

Limitations of Formal Methods

Requirements Engineering II

Michał Kołtonik Serge Hänni

29.11.2010

Program verification history

- Robert Floyd
- 40 years of formal methods
- Hoare logic,
- Z, Petri nets, ...

Problems

- the same program design errors persist for 40 years
- the industry rarely formal methods
- the gap between formal methods and practical software development is growing
- developers
- research papers focused on mathematical part
- research papers do not show how to deal with complexity of software systems
- developers avoid formal methods
- education gaps (software engineers vs. software developers)

Why it fails

- models more complex than the code
- it's not easier to write or understand the model than the code
- models oversimplify problems by ignoring many ugly details
- often only author of the model is able to answer the questions about it

What is to be considered?

- variable in mathematics vs. identifier in program
 - pointers, arrays
 - representation of states should be rethought
- mathematical expressions
 - easy to read and write correctly only for simple expressions and continuous functions
 - new forms of expressions designed for use with complex programs needed
- there were no *hidden state* notion when formal methods were created
- program termination
- execution time
 - no time concern in formal methods
 - program running too slow shouldn't be proofed correct

- Hoare's axioms could be replaced by relational algebra
- direction of analysis
- dealing with side effects
- nondeterminism should be set as a normal case
- formal methods shouldn't use simplified models
- techniques of deriving mathematical model from program code needed
 - natural language documentation often too incomplete, imprecise and bad-structured
- formal methods to provide documentation to facilitate understanding the program
- correctness proof vs. properties calculation

Difficulties in Formalization

- physical world descriptions
- logic can show presence of errors, but not their absence
- need of reasoning about physical world
- state compatibility between software and environment

Addressing Difficulties in Formalization

- physical world descriptions
 - Four-variable model: model I/O
 - Rely and guarantee: guarantee certain properties of environment, relying on other
 - Reality variables: specifies environment as ghost variables
- decompose problems
- known subclasses of problems
- known device types