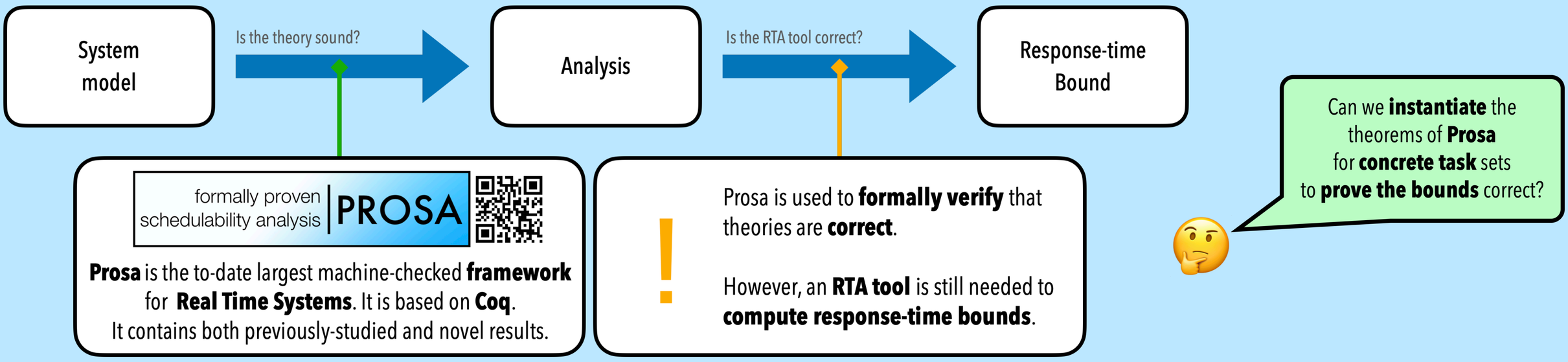
MAX PLANCK INSTITUTE
FOR SOFTWARE SYSTEMS

Automatically Generated Response-Time Proofs as Evidence of Timeliness

Marco Maida, Sergey Bozhko, and Björn B. Brandenburg

Closing the Verification Gaps of Response-time Analysis (RTA)



Introducing POET, the first foundational[1] RTA tool

The user specifies the **scheduling policy**, the **preemption model**, and the **task set** to analyze.

```
---
scheduling policy: FP     # EDF or FP
preemption model: FP      # FP, FNP, LP, FNPS
task set:

- id: 1
  period: 200
  worst-case execution time: 100
  deadline: 300

- id: 2
  period: 500
  worst-case execution time: 200
  deadline: 600
...
```

POET **generates certificates** (i.e., Coq proof scripts) to **formally prove** that the **response-time bounds** it calculates are correct.

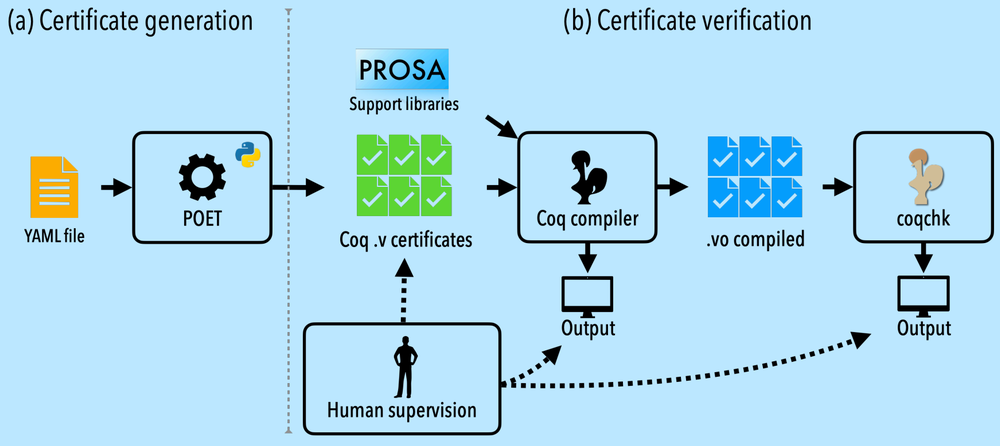
This makes of POET the first **foundational**[1] **RTA tool**.

```
Let tsk1 := {|                                          ← Task set
    task_id := 1;
    task_cost := 100;
    task_deadline := 300;
    task_period := 200;
    task_priority := 2 |}.
Let tsk2 := ...

Let tsk := tsk1.                                        ← Task under analysis

...
Let R := 100%N.                                         ← Response time
...

Lemma R_is_maximum:                                     ← Fix-point equation
  ∀ A, is_in_search_space tsk L A →
    ∃ F, rbf tsk (A + ε) + rbf ts tsk (A + F) = A + F
         ∧ F ≤ R.
Proof.
  move ⇒ A SS.                                          ← Auto-generated proof
  unfold search_space_periodic_FP, L in *.
  ...

Theorem deadline_is_respected:                          ← Final conclusion
  R <= task_deadline tsk.
Proof.
...
```

[1] A. W. Appel, "Foundational proof-carrying code," in *Proceedings 16th Annual IEEE Symposium on Logic in Computer Science*. IEEE, 2001, pp. 247–256.

