CSCE 747 Software Testing and Quality Assurance

Lecture 08 – Structural Testing Wrap-up

Last Time

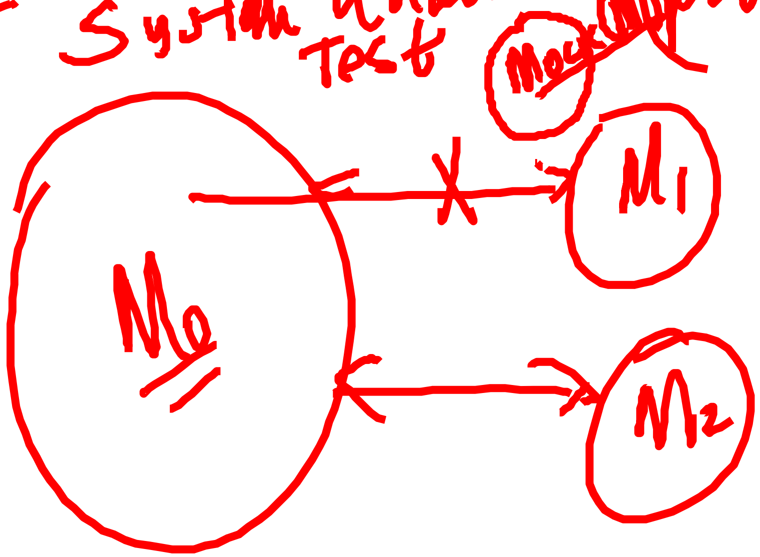
- Dataflow Testing
- Ch 10, pp 151-167
- Reading assignment *Google(Mockito Tutorial Vogel)*
- .

SUT System under Test

Today

- Structural Testing WrapUp
- Ch 11, pp 169-177

(path coverage tools)

