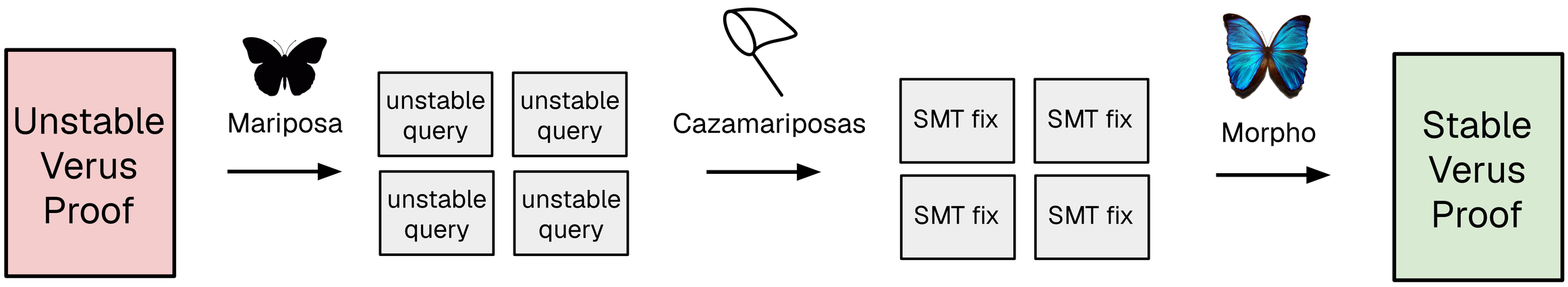
Morpho: Automatically Repairing Proof Instability in Verus

Emma Sudo[1], Natalie Neamtu[2], Amar Shah[2], Elanor Tang[2], and Bryan Parno[2]
Stanford University[1], Carnegie Mellon University[2]

What is proof instability?
Proof instability is when semantically meaningless changes to the source program cause proof failures in a verification tool. It is caused by the heuristics that SMT solvers use to reason about undecidable program properties. Instability is one of the most significant challenges in verifying real systems.

- Developers need to **fix broken proofs after updating Z3 or Verus**
- Renaming a variable or reordering lemmas can cause a **distant, unrelated proof to suddenly fail**
- The same proof on the same machine and the same solver can **randomly pass or fail**
- One developer's proof can **cause another developer's proof to fail**

The Debugging Pipeline



Mariposa and Cazamariposas work together to generate fixes on the SMT level

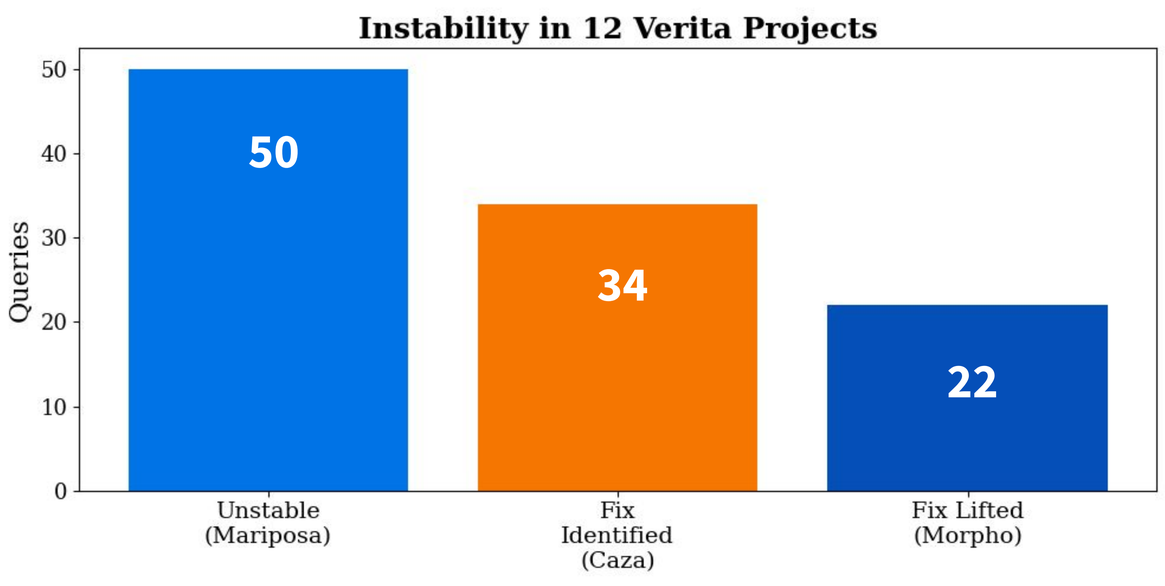
- Mariposa uses Z3 to try to solve many semantically equivalent mutants. If the mutants do not consistently pass, the query is unstable.
- Cazamariposas analyzes the Z3 failure trace and proof object to propose a SMT level fix.

Morpho lifts SMT level fixes up to Verus

- Morpho translates the three types of Cazamariposas fixes to Verus.
- Morpho **erases** SMT statements using Verus' hide() and #[verifier::opaque].
- Morpho **inserts** SMT statements using Verus' reveal(), assert(), and broadcast use.
- Morpho **skolemizes** SMT statements using Verus' choose.

Morpho addresses the instability caused by "fuel-gating" spec function definitions

- For Verita, the majority of the Cazamariposas-generated fixes are related to the way Verus uses a quantifier to define the default number of times Z3 can unfold a definition. The order that Z3 searches for the definitions it needs for a proof is nondeterministic.
- Morpho "erases" the fuel quantifier with a new Verus attribute #[verifier::fuel_default_off]. Now, all the necessary definitions will need to be marked for unfolding. Then, Morpho "inserts" the necessary fuel assertions by using reveal() and broadcast use on all of the definitions that Z3 actually used to solve the proof.



Conclusion

Proof instability makes it hard to maintain verified systems, but Morpho leverages prior work to repair instability without developer involvement.

Future Work

- Investigate VeruSAGE to find more examples of user (or LLM) generated instability
- Use AirLift to lift arbitrary SMT expressions up to Verus
- Optimize Mariposa and Cazamariposas to speedup the repair process