Trusted computing base

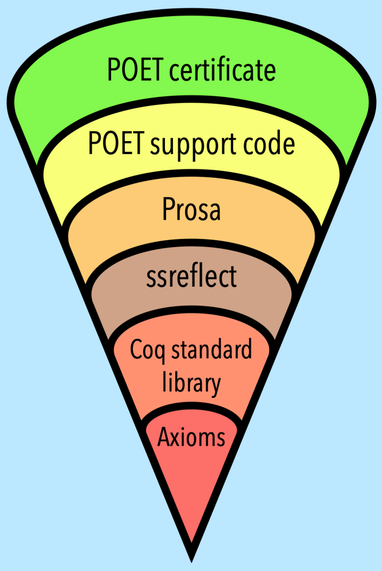
POET **does not need to be trusted**. Hence, it can be updated and integrated with other technologies as any standard Python tool.



Transparent evidence of correctness

POET's certificates are **short** and **readable** Coq proof scripts, that can be reviewed and **studied up to their fundamental axioms**.

This makes them suitable as **evidence of temporal correctness** for third-party auditors or **certification authorities**.

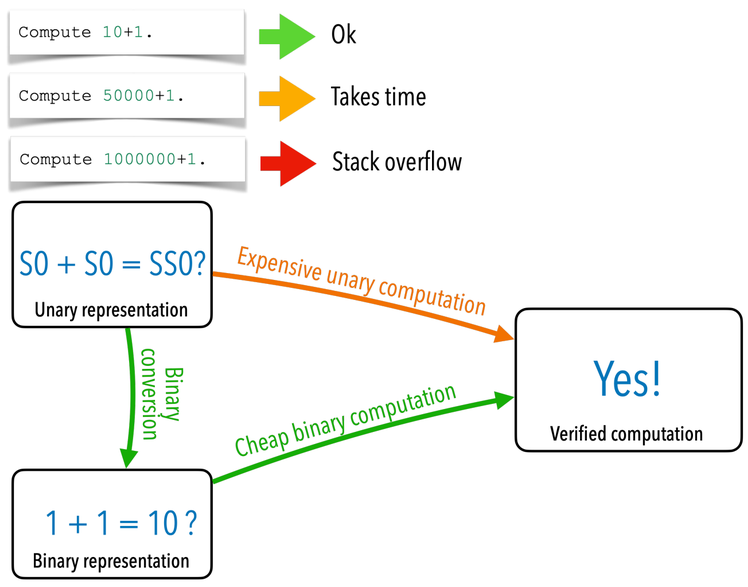


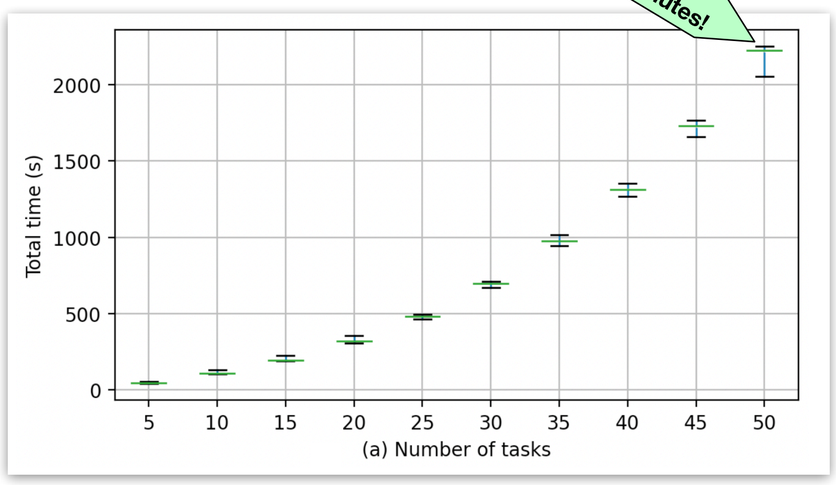
Scaling to large numerical magnitudes

POET's certificates require Coq to perform **numerical computations** using data with **nanosecond resolution** ($\sim 10^9$).

However, Coq struggles with computing, as it uses a **unary representation** of numbers. We hence converted functions to a binary form and proved them to be isomorphic to their unary counterpart using **CoqEAL**[2].

[2] C. Cohen, M. De´ne`s, and A. Mo¨rtberg, "Refinements for free!" in International Conference on Certified Programs and Proofs. Springer, 2013, pp. 147–162.





In our experiments, POET successfully verified the response-time bounds of a **periodic fixed-priority task set** comprised of **50 tasks**, taking always **less than 45 minutes**.