CSCE 747 Software Testing and Quality Assurance

Lecture 06 – Path Testing Part II

Last Time

- JUnit-4 Vogel tutorial finish
- Wrapup Functional Testing
- Ch 8 pp 117-127
- Testing Effort
- Testing Efficiency
- Testing Effectiveness
- Guidelines
- Case Study – Insurance Premium
- Slides 1-19 of this set were covered in class during Lec05
- Testing Overview Again
  - Definition of Testing
  - Verification vs Validation
- Path Testing
- Ch 9 pp 131-149

Today

- Lecture 06 Slides 1-19 covered last time
- Case Study Question after class
- Path Testing continued
- Ch 9 pp 131-149

Exam = Wednesday, Dec 11 @ 9:00AM

- **Email**
  - **I think we said it would be the earlier of two choices.**
  - **Again there were two choices since this is an APOGEE class and MW APOGEE classes follow the T-Th schedule.**
  - **So the two closest matches for MWF classes were:**

1. **Saturday, December 14 - 9:00 a.m.**
2. **Wednesday, December 11 - 9:00 a.m.**

I have heard no complaints from the class about doing the exam on Wednesday, December 11 @ 9:00AM.

So this is the exam time.
MM

Test Coverage Metrics

- Metric - Description of Coverage
- C_0 - Every statement
- C_1 - Every DD-Path (predicate outcome)
- C_{1p} - Every predicate to each outcome
- C_{12} - $C_1$1 -coverage + loop coverage
- C_d - $C_1$1 -coverage + every dependent pair of DD-Paths
- C_{MCC} - Multiple condition coverage
- C_{ik} - Every program path that contains up to k repetitions of a loop (usually k = 2)
- C_{stat} - Statistically significant fraction of paths
- C_∞ - All possible execution paths
  - **Based on work of E.F. Miller (Miller, 1977)**

Questions from last class

- From Lec05 slide 24

Table 8.3 Decision Table Test Cases for the Insurance Premium Program

Age Is	*16–25*	*16–25*	*25–35*	*25–35*	*35–45*	*35–45*	*45–60*	*45–60*	*60–100*	*60–100*
Points	0	1–12	0–2	3–12	0–4	5–12	0–6	7–12	0–4	5–12
Age multiplier	2.8	2.8	1.8	1.8	1	1	0.8	0.8	1.5	1.5
Safe driving reduction	50	—	50	—	100	—	150	—	200	—

- Note that quite the usual form of Decision table.
- Conditions A1-A5, P1-P5
  - 10 conditions give 2^{10} =1024 rules
- So what is that 2^{10}?

Lec07 slide 18 - Safe Driving Reduction Table Points cutoffs

Age Range	*Age Multiplier*	*Points Cutoff*	*Safe Driving Reduction*
$16 \leq age < 25$	2.8	1	50
$25 \leq age < 35$	1.8	3	50
$35 \leq age < 45$	1.0	5	100
$45 \leq age < 60$	0.8	7	150
$60 \leq age < 100$	1.5	5	200

- **Note two cases for each age!**

Decision Table for Case Study: Insurance Premium Program Ch08 1rst try

- Woops!
- 512!=1024

Cond /Acts	R1	R2	R3	R4	R5	R6	R7	R8	R9	R10
A1?	T	T	F	F	F	F	F	F	F	F
A2?	-	-	T	T	F	F	F	F	F	F
A3?	-	-	-	-	T	T	F	F	F	F
A4?	-	-	-	-	-	-	T	T	F	F
A5?	-	-	-	-	-	-	-	-	T	T
P1?	T	F	-	-	-	-	-	-	-	-
P2?	F	T	T	F	-	-	-	-	-	-
P3?	-	-	F	T	T	F	-	-	-	-
P4?	-	-	-	-	F	T	T	F	-	-
P5?	-	-	-	-	-	-	F	T	T	F
Rule count	2^7 128	2^7 256	2^6 320	2^6 384	2^5 416	2^5 448	2^4 464	2^4 480	2^4 496	2^4 512
Act's										

Decision Table for Case Study: Insurance Premium Program Ch08 2nd try

- ????

