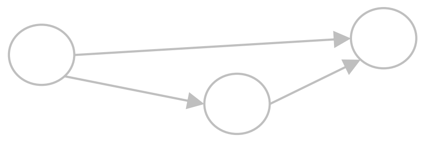
SYSC 4101 / 5105

Logic Criteria

Test Criteria Based on Structure [Offutt]

- Graphs



- Logical Expressions

(not X or not Y) and A and B

Can appear in:
- State machine
- Source code
- Software specification

- Input Domain C

A: {0,1,>1}

0,700,800}

e,cs,isa,infs}

- Syntactic Structures

```
if (x>y)
     z = x - y;
else
     z = 2 * x
```

Introduction

- A predicate is an expression that evaluates to a Boolean value.

  `((a>b) or C) and p(x))`

  A predicate contains:

  - Boolean variable: `C`
  - Non-Boolean variable(s) being compared: `a`, `b`
  - Function calls returning a Boolean value: `p(x)`

  Can be written as `X or Y and Z`, with three Boolean variables.

- A clause is a predicate with no logical operators

  `(a>b)`, `C`, and `p(x)` are clauses

- Logical operators

  not (! or $\overline{\ \ }$), and (. or ∧), or (+ or ∨), implies (→), x-or (⊕), equivalence (↔), …

  (order of precedence)

Introduction (cont.)

- *Boolean space:*
  - The n-dimensional space formed by the input variables
  - We talk about a "point" in the Boolean space.
- *Product term*:
  - String of clause related by the `and` operator
- *Sum-of-products form / Disjunctive Normal Form (DNF)*:
  - Product terms related by the `or` operator
  - Sum-of-Products Boolean expressions are easier to read and interpret, and can be used for testing purposes
- *Logic minimization*:
  - Deriving compact (irredundant) but equivalent Boolean expressions, using Boolean algebra
- Laws of Boolean algebra (e.g., De Morgan's laws)

Example (Boiler)

- Example:
  - Enable or disable the ignition of a boiler based on four input variables
    - NormalPressure (A): pressure within safe operating limit?
    - CallForHeat (B): ambient temperature below set point?
    - DamperShut (C): exhaust duct is closed?
    - ManualMode (D): manual operation selected?
  - Logic Function: $Z = A(B\bar{C} + D)$
    - If A false, i.e., the pressure is not safe, forget the other factors
    - If manual selection then the user/operator is responsible, B and C truth values do not matter

    (Sum-of-Product Form: $Z = A(B\bar{C} + D) = AB\bar{C} + AD$)

Boiler Truth Table

Input Vector Number	Normal Pressure	CallForHeat	DamperShut	ManualMode	IgnitionTable
	A	B	C	D	Z
0-7	0	-	-	-	0
8	1	0	0	0	0
9	1	0	0	1	1
10	1	0	1	0	0
11	1	0	1	1	1
12	1	1	0	0	1
13	1	1	0	1	1
14	1	1	1	0	0
15	1	1	1	1	1

Caption:

- 0 means either No, Off (CallForHeat), or On (IgnitionDisabled)
- Shortcut: - means the value does not matter

➢ We do not want to do exhaustive testing (16 test cases)

Simple Criteria

All-Predicates

For each predicate p, TR (set of Test Requirements) contains two requirements: p evaluates to true, and p evaluates to false.

All-Clauses

For each clause c (in each predicate), TR contains two requirements: c evaluates to true, and c evaluates to false.

All-Combinations

For each predicate p, TR has test requirements for the clauses in p to evaluate to each possible combination of truth values.

(All the elements of the truth table.)

Example (Boiler)

$Z = A(B\bar{C} + D)$

Notice that all the clauses do not have all possible values: All-Predicate does not subsume All-Clauses

All-Predicate:

t_1 : A=True, B=False, C=True, D=true

as a result, Z=True

t_2 : A=False, B=False, C=True, D=true

as a result, Z=False

Notice that the predicate does not have all possible values: All-Clauses does not subsume All-Predicates

All-Clauses:

t_3 : A=True, B=False, C=True, D=False

as a result, Z=False

t_4 : A=False, B=True, C=False, D=True

as a result, Z=False

All-Combinations:

16 entries in the truth table of Z.

Other Criteria - Definitions

- Definition: major clause
  - A major clause is a clause we are focusing on.
  - The other clauses of the predicate are called minor clauses.
- Definition: determination
  - Given a major clause c_i in predicate p, we say that c_i determines p if the minor clauses c_j of p (j≠i) have values so that changing the truth value of c_i changes the truth value of p.
- Example: $Z = A(B\bar{C} + D)$
  - Major clause: A
  - Minor clauses: B, C, D
  - Combination of values: A=?, B=True, C=False, D=True
    - Changing the value of A changes the value of Z.
  - So A determines Z.