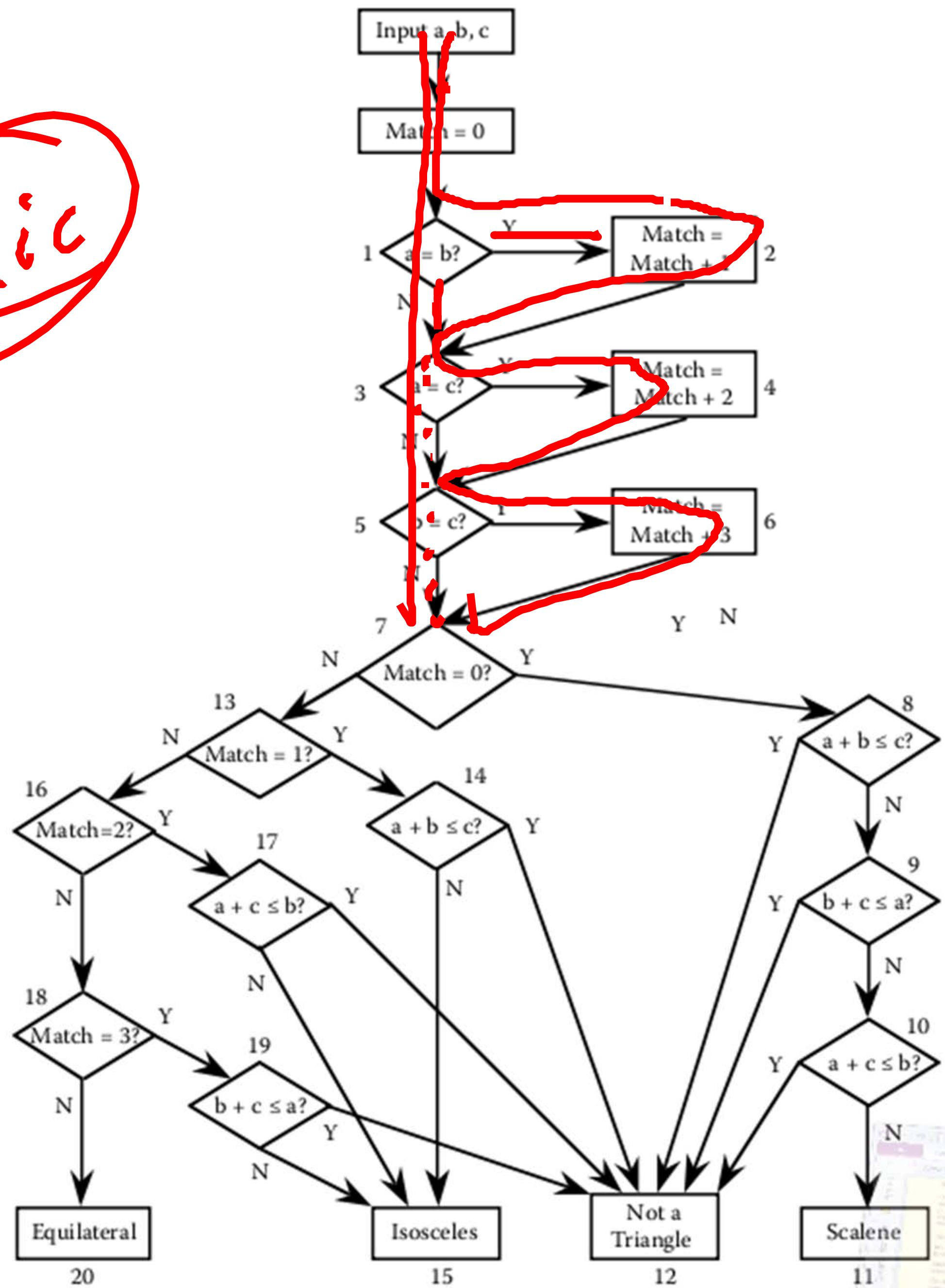


When should testing stop?

- Here are some possible answers:
- 1. When you run out of time X
- 2. When continued testing causes no new failures
- 3. When continued testing reveals no new faults
- 4. When you cannot think of any new test cases
- 5. When you reach a point of diminishing returns
- 6. When mandated coverage has been attained
- 7. When all faults have been removed X Not possible

- **Triangle Program**

acyclic



Gaps and Redundancies

- Paths in Triangle

Table 11.1 Paths in the Triangle Program

Path	*Node Sequence*	*Description*
p1	1–2–3–4–5–6–7–13–16–18–20	Equilateral
p2	1–3–5–6–7–13–16–18–19–15	Isosceles (b = c)
p3	1–3–5–6–7–13–16–18–19–12	Not a Triangle (b = c)
p4	1–3–4–5–7–13–16–17–15	Isosceles (a = c)
p5	1–3–4–5–7–13–16–17–12	Not a Triangle (a = c)
p6	1–2–3–5–7–13–14–15	Isosceles (a = b)
p7	1–2–3–5–7–13–14–12	Not a Triangle (a = b)
p8	1–3–5–7–8–12	Not a Triangle (a + b ≤ c)
p9	1–3–5–7–8–9–12	Not a Triangle (b + c ≤ a)
p10	1–3–5–7–8–9–10–12	Not a Triangle (a + c ≤ b)
p11	1–3–5–7–8–9–10–11	Scalene

Impossible
a=b, b=c
a!=c

nom min min+ max- max+

Table 11.2 Path Coverage of Nominal Values

Case	*a*	*b*	*c*	*Expected Output*	*Path*
1	100	100	1	Isosceles	p6
2	100	100	2	Isosceles	p6
3	100	100	100	Equilateral	p1
4	100	100	199	Isosceles	p6
5	100	100	200	Not a Triangle	p7
6	100	1	100	Isosceles	p4
7	100	2	100	Isosceles	p4
8	100	100	100	Equilateral	p1
9	100	199	100	Isosceles	p4
10	100	200	100	Not a Triangle	p5
11	1	100	100	Isosceles	p2
12	2	100	100	Isosceles	p2
13	100	100	100	Equilateral	p1
14	199	100	100	Isosceles	p2
15	200	100	100	Not a Triangle	p3

Table 11.3 Path Coverage of Worst-Case Values

	p1	*p2*	*p3*	*p4*	*p5*	*p6*	*p7*	*p8*	*p9*	*p10*	*p11*
Nominal	3	3	1	3	1	3	1	0	0	0	0
Worst case	5	12	6	11	6	12	7	17	18	19	12

Metrics for Method Evaluation

- Definition: The coverage of a methodology M with respect to a metric S is the ratio of n to s. We denote it as C(M, S).

  n cases

- Definition: The redundancy of a methodology M with respect to a metric S is the ratio of m to s. We denote it as R(M, S).

  redundant

- Definition: The net redundancy of a methodology M with respect to a metric S is the ratio of m to n. We denote it as NR(M, S).

  test cases

Table 11.4 Metrics for the Triangle Program

Method	*m*	*n*	*s*	$C(M, S) = n/s$	$R(M, S) = m/s$	$NR(M, S) = m/n$
Nominal	15	7	11	0.64	1.36	2.14
Worst-case	125	11	11	1.00	11.36	11.36
Goal	s	s	s	1.00	1.00	1.00

