1. Cyber-Physical Computation: Introduction

Renato Neves José Proença

CPC 2022/2023

Cyber Physical Computation

CISTER – ISEP, Porto, Portugal

U.Minho, Braga, Portugal

https://lmf.di.uminho.pt/CyPhyComp2223/

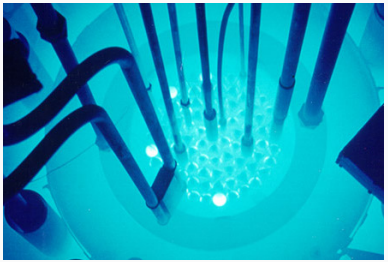
https://haslab.github.io/MFP/PCF/2223/

Universidade do Minho

Cyber-Physical Systems

Cyber-Physical Systems

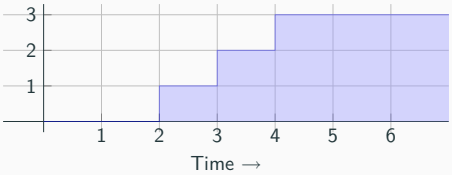
Computational devices that interact with their physical environment



Another example of a cyber-physical system

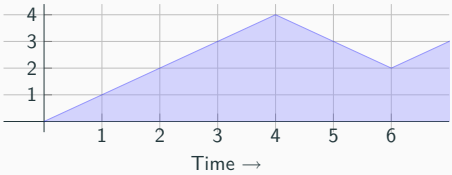
Computer Science meets Analysis

Wait Calls



$(\texttt{wait}\ 2)\,;\ \texttt{x} := \texttt{x} + 1\,;\ (\texttt{wait}\ 1)\ \ldots$

Differential Statements



$$
\begin{aligned}
&\texttt{while}\ (\texttt{true})\ \{\\
&\quad \texttt{if}\ \texttt{v} \leq 2\\
&\qquad \texttt{then}\ (\dot{\texttt{v}} = 1\ \texttt{for}\ 2)\\
&\qquad \texttt{else}\ (\dot{\texttt{v}} = -1\ \texttt{for}\ 2)\ \}
\end{aligned}
$$

Contents of the module

Cyber-Physical Computation

Genesis: David Hilbert and its
Entscheidungsproblem *(circa 1928)*

The problem fuelled the appearance of the first two models of computation ...

- Turing machines (*circa* 1936): state-based computation, part of automata theory
- λ-calculus (*circa* 1936): function-based computation, can be seen as a prototypical programming language

Contents of the module pt. I

We will study a myriad of models for cyber-physical computation

- timed automata,
- a hybrid while-language,
- λ-calculus extended with computational effects (monads!)

Contents of the module pt. I

We will study a myriad of models for cyber-physical computation

- timed automata,
- a hybrid while-language,
- λ-calculus extended with computational effects (monads!)

and often make detours through the mathematical foundations of automata and programming language theory.

Contents of the module pt. II

We will also get accquainted with a number of tools

- Uppaal – verification of real-timed systems modelled by (networks of) timed automata
- Lince – agile analysis of cyber-physical systems modelled by a hybrid while-language
- Haskell – a platform to study λ-calculus with effects

How deep will we go into the rabbit hole?

Our learning path will intersect theory and practice,
from the very basics to the state-of-the-art —
we will face current limitations and see what challenges lie ahead

Syllabus

- CSS: a simple language for concurrency
  - Syntax
  - Semantics
  - Equivalence
- Timed Automata
  - Syntax
  - Semantics (composition, Zeno)
  - Equivalence
  - UPPAAL tool
    - Specification
    - CTL and Verification
- A simple C-like language
  - Syntax
  - Semantics (operational)
- Hybrid-language: adding differential equations
  - Syntax
  - Semantics
  - Lince tool
    - Specification
    - Analysis
- Monads: semantics with computational effects

Logistics

Useful information

Relevant class material and announcements will be posted on the website periodically

https://lmf.di.uminho.pt/CyPhyComp2223
https://haslab.github.io/MFP/PCF/2223

E-mail

- nevrenato@di.uminho.pt
- pro@isep.ipp.pt

Office hours (please send an email the day before if you wish to meet):

- *Renato Neves:* Wednesday afternoon
- *José Proença:* Thursday morning

Assessment

Assessment will consist of

- 30% – an individual test at the end;
- 40% – a group assignment involving the use of the Uppaal model checker and of Haskell; and
- 15% + 15% – two sets of individual exercises to do at home.