Active Clause Coverage

- Making sure that major clauses do affect their predicates

- Active Clause Coverage: (a.k.a. MCDC in DO-178B/C)
  - For each predicate p and each major clause c_i of p, choose minor clauses c_j ($j \neq i$) 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.
- Ambiguity in definition: should the minor clauses have the same values for the two requirements?
  - Three different flavors discussed in textbook
    - General Active Clause Coverage (GACC)
    - Correlated Active Clause Coverage (CACC)
    - Restricted Active Clause Coverage (RACC)

Other Criteria (cont.)

- Restricted Active Clause Coverage (RACC):
  - For each predicate p and each major clause c_i of p, choose minor clauses c_j ($j \neq i$) 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.
  - The values chosen for the minor clauses c_j must be the same when c_i is true and when c_i is false.

RACC (Boiler)

$Z = A(B\bar{C} + D) = AB\bar{C} + AD$

- Major clause A: $(B\bar{C} + D)$ must be true so that A determines P
  - **A=true**, B=true, C=false, D=true ⇒ P=true
  - **A=false**, B=true, C=false, D=true ⇒ P=false
  - ➢ (true, true, false, true), (false, true, false, true)
- Major clause B: AD must be false, and $A\bar{C}$ must be true
  - A=true, **B=true**, C=false, D=false ⇒ P=true
  - A=true, **B=false**, C=false, D=false ⇒ P=false
  - ➢ (true, true, false, false), (true, false, false, false)
- Major clause C: AD must be false, and AB must be true
  - A=true, B=true, **C=true**, D=false ⇒ P=false
  - A=true, B=true, **C=false**, D=false ⇒ P=true
  - ➢ (true, true, true, false), (true, true, false, false)
- Major clause D: $AB\bar{C}$ must be false, and A must be true
  - A=true, B=false, C=true, **D=true** ⇒ P=true
  - A=true, B=false, C=true, **D=false** ⇒ P=false
  - ➢ (true, false, true, true), (true, false, true, false)

RACC adequate test suite:

(true, true, false, true) (false, true, false, true) (true, true, false, false) (true, false, false, false)
(true, true, true, false) (true, false, true, true) (true, false, true, false)

RACC—Algorithm

1. Build the truth table
2. For each row in the truth table
   a. Change each clause, one at a time
   b. If the predicate value changes, record which correlated row is used

Example: $A \wedge (B \vee C)$

	A B C	Result	Corr. False Case
1	T T T	T	A (5)
2	T T F	T	A (6) , B (4)
3	T F T	T	A (7) , C (4)
4	T F F	F	B (2) , C (3)
5	F T T	F	A (1)
6	F T F	F	A (2)
7	F F T	F	A (3)
8	F F F	F	-

3. Take a pair for each clause:
   - A : (1,5), or (2,6), or (3,7)
   - B : (2,4)
   - C : (3,4)

Two minimal sets to cover the criterion:
- (2,3,4,6) or (2,3,4,7)

That is 4 test cases (instead of 8 for all possible combinations)

Other Criteria—Based on DNF

- *Implicant*:
  - Each term of a sum-of-products expression (DNF) is an implicant
  - Each implicant is a sufficient condition to fulfill for *True* output of the predicate
- *Cube*:
  - Compressed partial truth table, input vector with wild cards (x)
- *Prime implicants*:
  - An implicant which is not a subset (set of values of variables) of any other implicant

- DNF must be minimal
  - Every implicant is prime
  - No implicant is redundant (cannot be omitted)

Example (Boiler)

- Logical expression/formula: $Z = A(B\bar{C} + D) = AB\bar{C} + AD$
- Implicants: $AB\bar{C}$, AD
- Cube:
  - Five cubes of the boiler control table are 110x|1, 1xx1|1, 0xxx|0, 1x10|0, 10x0|0
- Prime implicant:
  - $AB\bar{C}$ = 110x = {1101, 1100},
  - AD =1xx1={1001, 1011, 1101, 1111}

  => Term 1 is not a subset of term2 (and vice versa)

  => Both terms are prime implicants

- Logical expression: $AB\bar{C} + AD + ABD$
  - $AB\bar{C}$ = 110x = {1101, 1100},
  - AD = 1xx1={1001, 1011, 1101, 1111}
  - ABD = 11x1 = {1101, 1111}
  - ABD is an implicant, but not a prime implicant because {1101, 1111} $\subset$ {1001, 1011, 1101, 1111}

Other Criteria—Based on DNF (cont.)