Cond /Acts	R1	R2	R3	R4	R5	R6	R7	R8	R9	R10
A1?	T	T	F	F	F	F	F	F	F	F
A2?	-	-	T	T	F	F	F	F	F	F
A3?	-	-	-	-	T	T	F	F	F	F
A4?	-	-	-	-	-	-	T	T	F	F
A5?	-	-	-	-	-	-	-	-	T	T
P1?	T	F	-	-	-	-	-	-	-	-
P2?	F	T	T	F	-	-	-	-	-	-
P3?	-	-	F	T	T	F	-	-	-	-
P4?	-	-	-	-	F	T	T	F	-	-
P5?	-	-	-	-	-	-	F	T	T	F
Rule count	2^7 128	2^7 256	2^6 320	2^6 384	2^5 416	2^5 448	2^4 464	2^4 480	2^4 496	2^4 512
Act's										

- **Why? Explanation ???**
- **Note ignored some errors!**
  - **Point cases that matter are different for different ages**
  - **This means that can't us induced equiv. classes on product A x P**
  - **But there are two equivalence classes for each A_j even though the P_i depends on A_j.**

DD-Paths revisited

- Program graphs
- basic blocks = DD-paths ?
  - one entry=leader

- **Miller's test coverage metrics are based on program graphs in which nodes are full statements,**

- **Most quality organizations now expect the C1 metric (DD-Path coverage) as the minimum acceptable level of test coverage.**

Metric-Based Testing

- Metric-Based Testing
- Statement and Predicate Testing
- What's a statement anyway?
- S→ IF Condition THEN S1 ELSE S2

- predicate outcome coverage.

- Really better to use ideas from compilers
  - cover each basic block

DD-Path Testing

- For if-then and if-then-else statements, this means that both the true and the false branches are covered (C1p coverage).
- For CASE statements, each clause is covered.

Dependent Pairs of DD-Paths

- C_d,
- the problem of infeasible paths.

