MODEL-BASED, MUTATION-DRIVEN TEST CASE GENERATION VIA HEURISTIC-GUIDED BRANCHING SEARCH

Andreas Fellner

FMCAD Student Forum
Wien, October 4th 2017
TEST CASE GENERATION WITH MoMuT
TEST CASE GENERATION WITH MoMuT

Abstract Model
UML / Event-B
TEST CASE GENERATION WITH MoMuT

Abstract Model
UML / Event-B

Executable Model
Action System
TEST CASE GENERATION WITH MoMuT

Abstract Model
UML / Event-B

Executable Model
Action System

Automatic Translation
TEST CASE GENERATION WITH MoMuT

Abstract Model
UML / Event-B

Executable Model
Action System

Automatic Translation

Mutated Action Systems
TEST CASE GENERATION WITH MoMuT

Abstract Model
UML / Event-B

Executable Model
Action System

Automatic Translation

Mutated Action Systems
TEST CASE GENERATION WITH MoMuT

Abstract Model
UML / Event-B

Executable Model
Action System

Automatic Translation

Mutated Action Systems

Original: \( c := c + 1 \)

Mutation: \( c := c \)
TEST CASE GENERATION WITH MoMuT

Abstract Model
UML / Event-B

Executable Model
Action System

Automatic Translation

Mutated Action Systems

Original: \( c := c + 1 \)
Mutation: \( c := c \)

Original: if signal > 100
... Mutation: if false
...
TEST CASE GENERATION WITH MoMuT

Abstract Model
UML / Event-B

Executable Model
Action System

Automatic Translation

Mutated Action Systems

Original:  \( c := c + 1 \)
Mutation:  \( c := c \)

Original:  if signal > 100
Mutation:  if false

Test Cases
TEST CASE GENERATION WITH MoMuT

Abstract Model
UML / Event-B

Executable Model
Action System

Automatic Translation

Mutated Action Systems

Original:  \( c := c + 1 \)
Mutation:  \( c := c \)

Original:  if signal > 100 ...
Mutation:  if false ...

Test Cases

Input/Output Sequence
TEST CASE GENERATION WITH MoMuT

Abstract Model
UML / Event-B

Executable Model
Action System

Branching Search

Automatic Translation

Mutated Action Systems

Original: \( c := c + 1 \)
Mutation: \( c := c \)

Original: \( \text{if signal > 100} \)
Mutation: \( \text{if false} \)
TEST CASE GENERATION WITH MoMuT

Abstract Model
UML / Event-B

Executable Model
Action System

Branching Search

Automatic Translation

Test Cases

Mutated Action Systems

Mutants killed / alive / equivalent?

Original:  $c := c + 1$
Mutation:  $c := c$

Original:  if signal > 100
Mutation:  if false
TEST CASE GENERATION WITH MoMuT

Abstract Model
UML / Event-B

Executable Model
Action System

Branching Search

Automatic Translation

Mutated Action Systems

Test Cases

Mutants killed / alive / equivalent?

Original: `c := c + 1`
Mutation: `c := c`

Original: `if signal > 100` ...
Mutation: `if false` ...

Original: `if signal > 100` ...
Mutation: `if false` ...

Mutant Killed:
- Same Input
- Different Output

Input/Output Sequence
BRANCHING SEARCH AND MUTATION KILLING
BRANCHING SEARCH AND MUTATION KILLING

• Explore model in branches
BRANCHING SEARCH AND MUTATION KILLING

• Explore model in branches
BRANCHING SEARCH AND MUTATION KILLING

• Explore model in branches
BRANCHING SEARCH AND MUTATION KILLING

• Explore model in branches
BRANCHING SEARCH AND MUTATION KILLING

- Explore model in branches
BRANCHING SEARCH AND MUTATION KILLING

• Explore model in branches
  • Each branch is explored in parallel
BRANCHING SEARCH AND MUTATION KILLING

- Explore model in branches
  - Each branch is explored in parallel
- Explore mutated models in parallel
BRANCHING SEARCH AND MUTATION KILLING

• Explore model in branches
  • Each branch is explored in parallel

• Explore mutated models in parallel
BRANCHING SEARCH AND MUTATION KILLING

- Explore model in branches
  - Each branch is explored in parallel
- Explore mutated models in parallel
BRANCHING SEARCH AND MUTATION KILLING

- Explore model in branches
  - Each branch is explored in parallel

- Explore mutated models in parallel
  - Only explore relevant parts
BRANCHING SEARCH AND MUTATION KILLING

• Explore model in branches
  • Each branch is explored in parallel

• Explore mutated models in parallel
  • Only explore relevant parts

• Set of heuristics guiding branching search
BRANCHING SEARCH AND MUTATION KILLING

- Explore model in branches
  - Each branch is explored in parallel

- Explore mutated models in parallel
  - Only explore relevant parts

- Set of heuristics guiding branching search
  - Where to start new branches
  - How to expand branches
BRANCHING SEARCH AND MUTATION KILLING

- Explore model in branches
  - Each branch is explored in parallel

- Explore mutated models in parallel
  - Only explore relevant parts

- Set of heuristics guiding branching search
  - Where to start new branches
  - How to expand branches

- Construct test cases from exploration graph
BRANCHING SEARCH AND MUTATION KILLING

• Explore model in branches
  • Each branch is explored in parallel

• Explore mutated models in parallel
  • Only explore relevant parts

• Set of heuristics guiding branching search
  • Where to start new branches
  • How to expand branches

• Construct test cases from exploration graph
  • Prune irrelevant exploration steps
BRANCHING SEARCH AND MUTATION KILLING

• Explore model in branches
  • Each branch is explored in parallel

• Explore mutated models in parallel
  • Only explore relevant parts

• Set of heuristics guiding branching search
  • Where to start new branches
  • How to expand branches

• Construct test cases from exploration graph
  • Prune irrelevant exploration steps
**BRANCHING SEARCH AND MUTATION KILLING**

- Explore model in branches
  - Each branch is explored in parallel

- Explore mutated models in parallel
  - Only explore relevant parts

- Set of heuristics guiding branching search
  - Where to start new branches
  - How to expand branches

- Construct test cases from exploration graph
  - Prune irrelevant exploration steps
BRANCHING SEARCH AND MUTATION KILLING

- Explore model in branches
  - Each branch is explored in parallel

- Explore mutated models in parallel
  - Only explore relevant parts

- Set of heuristics guiding branching search
  - Where to start new branches
  - How to expand branches

- Construct test cases from exploration graph
  - Prune irrelevant exploration steps
  - Map test cases to mutant kills
SUMMARY

• Model based testing
  • Test high level behavior based on abstract description of the system
• Mutation testing
  • Connect tests to faults
  • Prune irrelevant test steps
• Demanding models from industrial context, thus emphasis on scalability

• Branching Search
  • Fully leverage parallelism
  • Flexibility through set of heuristics
  • Shorter and more effective tests
FUTURE WORK

• Distance metric based on mutant constraints

• Designated strong killing algorithm

• Semi symbolic methods
  • Dynamic symbolic execution

• Unfoldings, Partial Orders & Petri Nets

• Static analysis
  • Better estimation of state space
  • Eliminate equivalent mutants