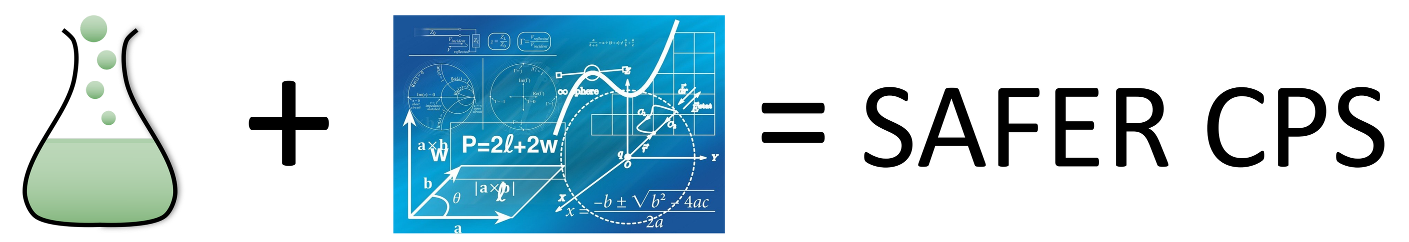
Towards Scalability of Cyber-Physical Systems Verification

Katherine Cordwell, Stefan Mitsch, André Platzer (PI), Andrew Sogokon, Yong Kiam Tan
Computer Science Department, Carnegie Mellon University Resources: https://www.ls.cs.cmu.edu/Pegasus/

Why Verification?

- It is crucial to ensure that safety-critical CPS function properly.
- Experimental testing by itself is insufficient.
- Supplement with deductive verification: *model CPS in logic and PROVE properties about the models!*



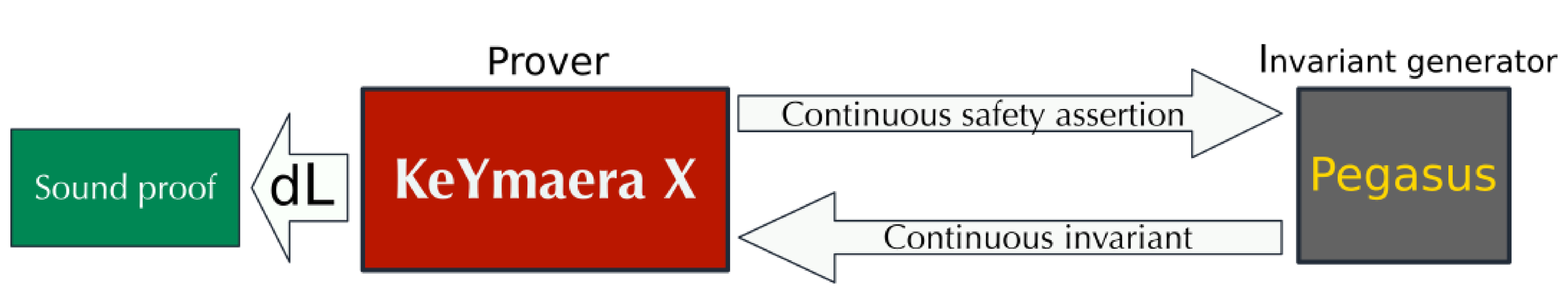
Challenges for CPS Proofs

- *Ordinary differential equations (ODEs)* are used to model continuous behavior of CPS.
- *Invariant regions* are often used to reason about ODEs. Finding these regions is quite challenging.
- Once ODEs are handled, proofs often reduce to *quantified statements* in first-order real arithmetic.
- There is a dearth of efficient formally verified support for *quantifier elimination (QE)*.

Approaches

ODEs: We further develop the tool *Pegasus*.

- Pegasus automatically generates continuous invariants for systems of ODEs.
- The generated invariants are checked by the theorem prover *KeYmaera X*.