Multiple Condition Coverage

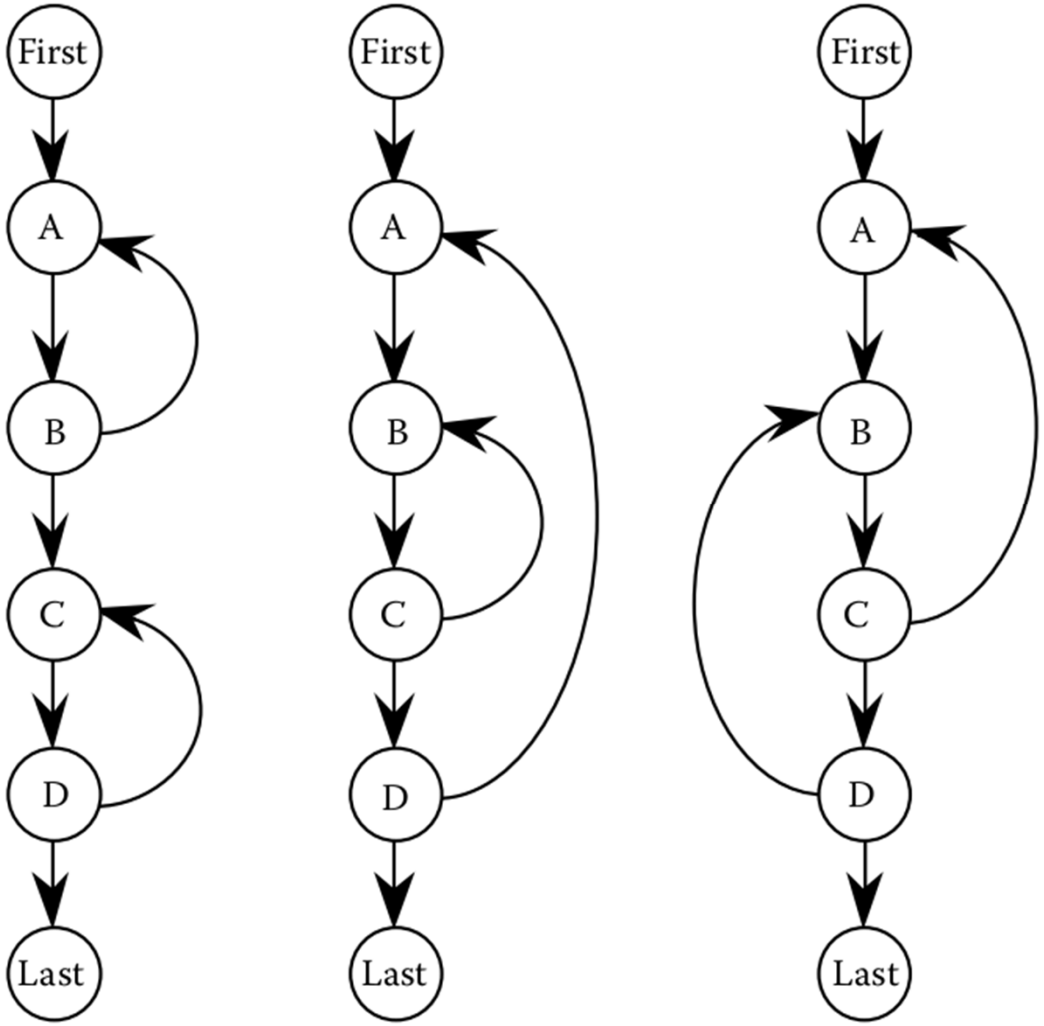
- **Look closely at the compound conditions in DD-Paths B and H.**
- **Instead of simply traversing such predicates to their true and false outcomes, we should investigate the different ways that each outcome can occur.**
- **One possibility is to make a truth table; a compound condition of three simple conditions would have eight rows, yielding eight test cases.**
- **Another possibility is to reprogram compound predicates into nested simple if-then-else logic, which will result in more DD-Paths to cover**.
- We see an interesting trade-off: statement complexity versus path **complexity.**

Loop Coverage

- loops are a highly fault-prone portion of source code.

- Beizer taxonomy

Loops: Concatenated, Nested, Knotted



Software Testing A Craftsman's Approach
Jorgensen – 2008

- Knotted loops cannot occur when the structured programming precepts are followed,
- but they can occur in languages like Java with try/catch.
- simple view of loop testing is that every loop involves a decision, and we need to test both outcomes of the decision:

