], possibly taking inspiration of existing implementation of forward-only CCS (among wich this project or this web-interface [15]) or of intermediate languages such as HOcore [4]. A great care will be required toward good software engineering practices, the development of good examples, the possible certification of some results using machine-checked proof, and/or the implementation of an efficient mechanism to distribute the generation of keys and identifiers.

Perks. Dr. Aubert has an history of involving undergraduate, graduate and post-graduate students in his research, and can tune the level and nature of his engagement in the student’s research based on their needs and tastes. Even if the advisor have a strategy in mind, they remain open to suggested deviations from this program based on mutual interest. The student will be working on a ground-breaking topic that is still within reach with limited theoretical background.

References.