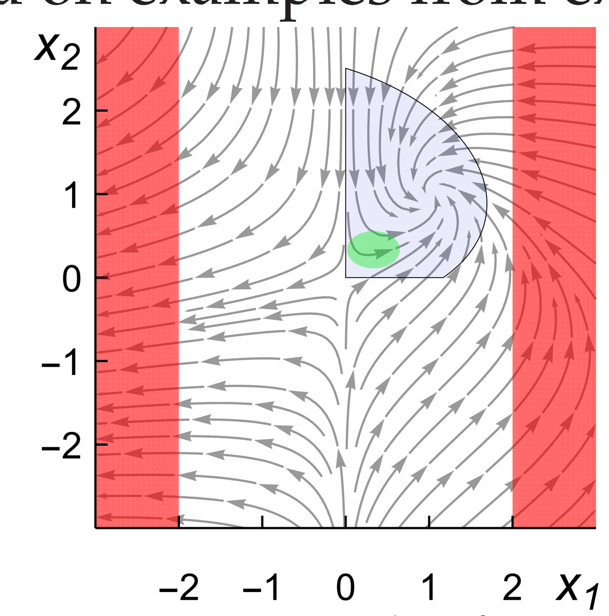


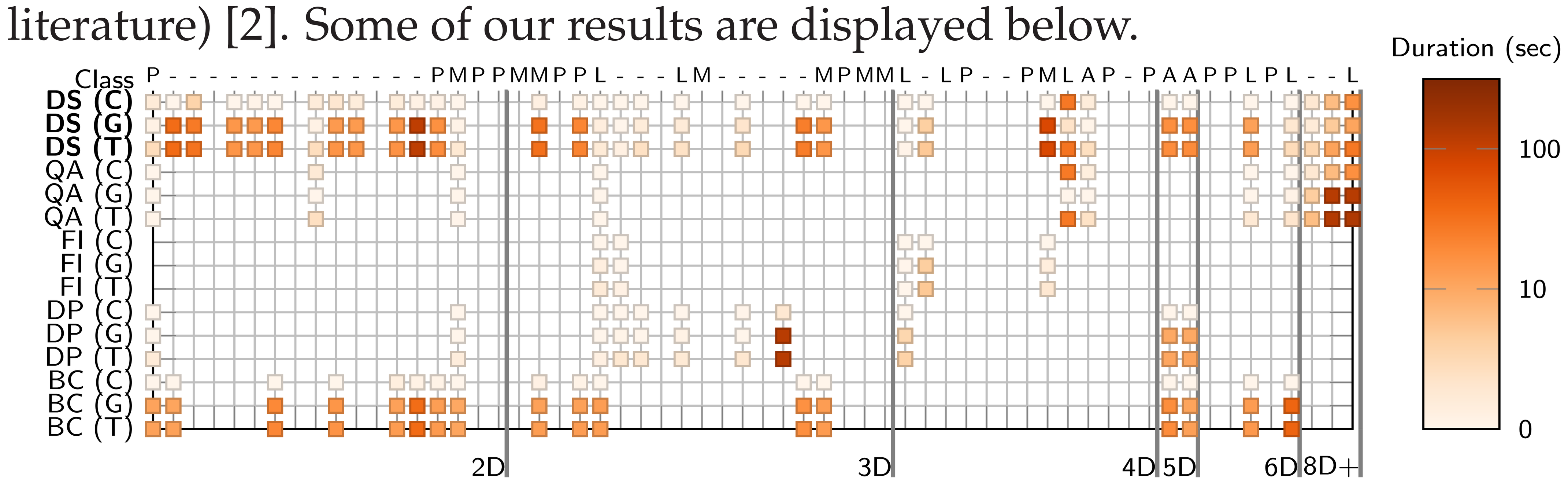
QE: We propose formally verifying the *Ben-Or, Kozen, and Reif (BKR)* QE algorithm.

- BKR has good potential for parallelism.
- In general, there is an inverse correlation between practicality and ease of formalization
- *BKR is in a potential sweet spot*; multivariate BKR builds directly on univariate BKR

Results: ODEs

We have added a number of new strategies for invariant generation and tested Pegasus on 60 new benchmarks (all based on examples from existing literature) [2]. Some of our results are displayed below.



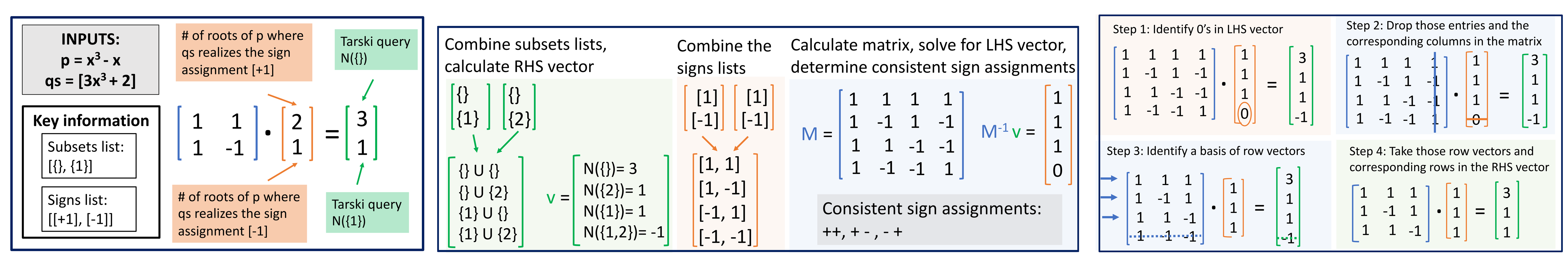


Left Figure: An example of an invariant found by Pegasus. Starting regions are in green, unsafe regions are in red. The arrows show the vector field. **Right Figure:** Each column corresponds to a benchmark problem (empty columns are unsolved). ODE classifications for each benchmark are annotated at the top of the figure (homogeneous polynomial (H), polynomial (P), linear (L), affine (A), multi-affine (M), dashes indicate same class as the enclosing labels). Abbreviations on the LHS indicate different strategies that Pegasus is trying (e.g., FI means "First Integrals" and BC means "Barrier Certificates") on total proof duration (T), generation duration (G), and checking duration (C).

Results: QE

We have formally verified the *univariate* BKR algorithm in the theorem prover Isabelle/HOL [1].

- Key step: Find the set of all *consistent sign assignments* to (univariate) $\{q_1, \ldots, q_n\}$ at the zeros of (univariate) p.
- To solve this, inductively construct a matrix equation.
- The idea of using a matrix equation dates back to Tarski; BKR makes it practical by doing a *reduction step*.



A visualized example. In all three figures, $p = x^3 - x$. The LHS figure shows the base case for $q_1 = 3x^3 + 2$. The center figure shows combining cases for $q_1 = 3x^3 + 2$ and $q_2 = 2x^2 - 1$. The RHS figure shows the reduction of the combined system.

Broader Impacts

- **Education:** Support for invariant generation helps students verify complicated models.
- **Societal:** We focus on the challenges in CPS verification to make it more practical.
- **Societal:** More practical CPS verification means more trustworthy CPS.
- **Beyond CPS:** Formally verified QE has broader applications in diverse fields, like life sciences.

References

[1] Katherine Cordwell, Yong Kiam Tan, and André Platzer. A verified decision procedure for univariate real arithmetic with the BKR algorithm. ITP 2021, to appear.

[2] Andrew Sogokon, Stefan Mitsch, Yong Kiam Tan, Katherine Cordwell, and André Platzer. Pegasus: Sound continuous invariant generation. *Form. Methods Syst. Des.*, 2021. Special issue for selected papers from FM'19.