



gcecilia@augusta.edu

# FORMAL METHODS FOR REVERSIBLE CONCURRENT CALCULI

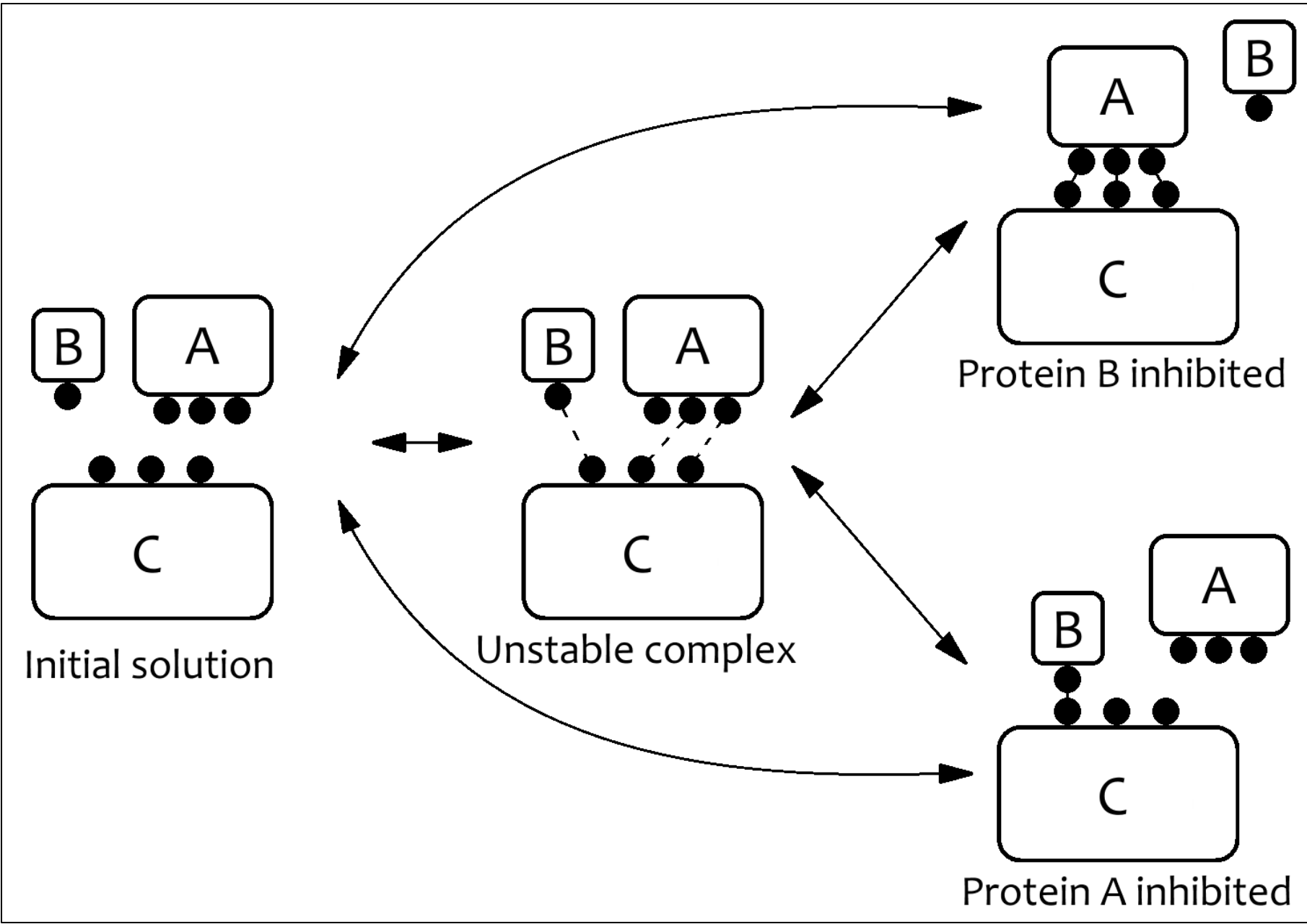
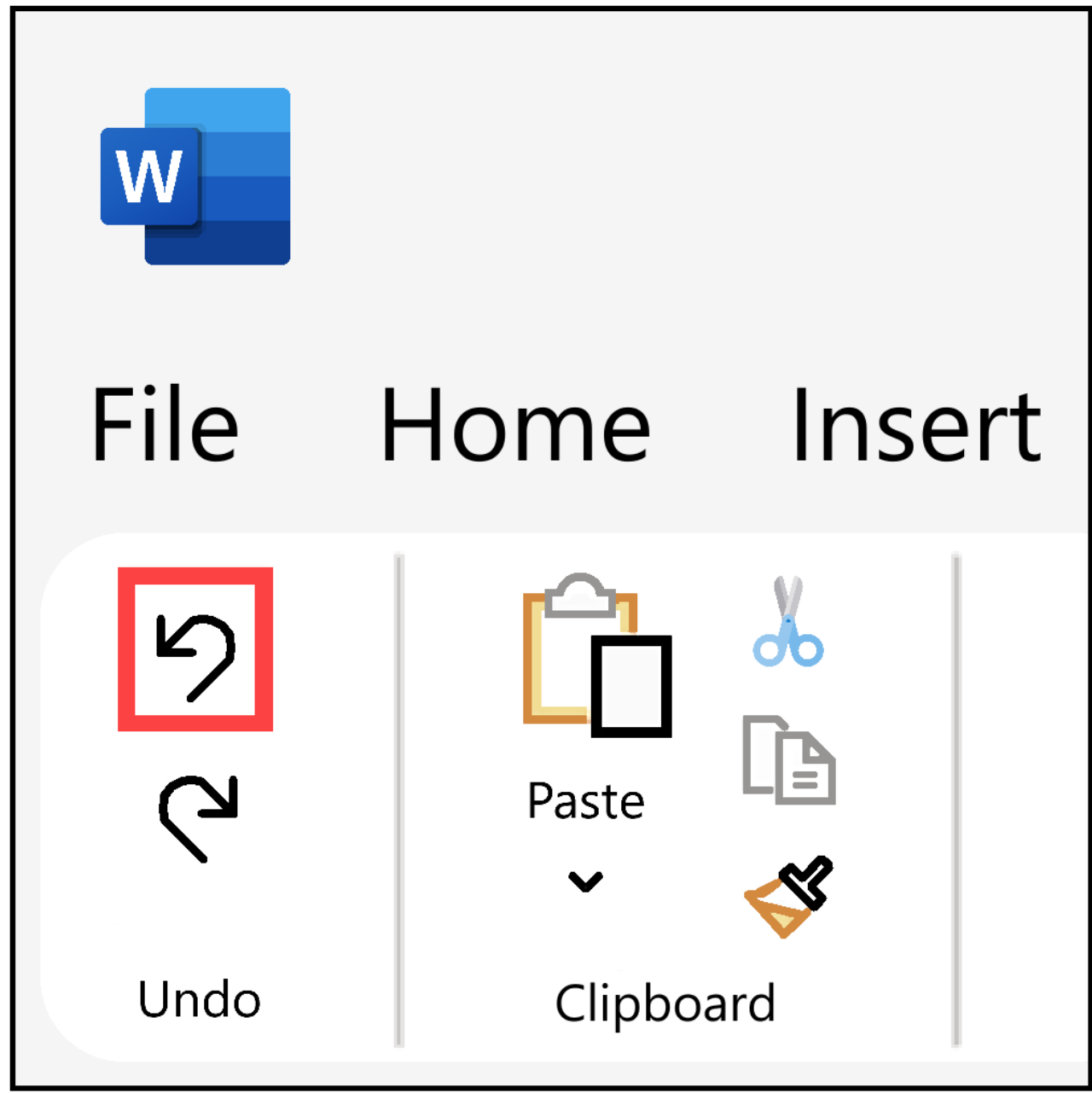
Gabriele Cecilia