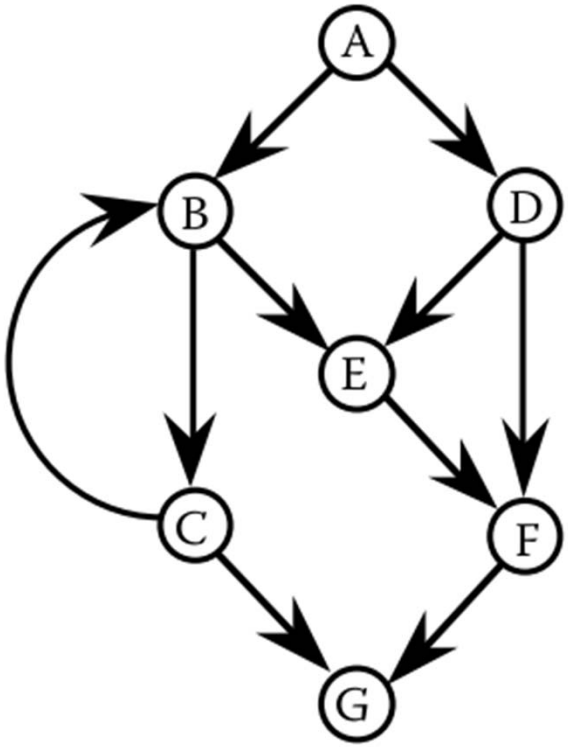
9.2.2 Test Coverage Analyzers

Basis Path Testing

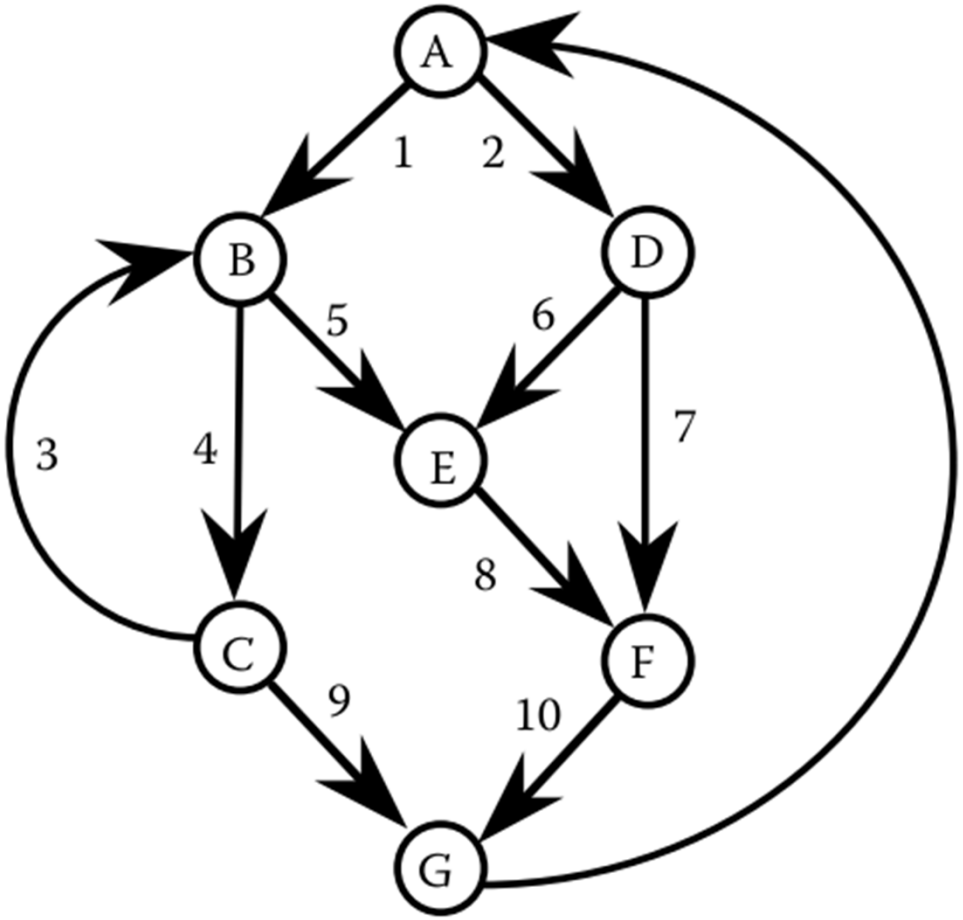
- Recall vector spaces and basis
- "view a program as a vector space, then the basis for such a space would be a very interesting set of elements to test"

McCabe's Basis Path Method

- McCabe's control graph
- cyclomatic number
- V(G) = e – n + 2p = 10 – 7



$$V(G) = e - n + p = 11 - 7 + 1 = 5$$



- The cyclomatic complexity of the strongly connected graph in Figure 9.7 is 5;
- thus, there are five linearly independent circuits.
- p1: A, B, C, G
- p2: A, B, C, B, C, G
- p3: A, B, E, F, G
- p4: A, D, E, F, G
- p5: A, D, F, G

Table 9.3 Path/Edge Traversal

Path/Edges Traversed	1	2	3	4	5	6	7	8	9	10
p1: A, B, C, G	1	0	0	1	0	0	0	0	1	0
p2: A, B, C, B, C, G	1	0	**1**	2	0	0	0	0	1	0
p3: A, B, E, F, G	1	0	0	0	**1**	0	0	1	0	1
p4: A, D, E, F, G	0	1	0	0	0	**1**	0	1	0	1
p5: A, D, F, G	0	1	0	0	0	0	**1**	0	0	1
ex1: A, B, C, B, E, F, G	1	0	1	1	1	0	0	1	0	1
ex2: A, B, C, B, C, B, C, G	1	0	2	3	0	0	0	0	1	0

McCabe's baseline method

- McCabe's baseline method to determine a set of basis paths.

