

CS 3250: Software Testing (Fall 2025)

POTD 7: Logic coverage for source code (highEnough)

Due 1-November-2025, 11:00am EST

Purpose: Practice applying logic-based testing for program source code; get ready to work on homework assignment, and prepare for quiz4 and the final exam

You may make a copy of a worksheet and complete this activity, or type your answers in any text editor. You may work alone or with at most two other students in this course.

Consider the following Java method, highEnough(), and tests t1 - t7

```
Line 1: public String highEnough(boolean e, int lft, int rht) {
Line 2:     int result = 8;
Line 3:     int n = rht % lft;
Line 4:     if (n == 0 || e)
Line 5:         result = rht;
Line 6:     else
Line 7:     {
Line 8:         for (int i=2; i <= n; i++)
Line 9:             result = result * 4;
Line 10:    }
Line 11:    if (result >= 17)
Line 12:        return ("Yes");
Line 13:    else
Line 14:        return ("No");
Line 15: }
```

Identify a **minimal** set of tests for highEnough() that achieves **Correlated Active Clause Coverage (CACC)**. For credit, you **must** use tests from the list of seven tests (t1, t2, ..., t7) given in the following table. Show your analysis and thought process.

[optional] You may start by filling out the following table. The values for test t1 are already filled in for you. (F = False, T = True, n/a = not applicable).

test#	e	lft	rht	highEnough(e,lft,rht)	n == 0	e	i <= n	result >= 17
t1	F	4	6	Yes	F	F	T	T
t2	F	3	9					
t3	T	4	7					
t4	F	8	9					
t5	F	4	8					
t6	T	4	4					
t7	T	5	0					

Sample solution

test#	e	lft	rht	highEnough(e, lft, rht)	n==0	e	i<=n	result>=17
t1	F	4	6	Yes	F	F	T	T
t2	F	3	9	No	T	F	n/a	F
t3	T	4	7	No	F	T	n/a	F
t4	F	8	9	No	F	F	F	F
t5	F	4	8	No	T	F	n/a	F
t6	T	4	4	No	T	T	n/a	F
t7	T	5	0	No	T	T	n/a	F

Note on the table:

- Since i is initialized in a for loop, i does not exist when the if-statement (line 5) is true. Thus, the truth value of $i \leq n$ is not applicable.
- Notice that the n/a does not impact how the tests are derived (see next page for an analysis)

The highEnough() method has 3 predicates:

- (line 5) $P1 = (n == 0 \parallel e)$
- (line 9) $P2 = (i \leq n)$
- (line 14) $P3 = (result \geq 17)$

Let $P1 = (a \vee e)$, where a is $n == 0$.

Let $P2 = b$, where b is $i \leq n$.

Let $P3 = c$, where c is $result \geq 17$.

Since the given source code consists of 3 predicates, CACC tests must be created from all three predicates.

To determine tests for each predicate:

1. plug in the truth table
2. determine pairs of rows that satisfy CACC for the predicate
3. map pairs of rows to the given set of tests (t1,...,t7).

For **$P1 = (a \vee e)$** . Determine tests for P1

row	a	e	$P1 = (a \vee e)$	Pa	Pe
1	T	T	T		
2	T		T	T	
3		T	T		T
4				T	T

From the above truth table, possible pairs of rows that satisfy CACC for P1:

- clause a: (row2, row4) $\rightarrow$ (TF, FF)
- clause e: (row3, row4) $\rightarrow$ (FT, FF)

Consider the possible pairs of rows (written as the truth values of clauses aebc, and x could be T or F) vs. the given tests (t1, ..., t7).

TFxx: possible tests are t2, t5

FTxx: possible tests are t3

FFxx: possible tests are t1, t4

For **P2 = b**. Determine tests for P2

row	b	P2 = (b)	Pb
1	T	T	T
2			T

Possible pairs of rows that satisfy CACC for P2: (row1, row2) $\rightarrow$ (T, F).

To reach P2, P1 must be reached and P1 must be F.

For P1 to be F, both clauses a and e must be F.

Putting the truth values of clauses a, e, and b together: tests that satisfy CACC for P2 are FFT and FFF.

Consider the possible pairs of rows vs. the given tests (t1, ..., t7).

FFT_x: possible tests are t1

FFF_x: possible tests are t4

For **P3 = c**. Determine tests for P3 (similar to how we determine tests for P2). Possible pairs of rows that satisfy CACC for P3: (row1, row2) $\rightarrow$ (T, F).

P3 is always reached. Written as truth values of clauses aebc: Tests that satisfy CACC for P3 are xxxT (mapped to t1) and xxxF (mapped to tests t2, ..., t7).

Combine tests for all predicates:

P1 requires t3, one of (t2, t5), one of (t1, t4).

P2 requires t1 and t4.

P3 requires t1 and one of (t2, ..., t7).

Thus, need t1, t3, t4, one of (t2, t5).

Possible answers: {t1, **t2**, t3, t4} or {t1, t3, t4, **t5**}

Don't pick t2 and t5 together (not minimal).

Don't pick t6 or t7 (not minimal since either t2 or t5 is still needed for P1).

Grading rubric

[Total: 10 points]: Done (or provide evidence of your attempt, full or reasonable effort)

- (5 points) — Providing evidence of your attempt, minimal effort
(-2.5 points) for 24 hours late (submitted after 1-Nov-2025 11am EST, by 2-Nov-2025 11am EST)
(-5 points) for 48 hours late (submitted after 2-Nov-2025 11am EST, by 3-Nov-2025 11am EST)
-

Submission

- Take a selfie (or picture) of your team — get **creative** and have **fun!** — and submit it with your POTD.
- You may do one of the following:
 - take screenshot(s) of your POTD — if you write your answer(s) or draw a graph on papers, or
 - save your POTD as a .pdf file — No Word document.
- Log in to Canvas, then upload your report (.pdf) to **POTD 7 on Gradescope**.
- Make sure you connect your partner to your group on Gradescope so that everyone receives credit
- Each team submits only **one** copy

Making your submission available to instructor and course staff is **your** responsibility; if we cannot access or open your file, you will not get credit. Be sure to test access to your file before the due date.