NSF CPS PI Meeting 2021



iCyPhy

Toward Deterministic and Verifiable Cyber-Physical Systems

Soroush Bateni, Edward A. Lee, Shaokai Lin, Marten Lohstroh,
Christian Menard, Alexander Schulz-Rosengarten, Sanjit Seshia

* Thanks to Marten Lohstroh for helping with the slides.



“Everything is Asynchronous.”

- Cyber-physical systems
  - Performance, scalability, flexibility, complexity
- Dominant parallel and distributed programming paradigms have relinquished **determinism**: "everything is asynchronous."
  - Actors, publish-subscribe, service-oriented architectures, distributed shared memory
  - Even in safety-critical domains: e.g., ROS2, Autosar Adaptive Platform[1], etc.

1. Menard, Christian, et al. "Achieving determinism in adaptive AUTOSAR." *2020 Design, Automation & Test in Europe Conference & Exhibition (DATE)*. IEEE, 2020.



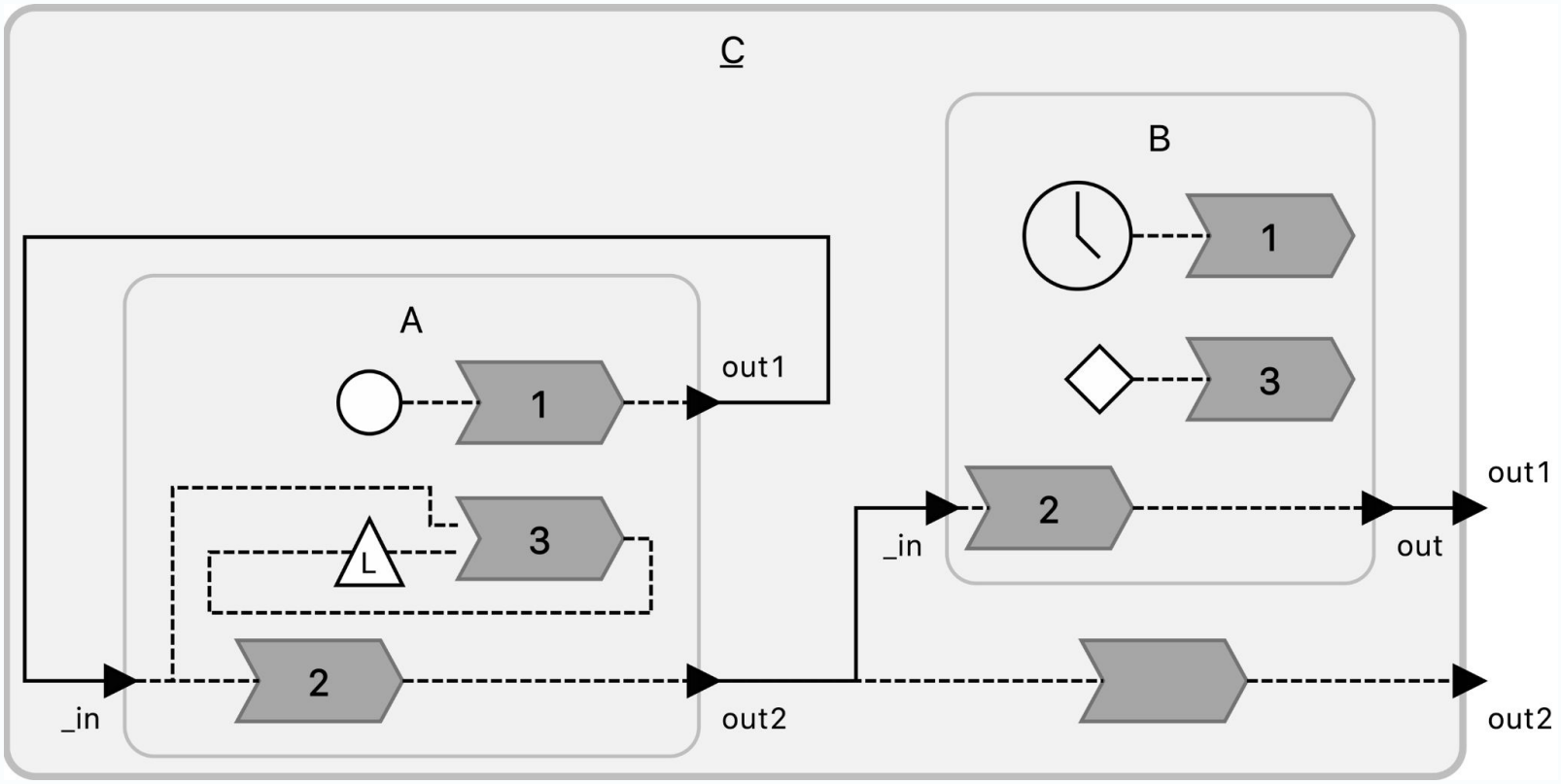
Deterministic Models are Useful

A model is deterministic if, given the initial state and the inputs, the model defines exactly one behavior.

