

EIA0082849: ITR: Formal Methods Education and Programming Effectiveness

University of Oklahoma

Investigator: Rex Page

Discrete Mathematics

Learning Goals

What we want to know

- Does using logic to reason about software make programmers better?
- Will anyone use 350 slides, 150 exam questions and solutions, 100 homework problems/solutions?
- Will stats convert skeptics? Or just bolster the choir?

Mathematical logic and proof

- ✓ Predicate calculus and natural deduction
- ✓ Boolean algebra and equational reasoning
- ✓ Inductive definitions
- ✓ Proof by induction on numbers, trees, and loops
- ✓ Verify software properties — correctness, resource usage

Fundamental data structures

- ✓ Sets and sequences
- ✓ Trees and graphs
- ✓ Functions and relations

*proofs galore!
proofs galore!
proofs galore!*

All examples relate discrete math to programming

- ✓ Two-thirds of material is logic, functions, and induction
- ✓ One third is sets, graphs, relations, and counting

What we've found out

Above-average students with experience in reasoning about software perform better in a subsequent programming course, Data Structures, than students who lack such experience.

Statistical results support this conclusion with 98% confidence.

Web search key: **Beseme Project**

Statistical Results

