Verified Programming in Gugu

Aaron Stump[1] Morgan Deters[2] Adam Petcher[3]
Todd Schiller[3] Timothy Simpson[3]

[1]Computational Logic Center
CS, The University of Iowa

[2]LSI, Universitat Politècnica de Catalunya, Spain

[3]CSE, Washington University in St. Louis

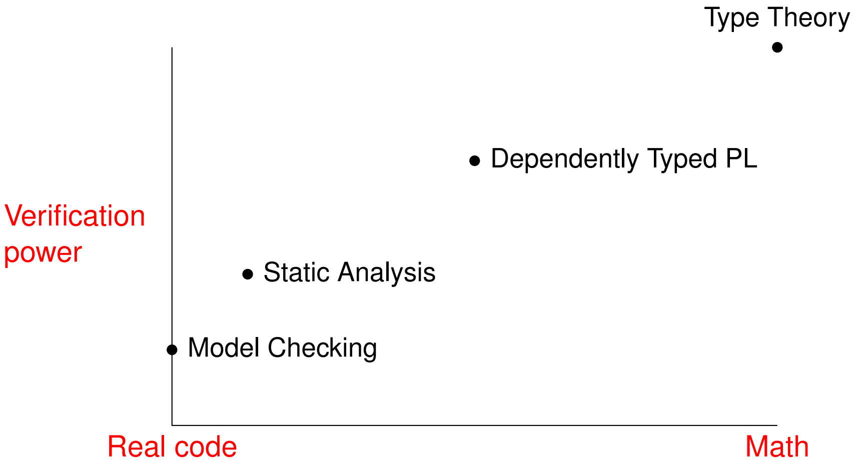
Funding from NSF CAREER.

A Vexing Continuum

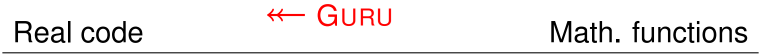
Real code	Math. functions
concurrent	sequential
imperative	pure
general recursive	terminating

Where is your verification method?

Plotting Some Approaches



The GURU Approach



General recursion
Dependently typed programs
External theorems about programs
Mutable state
No concurrency
No aliasing (yet)

Conclusion

- GOLFSOCK: towards verified, efficient language tools.
- OpTT makes this easier:
  - Not required *a priori* to prove termination.
  - Reason about code with annotations dropped.
  - Use dependent types for big functions (`check`, 1200 lines).
  - Supports functional modeling.
- Onward towards verified, efficient software!

www.guru-lang.org