Determinism

- Enables testing and more tractable analysis
- Makes simulation more useful
- Allows verification to scale better

Reactors[1] in a Nutshell



1. M. Lohstroh, "Reactors: A Deterministic Model of Concurrent Computation for Reactive Systems," PhD dissertation, University of California, Berkeley, 2020.

Logical Time and Physical time

Logical Time



- Steps or 'ticks'
- Discrete
- Absolute Simultaneity



- Deadlines
- External events
- Federation
- Fault handling

Physical Time



- Measurements
- Continuous
- Relativistic Simultaneity

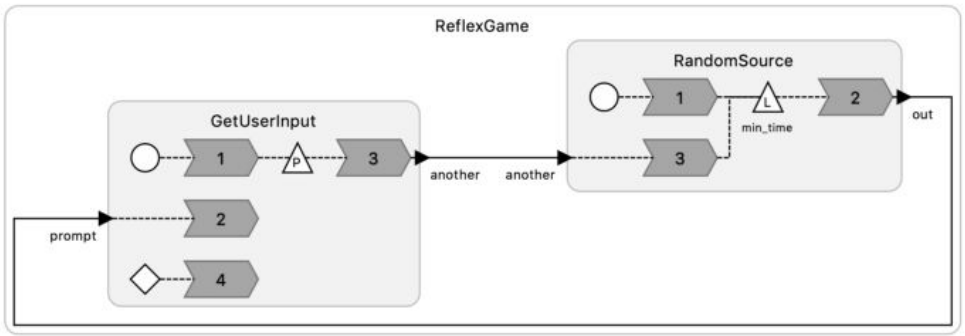
The Persistence of Memory — Salvador Dalí

Lingua Franca: It's About Time

ReflexGame.lf

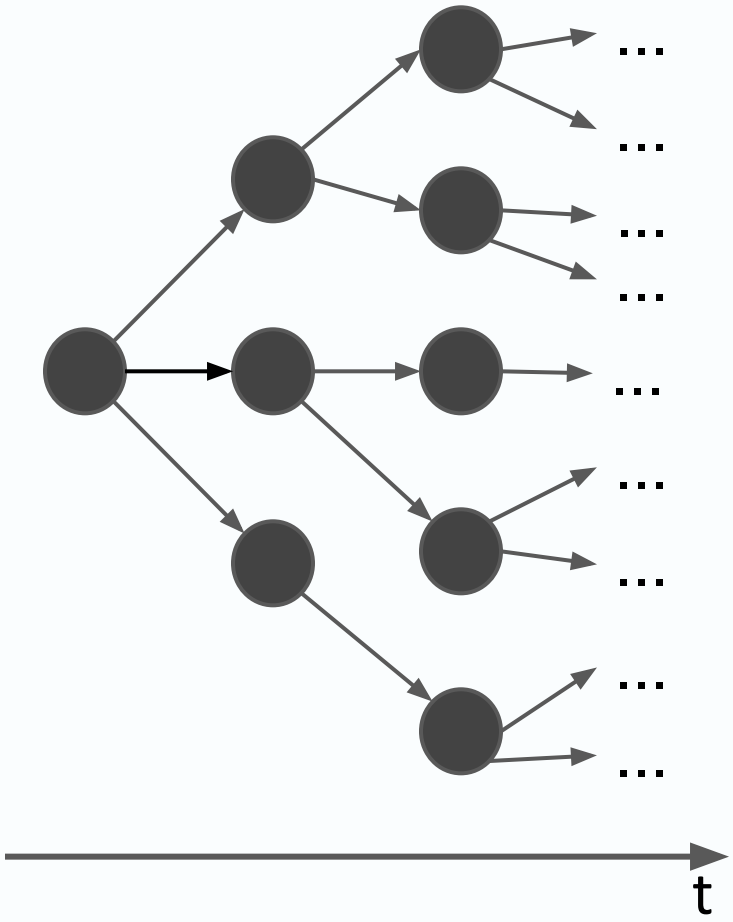
```
106
107     reaction(shutdown) {=
108         if (self->count > 0) {
109             printf("\n**** Average response time: %d.\n", self->
110         } else {
111             printf("\n**** No attempts.\n");
112         }
113     =}
114 }
115 main reactor ReflexGame {
116     p = new RandomSource();
117     g = new GetUserInput();
118     p.out -> g.prompt;
119     g.another -> p.another;
120 }
121
122
```

