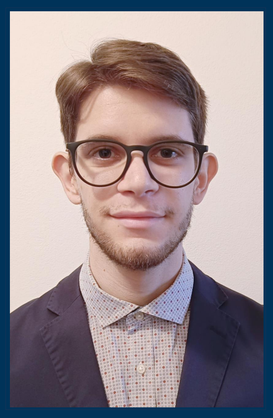
FORMAL METHODS FOR REVERSIBLE CONCURRENT CALCULI



Gabriele Cecilia

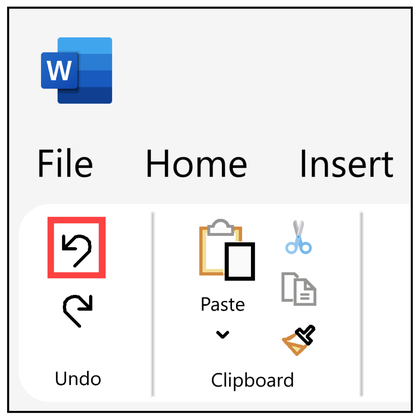
PhD student in Computer and Cyber Sciences

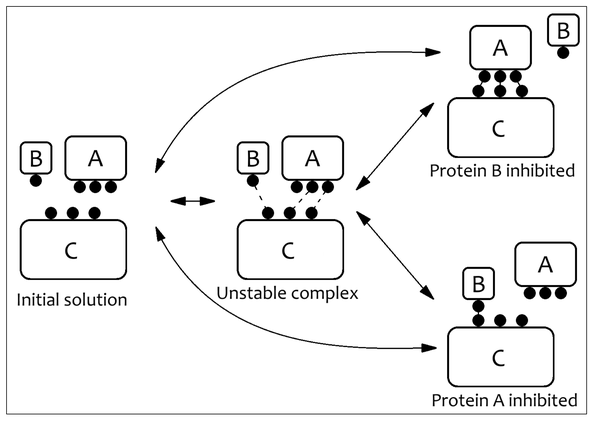
AUGUSTA UNIVERSITY

gcecilia@augusta.edu

REVERSIBILITY

Reversibility: the ability to undo an action or computation

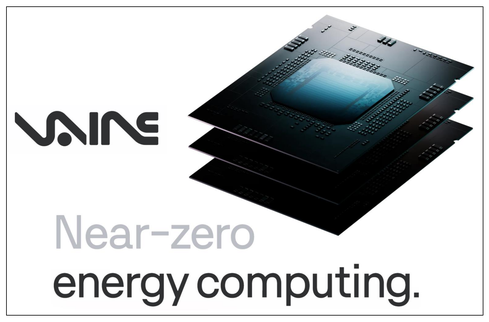


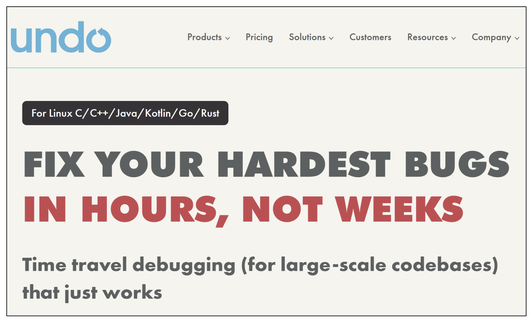


Irreversible computations ⟶ heat dissipation

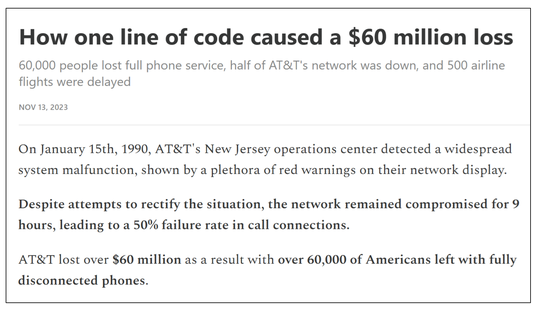
Reversible computations ⟶ no energy loss

Domains of application:





FORMAL METHODS



How one line of code caused a $60 million loss

60,000 people lost full phone service, half of AT&T's network was down, and 500 airline flights were delayed

NOV 13, 2023

On January 15th, 1990, AT&T's New Jersey operations center detected a widespread system malfunction, shown by a plethora of red warnings on their network display.

Despite attempts to rectify the situation, the network remained compromised for 9 hours, leading to a 50% failure rate in call connections.

