Coverage and Games in Model-Based Testing

Petra van den Bos
Software is everywhere
Software testing

- Bugs in programs may lead to inconveniences, costs, or even accidents.
- Use software testing to detect these bugs.
- My thesis is about software testing
Testing a coffee machine

Inputs:

Outputs:
Model-based testing

- Creating tests costs a lot of time.
Model-based testing

- Creating tests costs a lot of time.
- Test automation: model-based testing
  - Construct a model of the program.
  - Generate tests from the model.
Model-based testing

- Creating tests costs a lot of time.
- Test automation: model-based testing
  - Construct a model of the program.
  - Generate tests from the model.
An example test

Model:

Example test:

Petra van den Bos
An example test

Model:

Example test:

This test covers all transitions
An example test

This test covers all transitions
Is this test sufficient for finding bugs?
Test selection: model coverage

- Goal: find many bugs in limited time.
Test selection: model coverage

• Goal: find many bugs in limited time.
• Method: select a limited number of versatile tests
Test selection: model coverage

• Goal: find many bugs in limited time.
• Method: select a limited number of versatile tests
• Model coverage test selection:
  – Tests that cover all parts of the model
Test selection: model coverage

- **Goal**: find many bugs in limited time.
- **Method**: select a limited number of versatile tests.
- **Model coverage test selection**:
  - Tests that cover all parts of the model.
Models with data and constraints

- Coffee machine model with data and constraints:
- Data actions: ‘insert 10 cents’ or ‘insert 1 euro’
Models with data and constraints

• Coffee machine model with data and constraints:
  • Data actions: ‘insert 10 cents’ or ‘insert 1 euro’
  • Constraints: ‘coins less than 10 cents not accepted’ and ‘coffee costs 50 cents’
$n$-Complete testing

- Detect any bug in program of size $\leq n$.
- Estimate size $n$ of program
\( n \)-Complete testing

- Detect any bug in program of size \( \leq n \).
- Estimate size \( n \) of program

Model:

Actual behaviour (free coffee):

Petra van den Bos
Testing as a Game

Program
produces outputs
tries to hide bugs

Tester
provides inputs
tries to find bugs
Summary

• Chapter 2: \textit{n-}Complete test suites
• Chapter 3: Efficient building block for \textit{n}-complete test suites (state identification)
• Chapter 4: Coverage test selection for models with data and constraints
• Chapter 5: Model-based testing is a game
• Chapter 6: Win game by dodging adversary program actions