School of Computer and Cyber Sciences, Augusta University, Augusta, GA USA



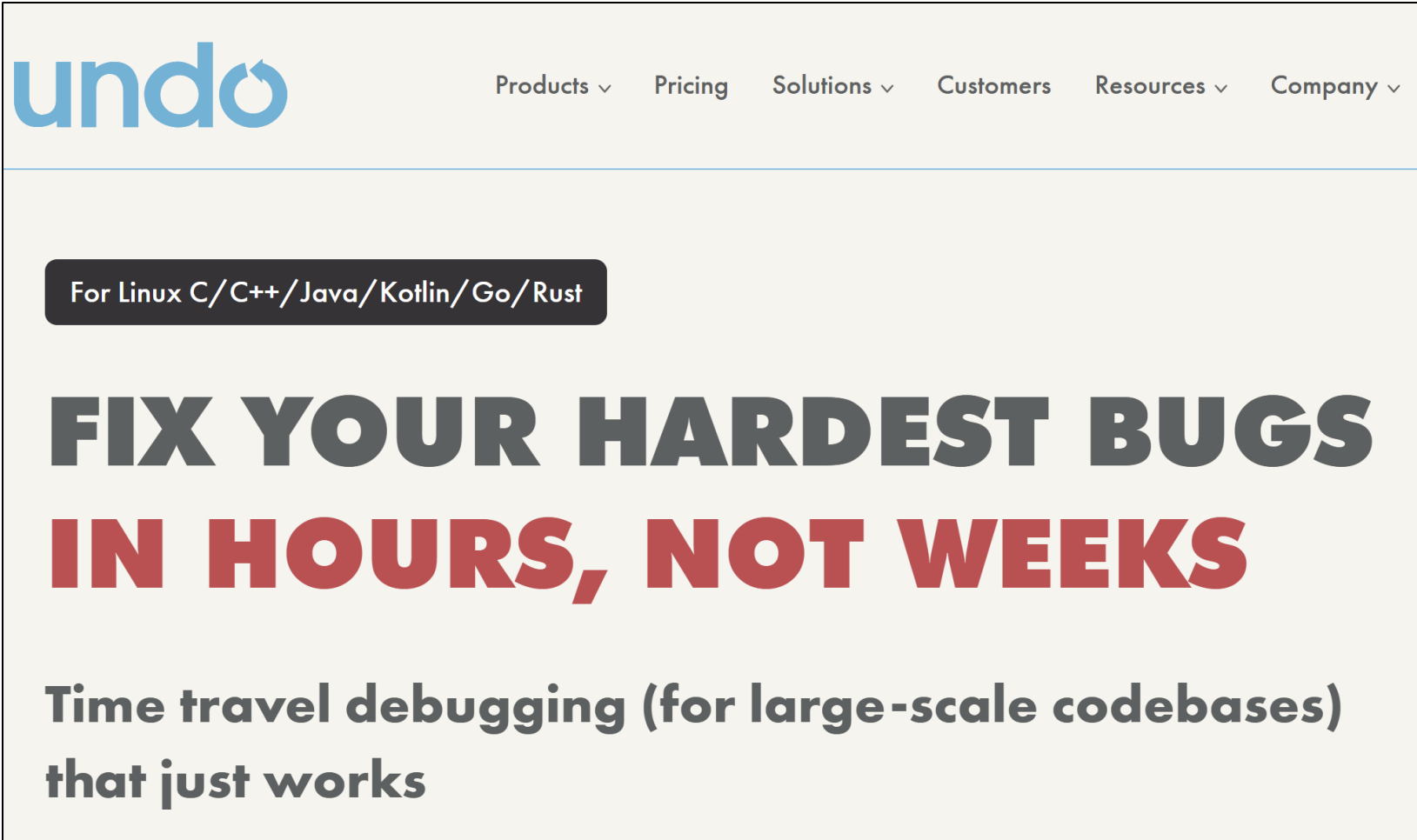
## REVERSIBILITY

**Reversibility:** the ability to undo an action or computation



Irreversible computations → heat dissipation  
Reversible computations → no energy loss

**Domains of application:**



## FORMAL METHODS

Programming errors can be deadly and expensive:

**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
- Automated and interactive **theorem provers**, such as: Rocq, Lean, Beluga, Z3, ...

