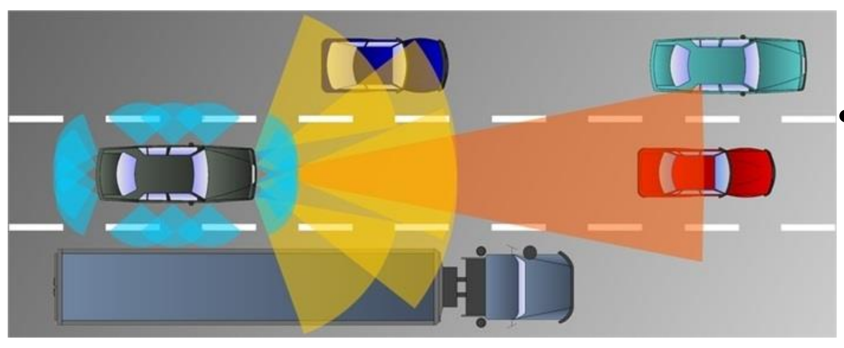
# Real Arithmetic in TLA+

## Towards proving properties in Cyber-Physical Systems

*Lancaster University*
*School of Computing and Communications*

PhD Researcher: Ovini V.W. Gunasekera
Supervisors: Dr. Antonios Gouglidis, Prof. Neeraj Suri

## Towards Secure Usage of Autonomous Systems



Collision-Avoidance safety property of Autonomous Systems

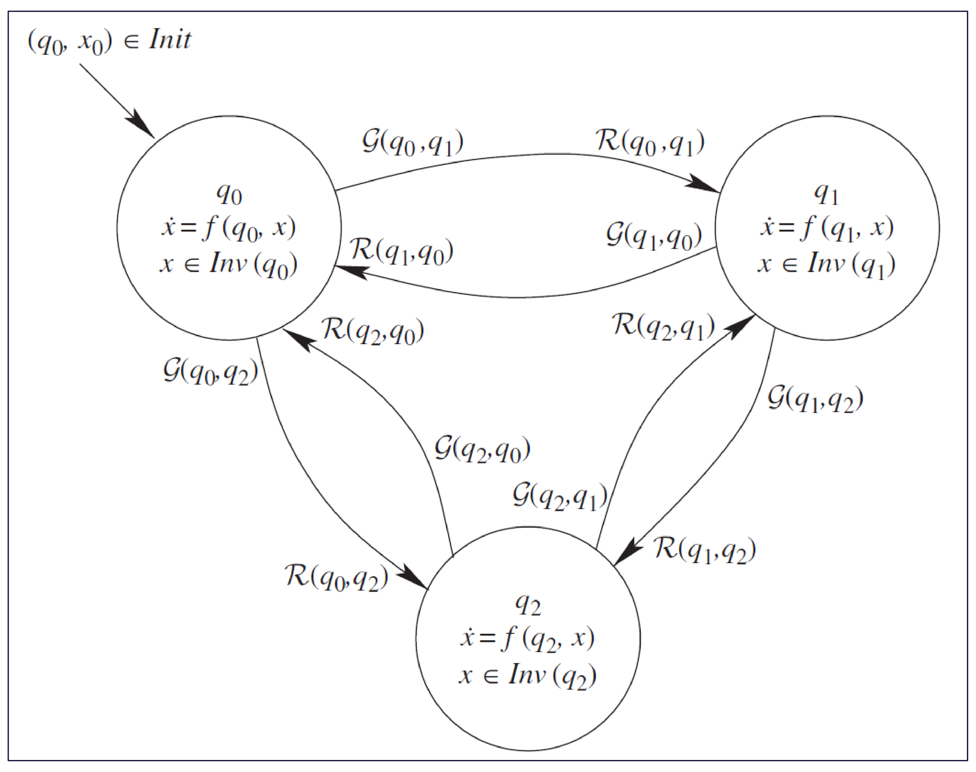
- Autonomous systems are typically considered as Cyber-Physical Systems (CPSs)
- CPSs are considered safety-critical as undetected faults can result in catastrophic consequences of serious injury or loss of life, therefore establishing **safety** is crucial
- This research work enables automatically proving safety properties of CPSs via a verification tool – TLA+ Proof Manager

## Formal Verification using TLA+

- **TLA+** is a formal specification language developed by Leslie Lamport to model systems and programs
- TLA+ toolbox is a software tool which provides an IDE for writing and verifying TLA+ specifications
- This toolbox provides support for model checking via an explicit model checker (TLC) and deductive verification via the TLA+ Proof Systems (TLAPS)