Tasks | Console | Error Log | Diagram

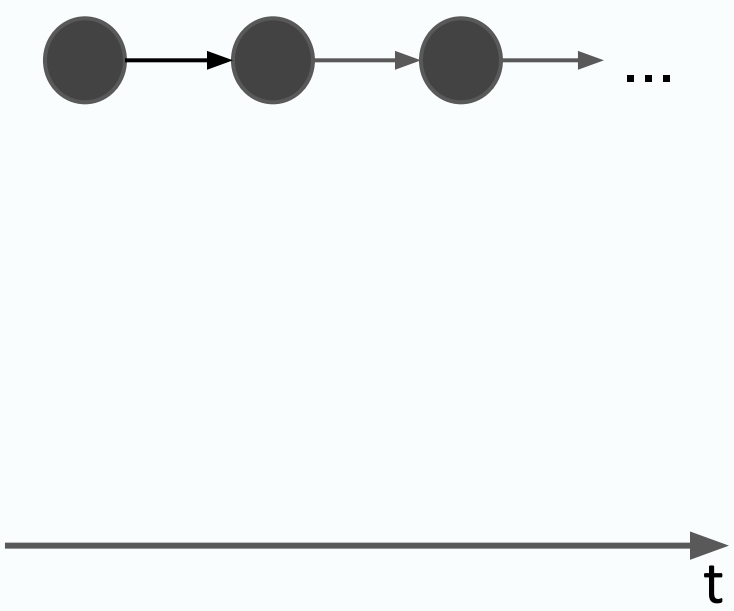


- Polyglot
- Explicit dependencies
- Discrete-event semantics
- Synchronous reactions
- Concurrency
- Distributed execution

Verifying Reactor-based Models

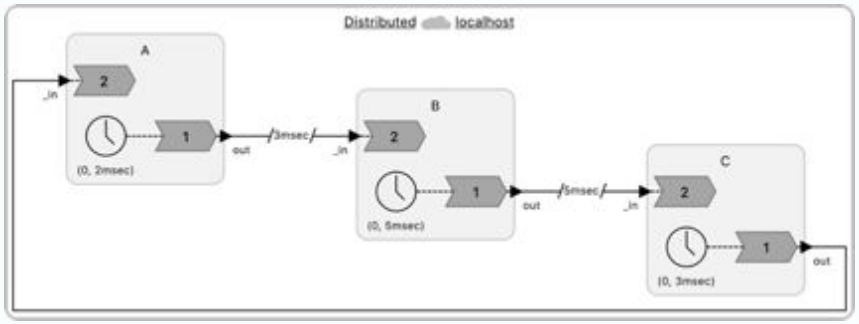


Non-deterministic exec.

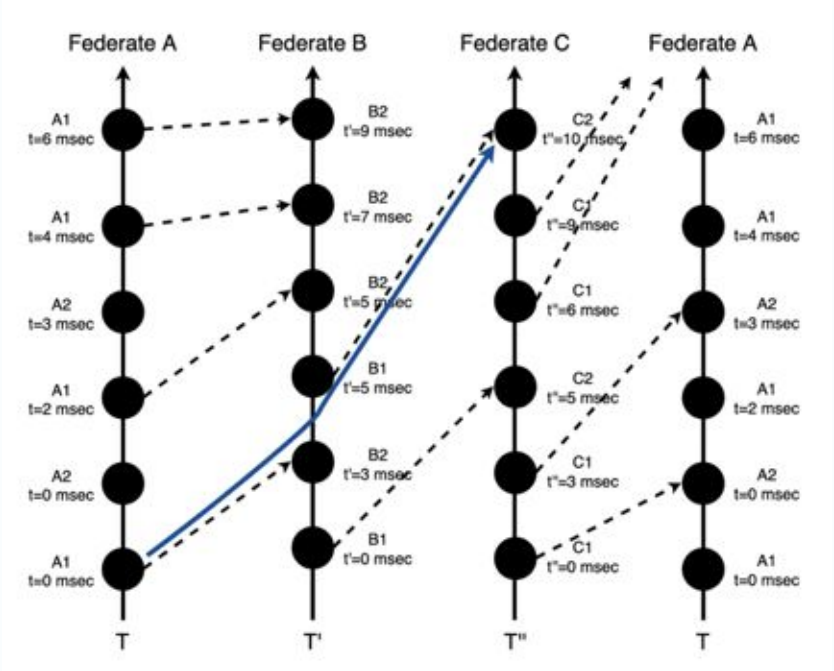


Deterministic exec.