## CONCURRENT CALCULI

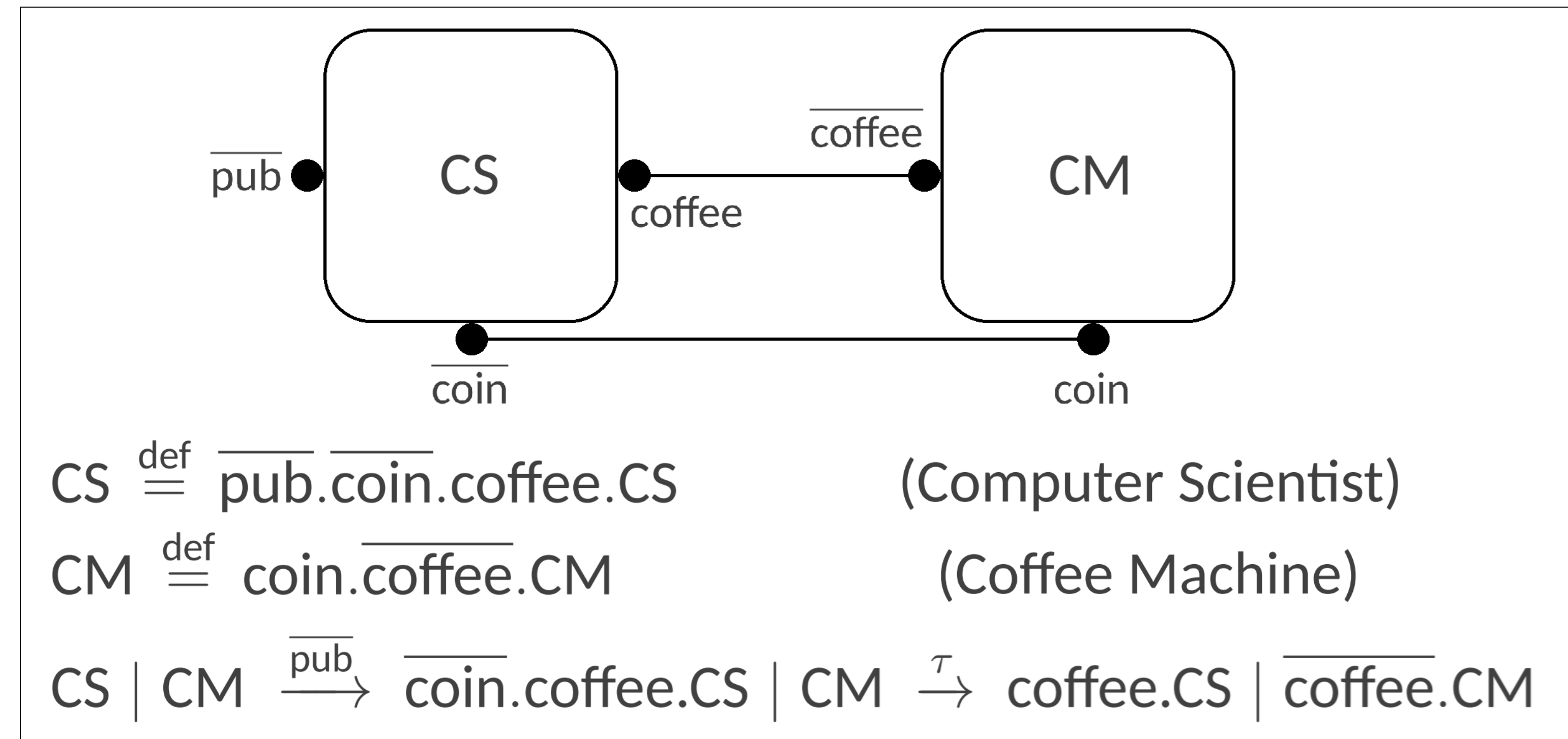
**Concurrent Calculi:**

- Mathematical models of concurrent systems
- A specialized domain of formal methods application

