SWINBURNE UNIVERSITY OF TECHNOLOGY

Prioritization of Test Cases in MUMCUT Test Sets: An Empirical Study

Presented by M. F. Lau

Centre for Software Engineering
School of Information Technology
Swinburne University of Technology
Melbourne, AUSTRALIA
http://www.it.swin.edu.au/centres/cse

Overview

- Boolean specifications
- Types of Fault
- MUMCUT Strategy
- Test Case Prioritization
- Experiment and Results
- Conclusions and Future work

Boolean Specifications

Example:

$$S = ac + abd + af + be$$

where *a*, *b*, *c*, *d*, *e*, and *f* are Boolean variables

SWINBURNE UNIVERSITY OF TECHNOLOGY

Boolean Specifications (cont'd)

- A Boolean variable is one which has a value of either *True* (1) or *False* (0).
- A Boolean formula connects Boolean variables with logic operators: **and** $\cdot$, **or** $+$, **not** $-$, etc.
- A Boolean formula *S* represents a function

$$f : \mathbf{B}^n \rightarrow \mathbf{B} \quad \text{where} \quad \mathbf{B} = \{ 0, 1 \}$$

- With *n* Boolean variables, there are 2^{2^n} distinct Boolean functions.

SWINBURNE UNIVERSITY OF TECHNOLOGY

Boolean specifications (cont'd)

- Complex conditions in software are often specified in the form of a Boolean formula.
- Input domain: n-dim Boolean space $\mathbf{B}^n$
- Requires all 2^n test points to distinguish a Boolean function from another
- **Problem**: *How to select a 'small' subset of test points to detect certain types of fault?*

SWINBURNE UNIVERSITY OF TECHNOLOGY

Types of Fault

- Expression Negation Fault (ENF)
  - The whole expression is negated
- Literal Negation Fault (LNF)
  - A literal in a term is negated
- Term Omission Fault (TOF)
  - A term is omitted
- Literal Omission Fault (LOF)
  - A literal in a term is omitted
- Operator Reference Fault (ORF)
  - An operator is replaced by another operator

Types of Fault (cont'd)

- Literal Insertion Fault (LIF)
  - A literal is inserted into a term
- Literal Reference Fault (LRF)
  - A literal is replaced by another literal

Types of Fault – Example

Original spec.	$S = ab + cd$
ENF	$I = \overline{ab + cd}$
LNF	$I = \overline{ab} + cd$
TOF	$I = ab$
LOF	$I = a + cd$
ORF	$I = abcd$ or $I = ab + c + d$
LIF	$I = ab + acd$
LRF	$I = ad + cd$

SWINBURNE UNIVERSITY OF TECHNOLOGY

Types of Fault (cont'd)

- *S* and *I* may be equivalent
  - e.g. $S = a + b$, $I = a + ab$
- Test cases that detect the non-equivalent implementations are good test cases.
  - e.g. $S = ab + cd$, $I = a + cd$
  - Good: 1000, 1001, ...
  - Not good: 0011, 1011, ...

SWINBURNE UNIVERSITY OF TECHNOLOGY

True Point

- Assume that *S* is in *irredundant disjunctive normal form* (e.g. $S = ab + cd$)
- True point: point such that *S* evaluates to *true* (1)
  - TP = { 1100, 1110, 1101, 1111, 0011, 0111, 1011 }
- Unique true point of i-th term: point such that only the i-th term of *S* evaluates to true
  - UTP(1) = { 1100, 1110, 1101 }
  - UTP(2) = { 0011, 0111, 1011 }

SWINBURNE UNIVERSITY OF TECHNOLOGY

False Point

- Example: $S = ab + cd$
- False point: point so that S evaluates to false (0)
  - FP = {0100,0101,0110,1000,1001,1010,0001,0010,0000}
- Near false point of j-th literal of i-th term: false point that $p_{i,j}$ evaluates to true where $p_{i,j}$ is the term obtained by negating the j-th literal of the i-th term
  - NFP(1,1)={0100,0101,0110} NFP(1,2)={1000,1001,1010}
  - NFP(2,1)={0001,0101,1001} NFP(2,2)={0010,0110,1010}

SWINBURNE UNIVERSITY OF TECHNOLOGY

MUMCUT Strategy

- A strategy by combining three different strategies
  - MUTP, MNFP and CUTPNFP strategy
- MUTP strategy
  - Select test points in UTP(i) such that every truth value of every missing variable is covered
  - e.g. { 1101, 1110, 0111, 1011 } ($S = ab + cd$)
  - Can detect ENF, LNF, TOF, and LIF
- MNFP strategy
  - Select test points in NFP(i,j) such that every truth value of every missing variable is covered
  - e.g. { 0101, 0110, 1001, 1010 } ($S = ab + cd$)
  - Can detect ENF, LNF, and LOF

MUMCUT Strategy (cont'd)

- CUTPNFP strategy
  - Select a unique true point in UTP(i) and a near false point in NFP(i,j) such that the two points differ only at the j-th literal of the i-th term
  - e.g. { 1101, 0101, 1001 , 0111, 0101, 0110 } ($S = ab + cd$)
- The MUMCUT strategy can detect all seven types of fault

MUMCUT Strategy (continued)

- A strategy for generating test cases
  - No guidelines on execution order
- Any particular execution order *can detect faults earlier in testing*?
  - MUTP strategy
  - MNFP strategy
  - CUTPNFP strategy

SWINBURNE UNIVERSITY OF TECHNOLOGY

Test Case Prioritization, TCP

- Faster detection of more faults facilitates earlier debugging and fault removal
- Problem:
  - What are the effects, if any, of the order of executing test cases that collectively satisfy the MUMCUT strategy on the rate of fault detection during testing?
- Two dimensions of assessment:
  - Rate of fault detection
  - Time for fault detection (wrt the percentage of test set)
- Metric used:
  - weighted Average of the Percentage of Faults Detected (APFD)

Test Case Prioritization, TCP (cont'd)

- Why study Black-box test cases?
  - Guidelines are independent of source code
- Why MUMCUT?
  - Is a fault-based strategy
  - Exists a test set that satisfies MUMCUT strategy
  - Contains different groups of test cases

Test Case Prioritization (cont'd)

- Previous results on prioritizing MUMCUT test cases
  - CNU order is better than random and serial
- Is that just a coincidence?
- Different possible orders
  - CNU (CUTPNFP, MNFP, MUTP)
  - CUN
  - NCU
  - NUC
  - UCN
  - UNC

Experiment

- Subject under study: Boolean specifications derived from TCAS II (Traffic Collision Avoidance System)
- Number of Boolean variables: 5 – 13
- For most specifications except a few, there is a large number of MUMCUT test sets
- Randomly pick 1000 MUMCUT test sets
- Monitor the executions of test cases to compute the APFD

Experimental Result

- UCN order gives the highest average values (APFD) over the 20 Boolean specifications under study
- The U-group is consistently better than the C-group which in turn is better than the N-group
- This is differently than as expected from Kuhn's fault hierarchy (VRF > VNF > ENF)
  - C-group first, U-group/N-group later

SWINBURNE UNIVERSITY OF TECHNOLOGY

Conclusions and Future Work

- Test cases executed in the “U–C–N” order yield highest APFD values.
- Need further investigation on the fault-class hierarchy based on the observations from the experiments.

SWINBURNE UNIVERSITY OF TECHNOLOGY