1. Select a baseline path, the "normal case"
2. Next the baseline path is retraced
3. in turn each decision is "flipped"

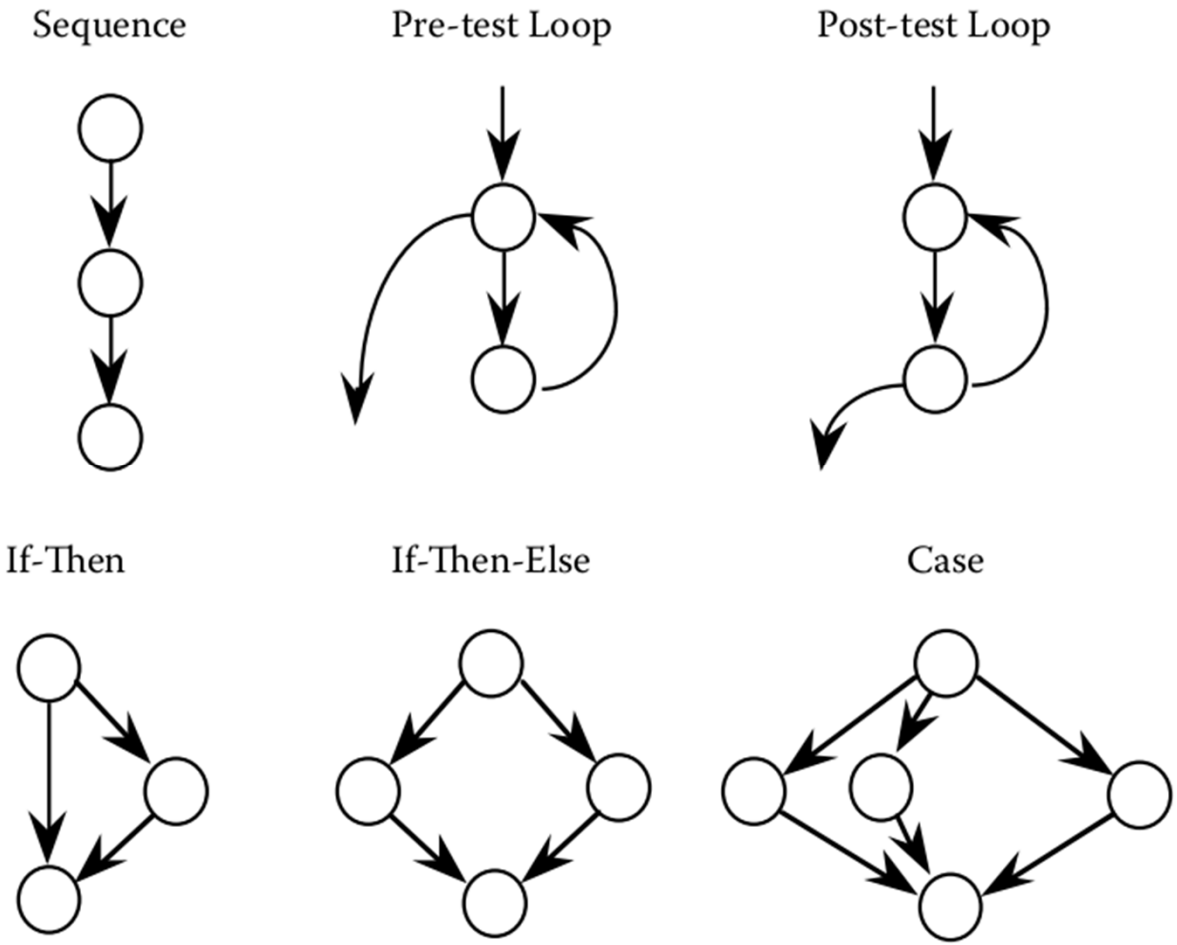
McCabe's Method on Triangle

Table 9.4 Basis Paths in Figure 9.4

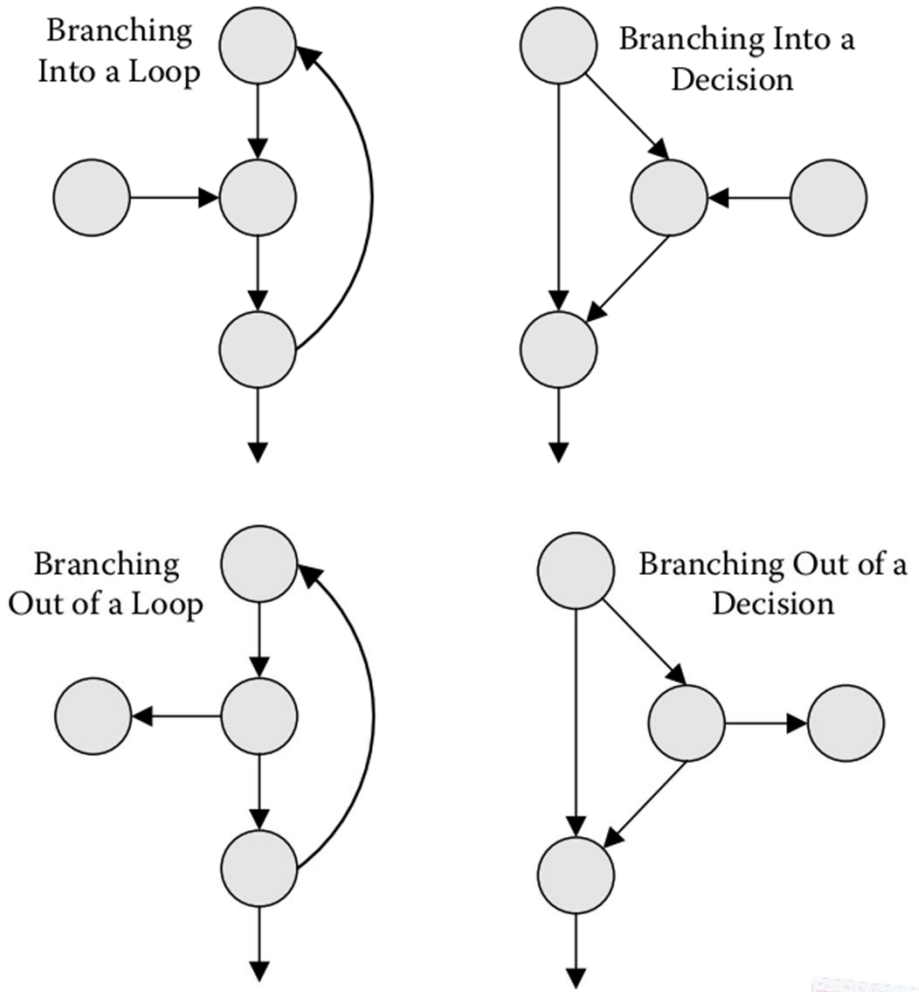
Original	p1: A-B-C-E-F-H-J-K-M-N-O-Last	Scalene
Flip p1 at B	p2: A-B-D-E-F-H-J-K-M-N-O-Last	Infeasible
Flip p1 at F	p3: A-B-C-E-F-G-O-Last	Infeasible
Flip p1 at H	p4: A-B-C-E-F-H-I-N-O-Last	Equilateral
Flip p1 at J	p5: A-B-C-E-F-H-J-L-M-N-O-Last	Isosceles