**Examples:** Calculus of Communicating Systems (CCS),  $\pi$ -calculus

**Ingredients:**

- **Syntax:** processes and communication channels
- **Semantics:** Labelled Transition System (LTS)

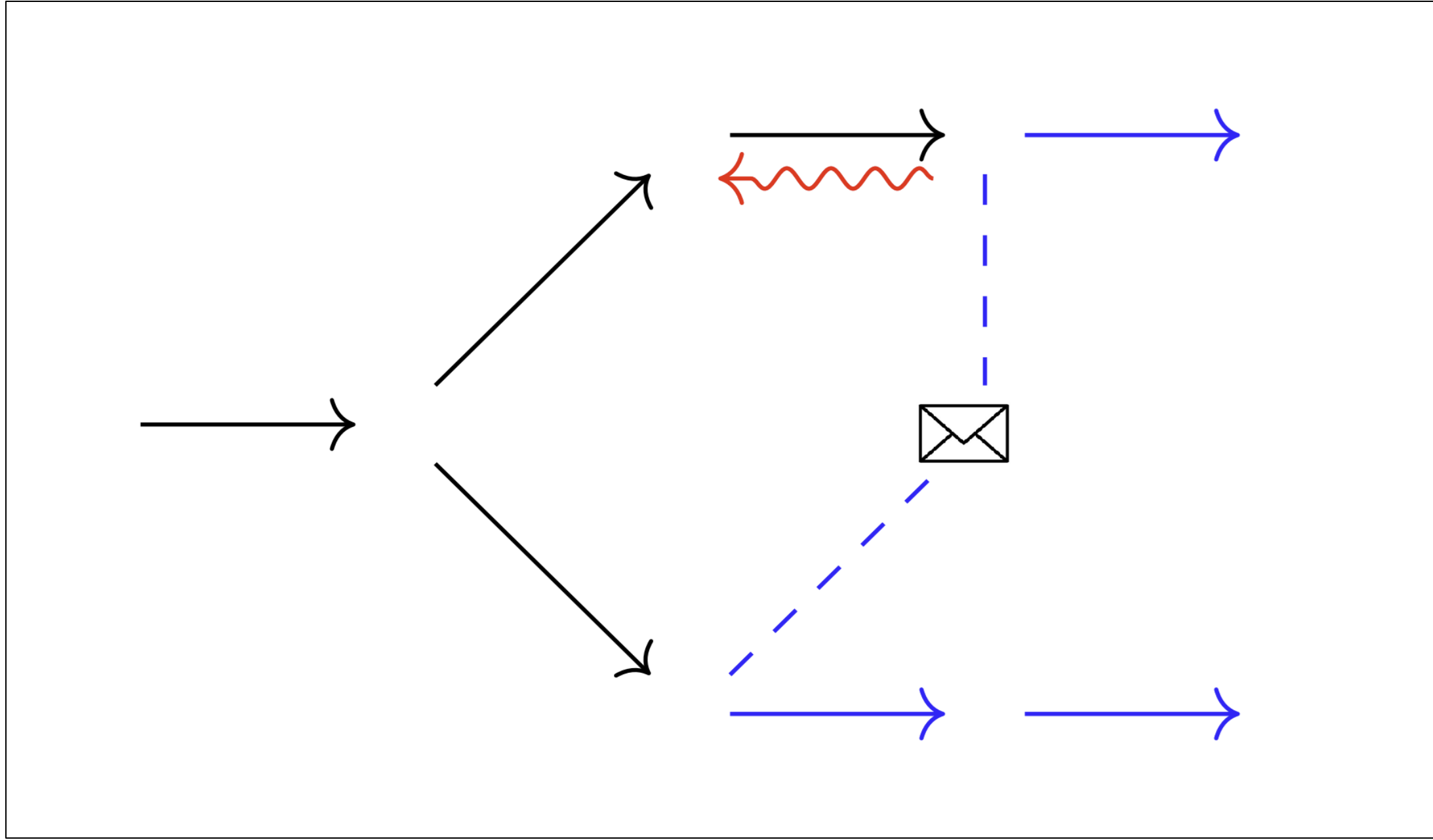


## REVERSIBLE CONCURRENT CALCULI

**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



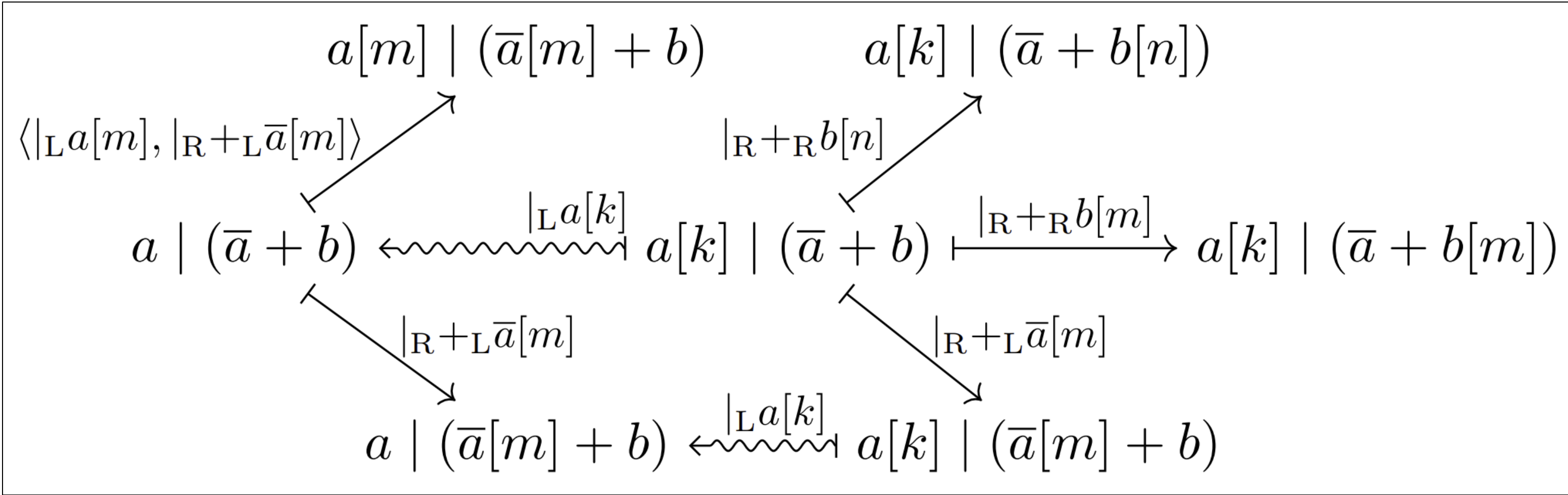
**Features:**

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

**Loop lemma:**

For all  $t : X \xrightarrow{\theta} Y$ , there exists  $\underline{t} : Y \xrightarrow{\ell} X$ , and conversely. Furthermore,  $\underline{\underline{t}} = t$ .

**Example of processes and transitions:**



## RESEARCH PROJECT

This project is aimed at the study and the improvement of existing reversible concurrent calculi, together with the formalization of their definitions and properties.

**Example of formalization:**

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

**Publications:**

- G.C. (2025): A Formalization of the Reversible Concurrent Calculus CCSK<sup>P</sup> in Beluga. ICE 2025.

## POSSIBLE FUTURE WORK DIRECTIONS

- Extending the formalization of CCSK<sup>P</sup>, e.g., by encoding **bisimulations**
- Providing formal definitions of constructs representing **infinite behavior** for reversible concurrent calculi
- Formalizing the key 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).