Many Timelines in Distributed Execution



Federated Reactor System

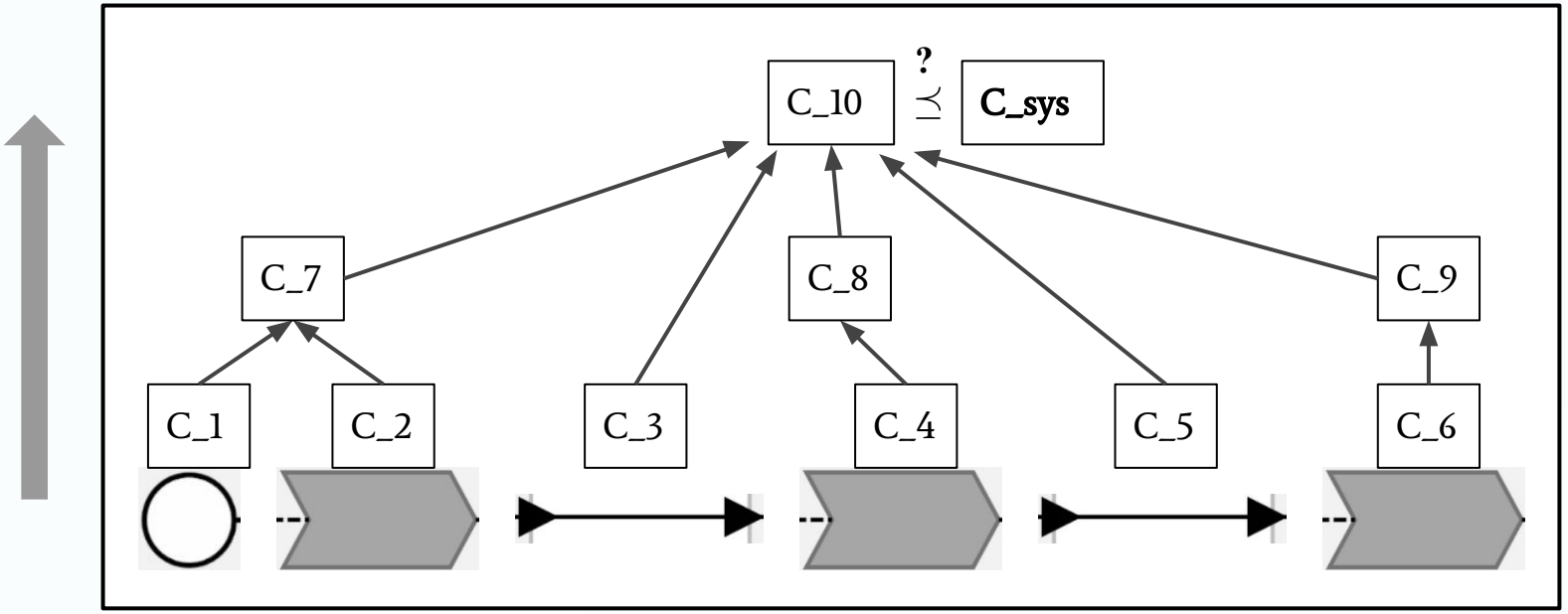


Partial Order Traces

Divide-and-Conquer Approach

Assume-guarantee contracts: $C = (A, G)$

Verification as a **bottom-up** process of abstraction:





Acknowledgements

The core Lingua Franca software development team currently consists of: Soroush Bateni, Edward A. Lee, Shaokai Lin, Marten Lohstroh, Christian Menard, Alexander Schulz-Rosengarten, and Efsane Soyer.

Others who have influenced LF with their ideas (in alphabetical order) are: Abanob Bostouros, Janette Cardoso, Jeronimo Castrillon, Julien Deantoni, Patricia Derler, Clement Fournier, Christopher Gill, Andrés Goens, Reinhard von Hanxleden, Hannes Klein, Zheng Liang, Íñigo Íncer Romeo, Marcus Rossel, Alberto Sangiovanni-Vincentelli, Martin Schoeberl, Sanjit Seshia, Marjan Sirjani, Felix Wittwer, and Sheng-Jung Yu.

The work in this paper was supported in part by the National Science Foundation (NSF), award #CNS-1836601 (Reconciling Safety with the Internet) and the iCyPhy (Industrial Cyber-Physical Systems) research center, supported by Denso, Siemens, and Toyota.

Visit our GitHub:



Github: repo.lf-lang.org
Discourse: lf-lang.discourse.group