AT&T lost over **$60 million** as a result with **over 60,000 of Americans left with fully disconnected phones.**

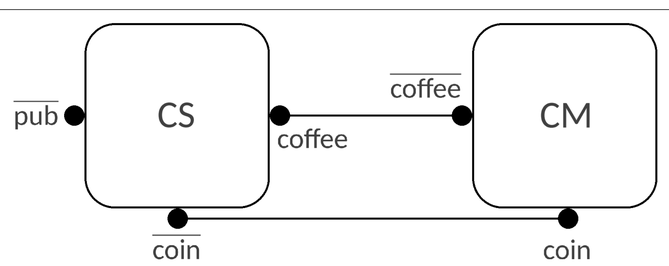
Formal Methods: highest level of software correctness

- Mathematical description of the specifications of a program
- Interactive theorem provers, such as: Rocq, Lean, Beluga, ...

CONCURRENT CALCULI

Concurrent Calculi: mathematical models of concurrent systems

Examples: Calculus of Communicating Systems (CCS), π-calculus



$CS \stackrel{\text{def}}{=} \overline{pub}.\overline{coin}.coffee.CS$ (Computer Scientist)

$CM \stackrel{\text{def}}{=} coin.\overline{coffee}.CM$ (Coffee Machine)

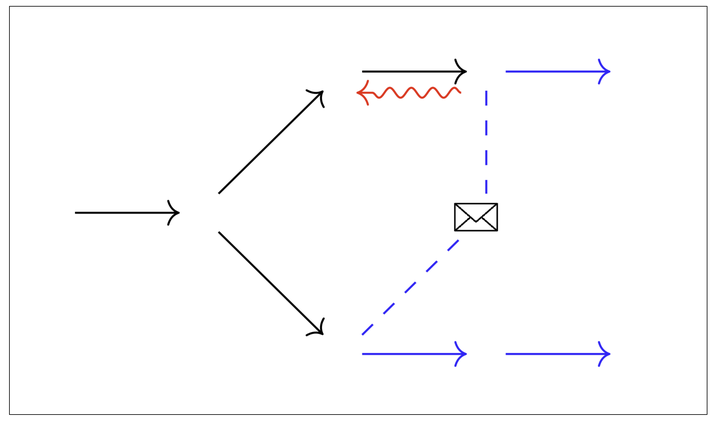
$CS \mid CM \xrightarrow{\overline{pub}} \overline{coin}.coffee.CS \mid CM \xrightarrow{\tau} coffee.CS \mid \overline{coffee}.CM$

REVERSIBLE CONCURRENT CALCULI

Reversible Concurrent Calculi: models of concurrent systems in which every action can be undone

Challenges addressed:

- Recovering inputs from outputs in a memory-efficient way
- Complexity of undoing computations in concurrent systems



Features:

- Syntax enriched with a memory or communication keys
- Semantics given by forward and backward transitions

RESEARCH PROJECT

This project aims at expanding and improving the existing reversible concurrent calculi, in particular by formalizing their definitions and properties with proof assistants.

Example of formalization:

Syntax of CCSK (CCS with Keys)		Beluga Formalization
X, Y ::=		`LF proc: type =`
$\mathbf{0}$	(Inactive)	`  | null: proc`
$\mid \alpha.X$	(Prefix)	`  | pref: labels → proc → proc`
$\mid \alpha[k].X$	(Keyed prefix)	`  | kpref: labels → keys → proc → proc`
$\mid X + Y$	(Sum)	`  | sum: proc → proc → proc`
$\mid X \mid Y$	(Parallel composition)	`  | par: proc → proc → proc`
$\mid X \setminus a$	(Restriction)	`  | nu: (names → proc) → proc;`

Publications:

- G.C. (2025): *A Formalization of the Reversible Concurrent Calculus CCSK^P in Beluga.* ICE 2025.

CURRENT AND FUTURE WORK DIRECTIONS

- Journal paper extending the formalization of CCSK^P and the paper "*Independence and Causality in the Reversible Concurrent Setting*" by Aubert, Phillips & Ulidowski
- Defining appropriate notions of behavioral equivalence (bisimulations) for reversible concurrent calculi
- Formalizing the results presented in "*An Axiomatic Theory for Reversible Computation*" by Lanese, Phillips & Ulidowski

ACKNOWLEDGMENTS

This research project is supported by the National Science Foundation under Grant No. 2242786 (SHF:Small:Concurrency In Reversible Computations).