## Why TLA+

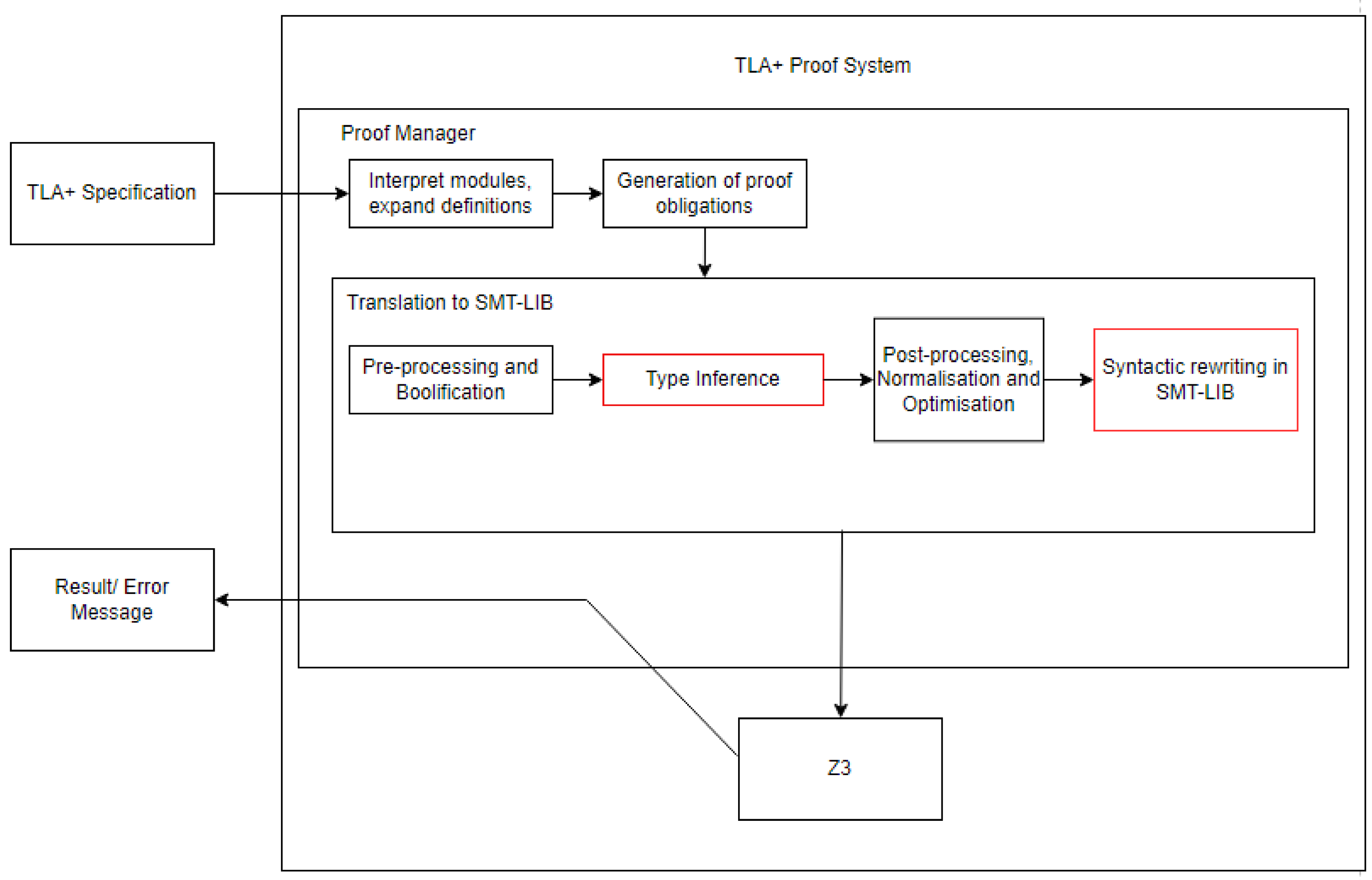
- TLA+ has gained attention from the academic community and industry and is used by major companies such as Amazon, Intel and Microsoft
- Based on Zermelo-Frankel set theory, the language enables specification and verification of wide range of systems from concurrent to distributed systems
- TLA+ is expressive enough to model hybrid systems i.e. systems which combine discrete and continuous behaviours (this includes CPSs)




Example of a CPS modelled as a hybrid automaton

- TLAPS currently supports automatically proving theorems containing integer arithmetic
- Verification of properties in CPSs require modelling continuous state evolution and thus the representation of real numbers and real arithmetic is needed
- We extended the TLA+ Proof Manager to support proving real arithmetic conjectures to ultimately facilitate proving safety properties of CPSs

## TLAPS Architecture




- TLAPS includes a proof manager which interfaces with several backend verifiers such as Z3, Isabelle and Zenon
- In extending the proof manager we enable automatically proving real arithmetic conjectures via Z3-SMT solver (SMT-LIB input) which facilitates proving of real arithmetic conjectures
- As highlighted in the TLAPS architecture diagram, we extended stages of the translation process from untyped TLA+ to multi-sorted SMT-LIB to interpret reals and real arithmetic

## Enabling translation from TLA+ to SMT-LIB

- Two types of translation
  - Typed Encoding: Type inference algorithm and TLA+ type system assigns types to TLA+ expressions
  - Untyped Encoding: Delegates type inference to SMT-solvers
- If typed encoding fails to infer types to TLA+ expressions, type inference is delegated to SMT solvers

<u>Extensions to typed encoding process</u>

- Extended TLA+ type system by introducing type *Real* and typing rules to enable the interpretation of real arithmetic
- Constraint generation and constraint solving phases of the TLA+ type inference algorithm was extended to interpret real arithmetic

<u>Extensions to untyped encoding process</u>

- Declared uninterpreted functions to embed SMT reals into a sort representing TLA+ values
- Introduced axioms to ensure soundness and consistency in translation during untyped encoding

| Operators | TLAPS | | Extended TLAPS | |
|---|---|---|---|---|
| | Real | Int | Real | Int |
| Addition(+) Subtraction(−) Multiplication(∗) | ✗ | ✓ | ✓ | ✓ |
| Integer division (\div) Modulus (%) | ✗ | ✓ | ✗ | ✓ |
| Division (/) | ✗ | ✗ | ✓ | ✗ |
| Range (..) | ✗ | ✓ | ✓ | ✓ |
| Unary minus (−) | ✗ | ✓ | ✓ | ✓ |
| Comparison ($<, >, \leq, \geq$) | ✗ | ✓ | ✓ | ✓ |
| Exponentiation (^) | ✗ | ✗ | ✗ | ✗ |

Sample of supported arithmetic operations in TLAPS and extended TLAPS

We acknowledge Dr. Andrew Sogokon in this work

This work is supported by the Engineering and Physical Sciences Research Council [grant number: EP/V026763/1]