- Corresponding Unique True Point and Near False Point Pair Coverage (CUTPNFP)
  - Given a minimal DNF representation of a predicate f, for each clause c in each implicant i, TR contains a unique true point for i and a near false point for c in i such that the two points differ only in the truth value of c.
  - A unique true point for an implicant is an assignment of truth values such that this implicant is true and all the other implicants in the predicate are false.
  - A near false point for predicate f with respect to clause c in implicant i is an assignment of truth values such that f is false, but if c is negated and all other clauses are left as is, i (and hence f) evaluates to true.

Warning: It is not sufficient to simply negate the clause in the UTPs to obtain the NFPs.

Other Criteria—Based on DNF (cont.)

- Why Unique True Points (UTP) and Near False Points (NFP)?
  - If there is a fault in the evaluation of a term (implicant) which would lead the predicate to evaluate to false when it should evaluate to true, a UTP will reveal the fault.
  - If there is a fault in the evaluation of a clause in a term (implicant) which would lead to that clause to be true (resp. false) when it should be false (resp. true) and this leads to the predicate to be true instead of false, a NFP will reveal the fault.

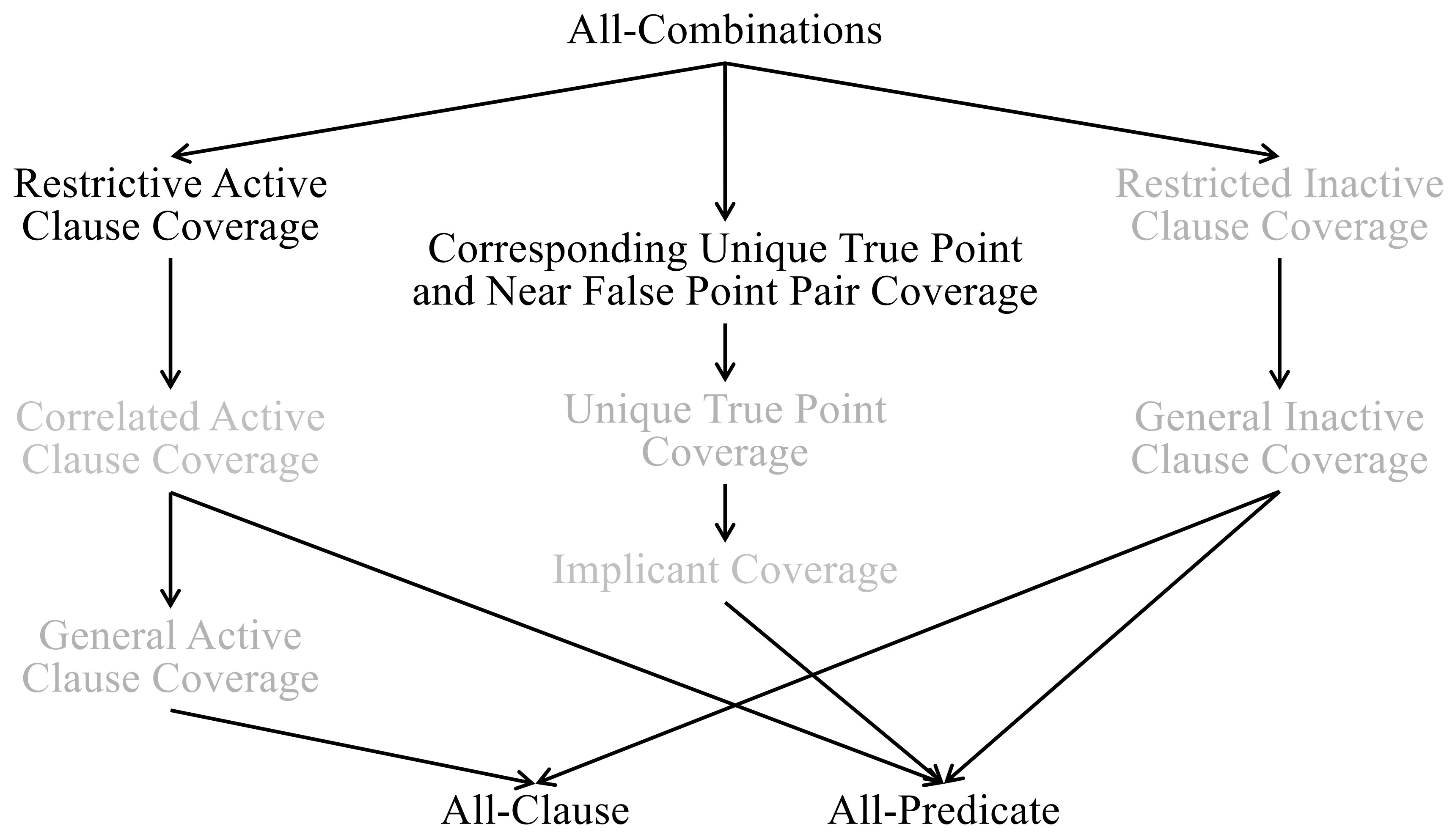
Algorithm

- Alternative (short and incomplete) definition of Near False Point
  - variants for each product term where one literal is negated such that the <u>overall</u> logic function evaluates to false
- Example: f(a,b,c) = $term_1$(a,b,c) + $term_2$(a,b,c)

  where $term_1$(a,b,c) = ab and $term_2$(a,b,c) = $b\bar{c}$

  f(a,b,c) = $ab + b\bar{c}$
- NFP for a in ab:
  - Cube representation of term1: (1,1,-)
  - Negating a: (0,1,-)
  - Overall logic function evaluates to false:
    - $b\bar{c}$ must be false
    - c=true (since b=true)
    - ➢ NFP=(0,1,1)

Subsumption


All-Combinations
Restrictive Active Clause Coverage
Corresponding Unique True Point and Near False Point Pair Coverage
Restricted Inactive Clause Coverage
Correlated Active Clause Coverage
Unique True Point Coverage
General Inactive Clause Coverage
Implicant Coverage
General Active Clause Coverage
All-Clause
All-Predicate

Applications of Logic Coverage Criteria

- Source code
  - Combined with graph coverage criteria