Table 11.5 Metrics for the Commission Problem

Method	*m*	*n*	*s*	*C(M, S) = n/s*	*R(M, S) = m/s*
Output bva	25	11	11	1	2.27
Decision table	3	11	11	1	0.27
DD-Path	25	11	11	1	2.27
du-path	25	33	33	1	0.76
Slice	25	40	40	1	0.63

data flow

Pseudo-code for the Insurance Premium Program

```
1. Input(baseRate, driverAge, points)
2. premium = 0
3. Select Case driverAge
4. Case 1: 16<= driverAge < 20
5.   ageMultiplier = 2.8
6. If points < 1 Then
7.    safeDrivingReduction = 50
8. EndIf
9. Case 2: 20<= driverAge < 25
10.   ageMultiplier = 1.8
11. If points < 3 Then
12.    safeDrivingReduction = 50
13. EndIf
14. Case 3: 25<= driverAge < 45
15. ageMultiplier = 1#
16. If points < 5 Then
17.    safeDrivingReduction = 100
18. EndIf
19. Case 4: 45<= driverAge < 60
20. ageMultiplier = 0.8
21. If points < 7 Then
22.    safeDrivingReduction = 150
23. EndIf
24. Case 5: 60<= driverAge < 120
25. ageMultiplier = 1.5
26. If points < 5 Then
27.    safeDrivingReduction = 200
28. EndIf
29. Case 6: Else
30. Output ("Driver age out of range")
31. End Select
32. premium = baseRate * ageMultiplier - safeDrivingReduction
33. Output (premium)
```

sdR = 0

calcPrem

Age Points

Base - F(A, P)

10 test cases

500, 18, 0

500, 15, 5

500, ...

10 test cases

Automate

Test!!

cyclomatic complexity

- The cyclomatic complexity of the program graph of the insurance premium program is V(G) = e – n + 2p = 11,

edges points # Connected Component

- Jorgensen, Paul C. (2011-07-16). Software Testing (Page 175). Auerbach Publications. Kindle Edition.

