Logic Coverage
Active Clause Coverage

CS 3250
Software Testing

[Ammann and Offutt, "Introduction to Software Testing," Ch. 8]

Active Clause Coverage (ACC)

- From the testing perspective, we would test each clause under circumstances where the clause determines the predicate

For each p in P and each major clause c_i in Cp, choose minor clauses c_j, j != i, so that c_i determines p. TR has two requirements for each c_i: c_i evaluates to true and c_i evaluates to false.

- Steps:
  - Analyze determination, i.e., making the clause active
  - Derive test requirements that evaluates the major clause to true and false

- This is a form of "Modified Condition Decision Coverage" (MCDC), which is required by the US Federal Aviation Administration (FAA) for safety critical avionics software

ACC Example

$p = a \vee b$

$p_a = p_{a=true} \oplus p_{a=false}$
$= (true \vee b) \oplus (false \vee b)$
$= true \oplus b$
$= \neg b$

For clause a, a determines p if and only if b is false
Thus, we have two test requirements {(a=true, b = false), (a=false, b= false)}

$p_b = p_{b=true} \oplus p_{b=false}$
$= (a \vee true) \oplus (a \vee false)$
$= true \oplus a$
$= \neg a$

For clause b, b determines p if and only if a is false
Thus, we have two test requirements {(a=false, b = true), (a=false, b= false)}

TR = {(a=true, b = false), (a=false, b= false), (a=false, b = true), ~~(a=false, b= false)~~}

- Overlap is common
- n ≤ number test requirements ≤ 2n, where n = number clauses

Ambiguity in ACC

$$\mathbf{p = a \wedge (b \vee c)}$$

$$\begin{aligned} p_a &= p_{a=true} \oplus p_{a=false} \\ &= (true \wedge (b \vee c)) \oplus (false \wedge (b \vee c)) \\ &= (b \vee c) \oplus false \\ &= b \vee c \end{aligned}$$

a = T, a = F, { TT, TF, FT } -- is this allowed?

- Do the minor clauses have to have the same values when the major clause is true and false?

This leads to 3 separate criteria:

1. Minor clauses do not need to be the same (GACC)
2. Minor clauses must be the same and force the predicate to become both true and false (RACC)
3. Minor clauses force the predicate to become both true and false (CACC)

General Active Clause Coverage (GACC)

- *For each major clause c, choose minor clauses such that c determines the predicate*
- *Clause c has to evaluate to true and false*
- *Minor clauses do not need to be the same*

- Allow minor clauses to have different values
- Possible to satisfy GACC without satisfying predicate coverage
- We really want to cause predicates to be both true and false

GACC Example

$p = a \wedge (b \vee c)$

Major clause: a

row	a	b	c	p	p_a	p_b	p_c
1	T	T	T	T	T		
2	T	T		T	T	T	
3	T		T	T	T		T
4	T					T	T
5		T	T		T		
6		T			T		
7			T		T		
8							

p_a = T
a = T

Rows:
1, 2, 3

p_a = T
a = F

Rows:
5, 6, 7

Set of possible test requirements:

{(1,5), (1,6), (1,7), (2,5), (2,6), (2,7), (3,5), (3,6), (3,7)}

GACC Example

p = a ∧ (b ∨ c) **Major clause: b**

row	a	b	c	p	p_a	p_b	p_c
1	T	T	T	T	T		
2	T	T		T	T	T	
3	T		T	T	T		T
4	T					T	T
5		T	T		T		
6		T			T		
7			T		T		
8							

p_b = T
b = T

Rows:
2

p_b = T
b = F

Rows:
4

Set of possible test requirements:

{(2,4)}

GACC Example

p = a ∧ (b ∨ c) **Major clause: c**

row	a	b	c	p	p_a	p_b	p_c
1	T	T	T	T	T		
2	T	T		T	T	T	
3	T		T	T	T		T
4	T					T	T
5		T	T		T		
6		T			T		
7			T		T		
8							

p_c = T
c = T

Rows:
3

p_c = T
c = F

Rows:
4

Set of possible test requirements:

{(3,4)}

GACC Example

$p = a \wedge (b \vee c)$

Set of possible test requirements:

Major clause a: {(1,5), (1,6), (1,7), (2,5), (2,6), (2,7), (3,5), (3,6), (3,7)}

Major clause b: {(2,4)}

Major clause c: {(3,4)}

Give GACC-adequate test set (= minimal test set that satisfies GACC)

Based on the possible test requirements, select a smallest combination pairs (one for each clause)

(2, 5), (2, 4), (3, 4)

