Design-by-contract and Behavioural types

Hernán Melgratti

ICC - University of Buenos Aires-Conicet

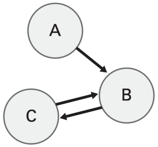
November 2025@GSSI

What is this course about?

- Development of **distributed systems**

  A **distributed system** *is a collection of components that work together to achieve a common goal.*

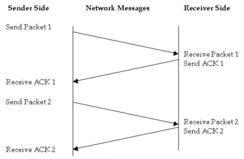
  - Nodes (**Components**): Perform computation
  - Edges (**Interaction / Communication / Synchronization**): Message exchanges
- We will focus on **communication**.
- What can go wrong with communication?
  - a component may receive a message it does not expect,
  - a component may wait for a message that never arrives,
  - a component may send messages that are never processed, ...



How do we describe components interaction?

- Communication protocols

  A **communication protocol** *is a set of rules that define how information is exchanged between components.*



- We will focus on:
  - How to formally describe communcation protocols.
  - Ensuring that components **correctly implement** a protocol.

A programming language perspective

We address the problem of correct protocol implementation by using programming language techniques:

- Behavioural types, which describe **how programs behave** in terms of sequence of actions, interactions, or state changes.
  - Formal definition of programming languages: syntax, semantics, types and type systems.
  - Models of concurrency: process calculi and concurrent lambda calculus
- Design-by-contract (DbC): a software design principle where each part of a program (such as a function or class) has a formal agreement that defines **what it can expect** and **what it should guarantee**.
  - Programs are equipped with
    - **Preconditions**: what must be true before the operation runs.
    - **Postconditions**: what must be true after the operation finishes.
  - A runtime mechanism identify **contract violations** and **blame** faulty components.

Syllabus

- Binary Session types
  - Session Process calculi
  - Concurrent functional language with session types
  - An Ocaml implementation of session types (hybrid approach)
- Design-by-Contract
  - Higher order contracts in the lambda calculus
- Contracts for sessions (dynamic approach)
- Multiparty session types
- Design-by-Contract for choreographies (static approach)

Timetable

- Is the 8:30–10:00 time slot suitable? during the afternoon?
- Upcoming meetings?:
  - **This week:** Tuesday and Thursday
  - **Next week:** Tuesday, Thursday, and Friday
  - **First week of December:** Open agenda