11.3.1 Path–Based Testing

- Acyclic → only a finite number of paths exist

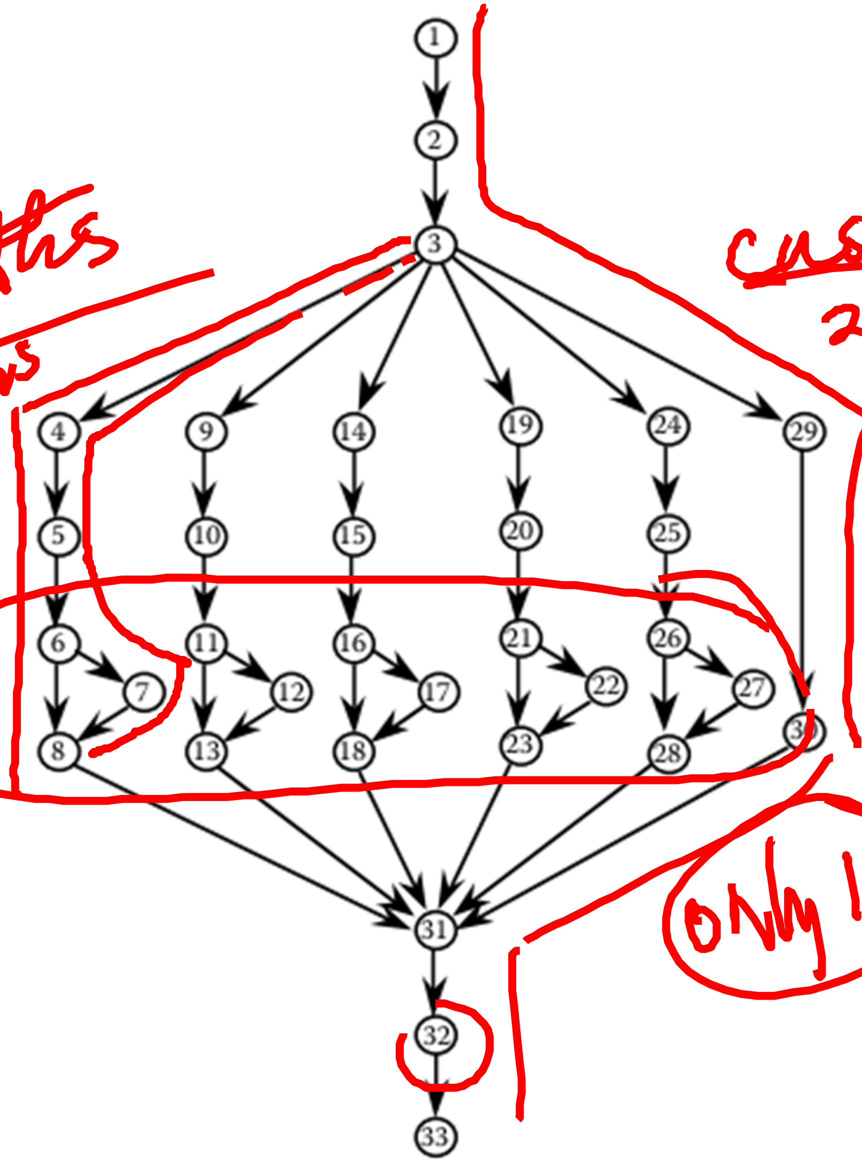
Insurance Program

Fig11.4

6x2 = 12 paths

5x2 + 1 = 11 paths

case 2



Only 1

Table 11.6 Paths in the Insurance Premium Program

Path	*Node Sequence*
p1	1–2–3–4–5–6–8–31–32–33
p2	1–2–3–4–5–6–7–8–31–32–33
p3	1–2–3–9–10–11–13–31–32–33
p4	1–2–3–9–10–11–12–13–31–32–33
p5	1–2–3–14–15–16–18–31–32–33
p6	1–2–3–14–15–16–17–18–31–32–33
p7	1–2–3–19–20–21–23–31–32–33
p8	1–2–3–19–20–21–22–23–31–32–33
p9	1–2–3–24–25–26–28–31–32–33
p10	1–2–3–24–25–26–27–28–31–32–33
p11	1–2–3–29–30–31–32–33

certain
(age, pt)
(18, 5)
(18, 0)
(18,)

For each A in Age
~~[illegible]~~ ∃ a threshold
$P_1 < T$
$P_2 > T$

Table 11.7 Path Coverage of Functional Methods in the Insurance Premium Program

Figure	*Method*	*Test Cases*	*Paths Covered*
8.7	Boundary value	25	p1, p2, p7, p8, p9, p10
8.8	Worst-case boundary value	273	p1, p2, p3, p4, p5, p6, p7, p8, p9, p10
8.9	Weak normal equivalence class	5	p2, p4, p6, p8, p9
8.9	Strong normal equivalence class	25	p1, p2, p3, p4, p5, p6, p7, p8, p9, p10
8.10	Decision table	10	p1, p2, p3, p4, p5, p6, p7, p8, p9, p10
8.11	Hybrid	25	p1, p2, p3, p4, p5, p6, p7, p8, p9, p10, p11

5 ages
5 points
point value
5 ages x 2 point = 10

There are more du paths than DD paths

- **Dataflow Testing**

du path

- **Jorgensen, Paul C. (2011-07-16). Software Testing (Page 177). Auerbach Publications. Kindle Edition.**

11.3.3 Slice Testing

- Slice testing does not provide much insight either. There are only four interesting slices:
- S(safeDrivingReduction, 32) = {1, 3, 4, 6, 7, 9, 11, 12 14, 16, 17, 19, 21, 22, 24, 26, 27, 31}
- S(ageMultiplier, 32) = {1, 3, 4, 5, 9, 10, 14, 15, 19, 20, 24, 25, 31}
- S(baseRate, 32) = {1}
- S(Premium, 31) = {2}
- The union of these slices

- References

- Brown, J.R. and Lipov, M., Testing for software reliability, Proceedings of the International Symposium on
- Reliable Software, Los Angeles, April 1975, pp. 518–527.
- ~~Pressman, R.S.~~, Software Engineering: A Practitioner's Approach, McGraw-Hill, New York, 1982.
- Jorgensen, Paul C. (2011-07-16). Software Testing (Page 177). Auerbach Publications. Kindle Edition.

Exercises

1. Repeat the gaps and redundancies analysis for the triangle problem using the structured implementation in Chapter 2 and its DD-Path graph in Chapter 9.
2. Compute the coverage, redundancy, and net redundancy metrics for your study in Exercise 1.
3. The pseudocode for the insurance premium program does not check for driver ages under 16 or (unlikely) over 120. The Else clause (case 6) will catch these, but the output message is not very specific. Also, the output statement (33) is not affected by the driver age checks. Which functional testing techniques will reveal this fault? Which structural testing coverage, if not met, will reveal this fault?