This means, we need at least 4 tests (i.e., test inputs) -- rows 2, 3, 4, 5

GACC-adequate test set = { 2, 3, 4, 5 }

GACC Does Not Subsume PC

$p = (\neg a \wedge \neg b) \vee (a \wedge \neg c) \vee (\neg a \wedge c)$ **Major clause: a**

row	a	b	c	p	p_a	p_b	p_c
1	T	T	T		T		T
2	T	T		T	T		T
3	T		T		T		T
4	T			T			T
5		T	T	T	T		T
6		T			T	T	T
7			T	T	T		
8				T		T	

p_a = T
a = T

Rows:
1, 2, 3

---

p_a = T
a = F

Rows:
5, 6, 7

Set of possible test requirements:

{(1,5), (1,6), (1,7), (2,5), (2,6), (2,7), (3,5), (3,6), (3,7)}

Restricted Active Clause Coverage (RACC)

- *For each major clause c, choose minor clauses such that c determines the predicate*
- *Clause c has to evaluate to true and false*
- *Predicate p has to evaluate to true and false*
- *Minor clauses must be the same*

- This has been a common interpretation by aviation developers
- RACC often leads to infeasible test requirements
- Stricter version of CACC – requires the same minor clause
- There is no logical reason for such a restriction

RACC Example

$p = (\neg a \wedge \neg b) \vee (a \wedge \neg c) \vee (\neg a \wedge c)$ **Major clause: a**

row	a	b	c	p	p_a	p_b	p_c
1	T	T	T		T		T
2	T	T		T	T		T
3	T		T		T		T
4	T			T			T
5		T	T	T	T		T
6		T			T	T	T
7			T	T	T		
8				T		T	

p_a = T
a = T

Rows:
1, 2, 3

---

p_a = T
a = F

Rows:
5, 6, 7

P = T, F

Same minor clauses

Set of possible test requirements:
{(1,5),(2,6),(3,7)}

Then, derive tests for clauses b and c

Correlated Active Clause Coverage (CACC)

- *For each major clause c, choose minor clauses such that c determines the predicate*
- *Clause c has to evaluate to true and false*
- *Predicate p has to evaluate to true and false*
- *Minor clauses do not need to be the same*

- A more recent interpretation
- Implicitly allows minor clauses to have different values
- Explicitly satisfies (subsumes) predicate coverage
- Stricter version of GACC – adds PC requirement to GACC

CACC Example

$p = (\neg a \wedge \neg b) \vee (a \wedge \neg c) \vee (\neg a \wedge c)$ **Major clause: a**

row	a	b	c	p	p_a	p_b	p_c
1	T	T	T		T		T
2	T	T		T	T		T
3	T		T		T		T
4	T			T			T
5		T	T	T	T		T
6		T			T	T	T
7			T	T	T		
8				T		T	

p_a = T
a = T

Rows:
1, 2, 3

---

p_a = T
a = F

Rows:
5, 6, 7

P = T, F

Set of possible test requirements:
{(1,5), (1,7), (2,6), (3,5), (3,7)}

Then, derive tests for clauses b and c

CACC vs. RACC

- What is the difference between CACC and RACC?
  - RACC imposes more constraints on the truth values of minor clauses
    - RACC requires the the truth values for minor clauses are consistent between the two requirements
  - In practice?
    - There are fewer possible truth table row combinations that satisfy RACC
    - RACC leads to more infeasible test requirements than CACC

RACC Leads to Infeasible Requirements

- If clauses are independent, there is no problem
- If clauses are dependent, some combination of clauses become infeasible

Suppose a program has

p = `valve is closed` AND (`system is in operational mode` OR `system is in standby mode`)

= $a \wedge (b \vee c)$

Three clauses:
- a = the valve is closed
- b = the system mode is operational
- c = the system mode is standby

Assume the following constraints:

- A valve must be open in "operational" mode and closed in all other modes
- Mode cannot be in both "operational" and "standby" at the same time

Constraints: $\neg a \leftrightarrow b$

$\neg(b \wedge c)$

CACC and RACC – Infeasible TRs

p = a ∧ (b ∨ c)

Constraints: ¬a ↔ b

¬(b ∧ c)

row	a	b	c	p	Constraint violations
1	T	T	T	T	1,2
2	T	T	F	T	1
3	T	F	T	T	
4	T	F	F	F	
5	F	T	T	F	2
6	F	T	F	F	
7	F	F	T	F	1
8	F	F	F	F	1

CACC: out of {1,2,3} x {5,6,7}, only (3,6) is feasible

RACC: out of (1,5), (2,6), (3,7), none are feasible

Summary

- A clause is active if the other clauses have values that allow the active clause to determine the value of the predicate
- Active Clause Coverage (ACC) requires each clause to be made active, and then true and then false
- ACC comes with three different possible interpretations
  - General Active Clause Coverage (GACC)
  - Correlated Active Clause Coverage (CACC)
  - Restricted Active Clause Coverage (RACC)