Automatic Device Driver Synthesis

Adam Walker
Motivation

• **70%** of OS failures are caused by driver bugs
• Drivers contain **3-7** times more bugs per loc than the rest of the kernel
• **70%** of OS code is in device drivers
• Solution: automatically synthesize device drivers
  – Correct by construction
Overview

- Separation of concerns
- Reuse
  - Specify once, synthesise many

Formal OS interface spec

requests

driver.c

device commands

Formal device spec
Approach

- Formalise the problem as a two player game (driver vs OS and device)
- Specs synchronize on shared events
- Driver objectives are temporal logic formulas
- Driver synthesis is controller synthesis problem on finite state machines
Challenges

• State space explosion
  – Symbolic state space representation
  – Predicate abstraction

• Synthesis with imperfect information
  – Driver cannot directly observe device state transitions

• Efficient code generation

• Verification: is the synthesized driver correct?
  – Errors in the specification
  – Errors in the synthesis tool