

# Final Meeting

- Favorite Papers
- Least Favorite Papers
- Program Committee Operations
  - The “Champion” Theory
- Themes and Analyses
  - Paper Weaknesses, Paper Generation

# Theme: Symbolic Execution

- ESC/Java
- Angelix
- KLEE
- From Tests To Proofs
  
- When should you use it? What are the weaknesses?

# Theme: Large Empirical Evaluation

- Producing Wrong Data
- Fair and Balanced
- ESC
- Bugs As Inconsistent
- FindBugs
- Tarantula
- GenProg
- Uniqueness / Naturalness
- (cf. FFTW)

When should  
you use it?

# Theme: “Too Formal”

- ESC
- CCured
- Random Interpretation
- DIG
- Life, Death, Critical

(but *not* “Precise Interprocedural” or “Program Verification to Program Synthesis”)

# Theme: “Run The Tests”

## (Dynamic Analysis)

- Delta Debugging
- Tarantula
- GenProg
- PAR
- SPR
- Angelix
- muTest / EvoSuite
- Daikon
- DIG
- FFTW

When should you use it? What are the weaknesses?

# Paper Idea Generation

- KLEE = DART + ESC
- GenProg = FFTW + Tarantula + DeltaDebugging
- “Debugging helping programmers?” = Tarantula + “Producing Wrong Data”
- Angelix = GenProg + (KLEE | ESC)
- SPR = GenProg + “Improving with Path Profiles”
- Pin = ATOM + “Fast Breakpoints”
- “Life Death Critical” = “Precise Interprocedural” + “Limits of Generic Recovery”

# Concluding Thoughts

- “Seems obvious in retrospect”
  - This is a good thing
- “How did this get in [when mine did not]?”
  - Avoid “should”
- Claim: formal papers, on average, have short-term benefits but lack long-term impact
  - Exceptions exist
- Recognize recurring themes
  - Identify weaknesses
  - Make your own papers