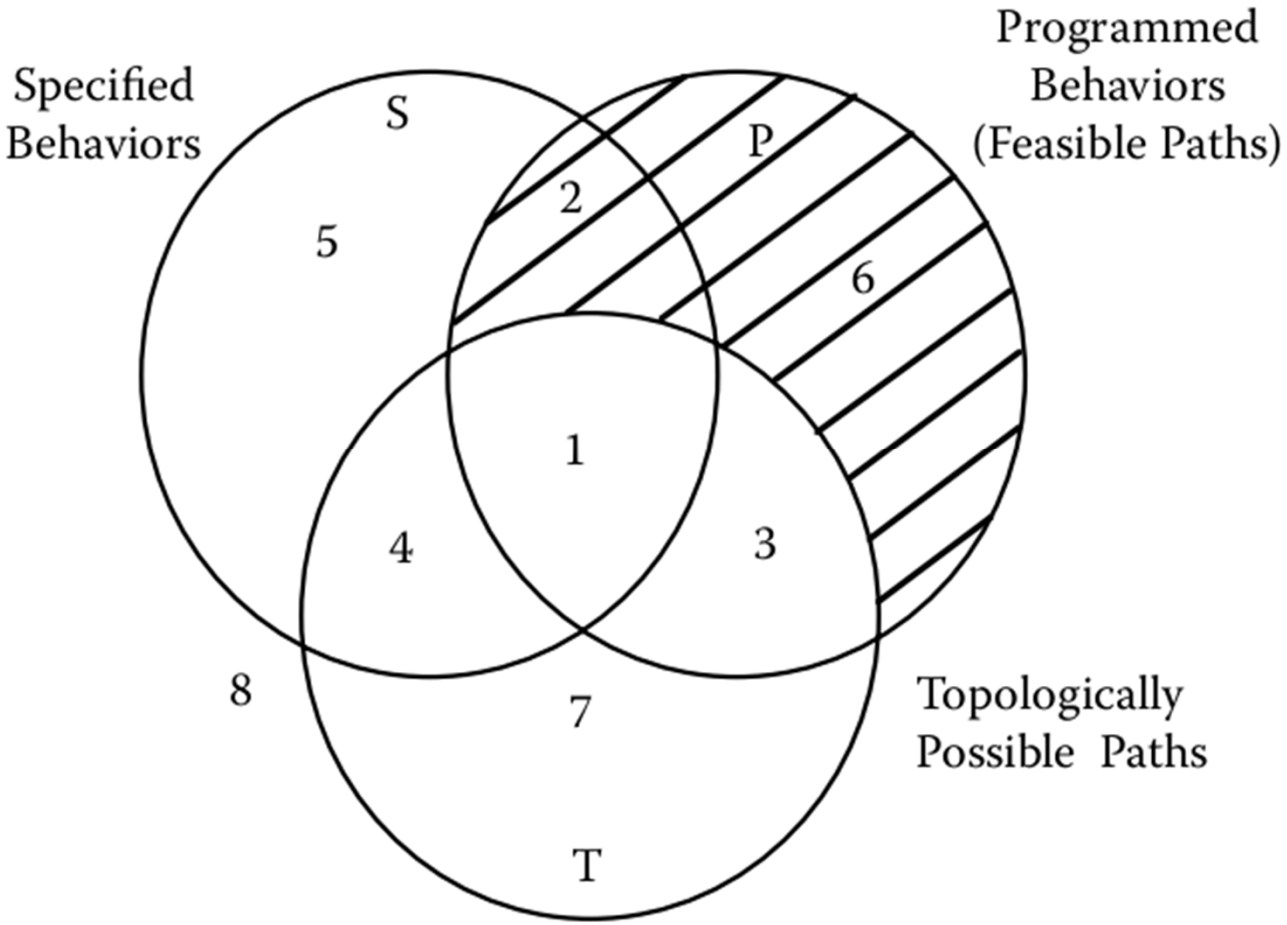
Essential Complexity

- x



Violations of structured programming




Specified
Behaviors
S
5
Programmed
Behaviors
(Feasible Paths)
P
2
6
1
4
3
8
7
Topologically
Possible Paths
T

References

- McCabe, T.J., Structural Testing: A Software Testing Methodology Using the Cyclomatic Complexity Metric, 1982.
- Beizer, Boris, Software System Testing and Quality Assurance, 1984.