- Specification: e.g., pre (and post) conditions
  - Combined with coverage criteria
- UML diagrams
  - State machines (guards)
    - Combined with graph coverage criteria
  - Sequence diagrams (guards)
    - Combined with graph coverage criteria
- No major difficulty, except one of reachability.

Examples

Example

P = A (B + C) = AB + AC

Restricted Active Clause Coverage

- Major clause A: (B + C) must be true so that A determines P
  - **A=true**, B=true, C=false ⇒ P=true
  - **A=false**, B=true, C=false ⇒ P=false
  - ➢ (true, true, false), (false, true, false)
- Major clause B: AC must be false, and A must be true
  - A=true, **B=true**, C=false ⇒ P=true
  - A=true, **B=false**, C=false ⇒ P=false
  - ➢ (true, true, false), (true, false, false)
- Major clause C: AB must be false, and A must be true
  - A=true, B=false, **C=true** ⇒ P=true
  - A=true, B=false, **C=false** ⇒ P=false
  - ➢ (true, false, true), (true, false, false)
- RACC adequate test suite: (true, true, false), (false, true, false), (true, false, false), (true, false, true)

Example

- f = ab + cd
- Corresponding Unique True Point and Near False Point Pair Coverage (CUTPNFP)
  - UTP for ab: (1,1,?,?) → (1,1,0,0) or (1,1,1,0) or (1,1,0,1)
  - NFP for a in ab: (0,1,?,?) → (0,1,0,0) or (0,1,1,0) or (0,1,0,1)
  - NFP for b in ab: (1,0,?,?) → (1,0,0,0) or (1,0,1,0) or (1,0,0,1)
  - UTP for cd: (?,?,1,1) → (0,0,1,1) or (0,1,1,1) or (1,0,1,1)
  - NFP for c in cd: (?,?,0,1) → (0,0,0,1) or (0,1,0,1) or (1,0,0,1)
  - NFP for d in cd: (?,?,1,0) → (0,0,1,0) or (0,1,1,0) or (1,0,1,0)
  - Test Set: {1,1,0,0), (0,1,0,0), (1,0,0,0), (0,0,1,1), (0,0,0,1), (0,0,1,0)}

Example

P = A ↔ B

Restricted Active Clause Coverage

- Major clause A
  - A=true, any value of B leads to A determining P, we choose B=true
    - The predicate evaluates to true
  - A=false, any value of B leads to A determining P, we choose B=true
    - The predicate evaluates to false
  - ➢ (true, true), (false, true)
- Major clause B
  - B=true, any value of A leads to B determining P, we choose A=true
    - The predicate evaluates to true
  - B=false, any value of A leads to B determining P, we choose A=true
    - The predicate evaluates to false
  - ➢ (true, true), (true, false)
- RACC adequate test suite: (true, true), (false, true), (true, false)
  - This test suite is adequate for Predicate Coverage
  - Same as CACC (coincidence)