3. Test Generation Strategies V

Based on requirements

Andrea Polini

Software Engineering II – Software Testing
MSc in Computer Science
University of Camerino

December 2nd, 2014

Predicate testing criteria

Three common criteria:

- BOR (Boolean Operator): A test set $\mathscr{T}$ that satisfied the BOR-testing criterion for a compound predicate p_r, guarantees the detection of single or multiple Boolean operator faults in the implementation of p_r. $\mathscr{T}$ is referred to as a BOR-adequate test set and sometimes written as $\mathscr{T}_{BOR}$.
- BRO (Boolean and relational Operator): A test set $\mathscr{T}$ that satisfied the BRO-testing criterion for a compound predicate p_r, guarantees the detection of single or multiple Boolean operator and relational operator faults in the implementation of p_r. $\mathscr{T}$ is referred to as a BRO-adequate test set and sometimes written as $\mathscr{T}_{BRO}$.
- BRE (Boolean and relational expression): A test set $\mathscr{T}$ that satisfied the BRE-testing criterion for a compound predicate p_r, guarantees the detection of single or multiple Boolean operator, relational operator and arithmetic expression faults in the implementation of p_r. $\mathscr{T}$ is referred to as a BRO-adequate test set and sometimes written as $\mathscr{T}_{BRE}$.

Generating BOR, BRO, BRE adequate tests

A predicate constraint C for predicate p_r is a sequence of $n + 1$ boolean and relational symbols.
A test case t satisfies C for predicate p_r, if each component of p_r satisfies the corresponding constraint in C when evaluted against t.
e.g.: given $p_r = b \wedge r < s \vee u \geq v$ and $C : (t, =, >)$ the following test case satisfies C: $<b = true, r = 1, s = 1, u = 1, v = 0>$

There exist algorithms for the generation of adequate tests given constraints on the predicate. They are based on the definition of:

- Cartesian product of sets
- *onto* set product operator
- $AST(p_r)$

Generating the BOR-constraint set

Let p_r be a predicate and $AST(P_r)$ its abstract syntax tree, S_N the constraint set attached to a node N (where $S^t{}_N$ and $S^f{}_N$ are the true and false constraints associated with the node). The following alg. generates the BOR-constraint set for p_r

Input: $AST(p_r)$ (only singular expressions)

Output: BOR-Constraint set attached to the root node

1. Label each leaf node N of $AST(p_r)$ with its constraint set $S_N = \{t, f\}$
2. Visit the AST bottom-up. Let N_1 and N_2 direct descendants of node N and S_{N1} and S_{N2} the corresponding BOR-constraint set. S_N is computed as follows:

   2.1 N is an OR-node:
   - $S^f{}_N = S^f{}_{N1} \otimes S^f{}_{N2}$
   - $S^t{}_N = (S^t{}_{N1} \times \{f_2\}) \cup (\{f_1\} \times S^t{}_{N2})$ where $f_1 \in S^f{}_{N1}$ and $f_2 \in S^f{}_{N2}$

   2.2 N is an AND-node:
   - $S^t{}_N = S^t{}_{N1} \otimes S^t{}_{N2}$
   - $S^f{}_N = (S^f{}_{N1} \times \{t_2\}) \cup (\{t_1\} \times S^f{}_{N2})$ where $t_1 \in S^t{}_{N1}$ and $t_2 \in S^t{}_{N2}$

   2.3 N is NOT-node:
   - $S^t{}_N = S^f{}_{N1}$
   - $S^f{}_N = S^t{}_{N1}$

BOR-constraint set example

Let's apply the BOR-constraint procedure to:

- $(a + b < c) \wedge \neg p \vee (r > s)$

Generating the BRO-constraint set

Input: $AST(p_r)$ (only singular expressions)
Output: BRO-Constraint set attached to the root node

1. Label each leaf node N of $AST(p_r)$ with its constraint set S_N. For each leaf node that represents a Boolean variable $S_N = \{t, f\}$. For each leaf node that is a relational expression $S_N = \{(>), (=), (<)\}$.
2. Visit the AST bottom-up. Let N_1 and N_2 direct descendants of node N and S_{N1} and S_{N2} the corresponding BRO-constraint set. S_N is computed as done for the BOR procedure.

Let's apply the BRO-constraint procedure to:

- $(a + b < c) \wedge \neg p \vee (r > s)$

Generating the BRO-constraint set

Input: $AST(p_r)$ (only singular expressions)
Output: BRO-Constraint set attached to the root node

1. Label each leaf node N of $AST(p_r)$ with its constraint set S_N. For each leaf node that represents a Boolean variable $S_N = \{t, f\}$. For each leaf node that is a relational expression $S_N = \{(>), (=), (<)\}$.
2. Visit the AST bottom-up. Let N_1 and N_2 direct descendants of node N and S_{N1} and S_{N2} the corresponding BRO-constraint set. S_N is computed as done for the BOR procedure.

Let's apply the BRO-constraint procedure to:

- $(a + b < c) \wedge \neg p \vee (r > s)$

Generating the BRE-constraint set

Input: $AST(p_r)$ (only singular expressions)
Output: BRE-Constraint set attached to the root node

1. Label each leaf node N of $AST(p_r)$ with its constraint set S_N. For each leaf node that represents a Boolean variable $S_N = \{t, f\}$. For each leaf node that is a relational expression $S_N = \{(+\epsilon), (=), (-\epsilon)\}$.
2. Visit the AST bottom-up. Let N_1 and N_2 direct descendants of node N and S_{N1} and S_{N2} the corresponding BRE-constraint set. S_N is computed as done for the BOR procedure.

Let's apply the BRO-constraint procedure to:

- $(a + b < c) \wedge \neg p \vee (r > s)$

Generating the BRE-constraint set

Input: $AST(p_r)$ (only singular expressions)
Output: BRE-Constraint set attached to the root node

1. Label each leaf node N of $AST(p_r)$ with its constraint set S_N. For each leaf node that represents a Boolean variable $S_N = \{t, f\}$. For each leaf node that is a relational expression $S_N = \{(+\epsilon), (=), (-\epsilon)\}$.
2. Visit the AST bottom-up. Let N_1 and N_2 direct descendants of node N and S_{N1} and S_{N2} the corresponding BRE-constraint set. S_N is computed as done for the BOR procedure.

Let's apply the BRO-constraint procedure to:

- $(a + b < c) \wedge \neg p \vee (r > s)$

Usage of predicate testing techniques

- Specification based testing
- program based testing