G.PULLAIAH COLLEGE OF ENGINEERING & TECHNOLOGY

(Accredited by NAAC with 'A' Grade of UGC Approved by AICTE, New Delhi & Permanently Affiliated to JNTUA, Ananthapuramu)

(Recognized by UGC under 2(f) & 12(B) &ISO 9001:2008 Certified Institution)

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

NAME OF THE SUBJECT: SOFTWARE TESTING METHODOLOGIES

SUBJECT CODE: 13A05605

YEAR/SEM: III B.Tech II SEM

NAME OF THE FACULTY: R.Sandeep Kumar

UNIT-1 INTROCUTION, FLOW GRAPH AND PATH TESTING

UNIT-2: TRANSITION FLOW TESTING, DATA FLOW TESTING

2 marks questions in UNIT-1 and UNIT-2

1. **Define Software Testing?**
   It a process of executing a program or application with the intent of finding the bugs (or)
   It involves any activity aimed at evaluating an attribute (or) capability of a program or system and determining that it meets its required results.
2. **Define Pesticide Paradox and Complexity barrier?**
   Pesticide Paradox: Every method you use to prevent or find bugs leaves a residue of subtler bugs against which those methods are ineffectual**.**
   Complexity barrier: - Software complexity grows to limits of our ability to manage that complexity.
3. **Define path testing and transaction flow testing?**
   Path Testing is the name given to a family of test techniques based on judiciously selecting a set of test paths through the program.
   Transaction flow testing: A transaction is a unit of work seen from a system user's point of view.
   A transaction consists of a sequence of operations, some of which are performed by a system, persons or devices that are outside of the system.
4. **Define path instrumentation and path interpretation?**
   Path Instrumentation: - What we have to do conform that the outcome was achieved by the intended path.
   Path Interpretation:- The act of symbolic substitution of operations along the path in order to express the predicate solely in terms of the input vector.
5. **Define path sensitization and path predicates?**
   Path Sensitizations: The act of finding a set of solutions for the path predicate expressions is said to be path sensitization.
   Path predicate: Path predicates are the specific form of predicates of the decision along the selected path after interpretation.
6. **Differentiate between verification and validation?**

Validation is the process of evaluating software at the end of the development process to determine whether software meets the customer expectations and requirements.
Validation is carried out by testing team.
Validation activity is carried out just after the Verification.
The objective of **Verification** is to make sure that the product being develop is as per the requirements and design specifications.
Verification is carried out by QA team to check whether implementation software is as per specification document or not.
Verification is carried out before the Validation.

7. **Differentiate between testing and debugging?**

Testing	Debugging
Testing starts with known conditions, uses predefined procedures and has predictable outcomes.	Debugging starts from possibly unknown initial conditions and the end cannot be predicted except statistically.
Testing can and should be planned, designed and scheduled.	Procedure and duration of debugging cannot be so constrained.
Testing is a demonstration of error or apparent correctness.	Debugging is a deductive process.
Testing proves a programmer's failure.	Debugging is the programmer's vindication (Justification).
Testing, as executes, should strive to be predictable, dull, constrained, rigid and inhuman.	Debugging demands intuitive leaps, experimentation and freedom.
Much testing can be done without design knowledge.	Debugging is impossible without detailed design knowledge.
Testing can often be done by an outsider.	Debugging must be done by an insider.
Much of test execution and design can be automated.	Automated debugging is still a dream.

8. **Differentiate between failure and fault?**
Fault, Error and **Failure**. **Fault** : It is a condition that causes the software to fail to perform its required function. Error : Refers to **difference between** Actual Output and Expected output.
Fault: Discrepancy in code that causes a **failure**.

Fault : It is a condition that causes the software to fail to perform its required function.

Failure : It is the inability of a system or component to perform required function according to its specification.

9. **Differentiate between a flow chart and a flow graph?**

   i) A program's flow chart resembles a control flow graph.
   ii) In flow graphs, we don't show the details of what is in a process block.
   iii) In flow charts every part of the process block is drawn.
   iv) The flowchart focuses on process steps, where as the flow graph focuses on control flow of the program.
   v) The act of drawing a control flow graph is a useful tool that can help us clarify the control flow and data flow issues.

10. **List the various iterations of Loop Testing?**

Nested Loops:- The number of tests to be performed on nested loops will be the exponent of the tests performed on single loops.

Concatinated Loops: Concatenated loops fall between single and nested loops with respect to test cases. Two loops are concatenated if it's possible to reach one after exiting the other while still on a path from entrance to exit.

Horrible Loops: A horrible loop is a combination of nested loops, the use of code that jumps into and out of loops, intersecting loops, hidden loops, and cross connected loops.

11. **Define loop free path segment?**

Loop-Free Path Segment is a path segment for which every node in it is visited at most once.

12. **What are the two different goals of testing?**

1. Bug Prevention: A prevented bug is better than a detected bug and corrected bug because if the bug is prevented there's no code to correct. i.e no retesting is needed to confirm that the correction was valid. Designing tests is one of the best bug preventers known.

2. Test design Thinking:- The thinking that must be done to create a useful test can discover and eliminate bugs before they are coded i.e eliminating the bugs at every stage is creation of software i.e a) from the specification write tests specifications first and then code.

b) Eliminate bugs at every stage of SDLC. c) If this fails testing is to detect the remaining bugs.

13. **List the different components in modelling a real world project?**

1) Application 2) Staff 3) Schedule 4) Specification 5) Acceptance test 6) Personnel 7) Standards 8) Objectives 9) Source 10) History

14. **Write the metric for bug importance?**

Importance of bug ($) = Frequency * (Correction cost + Installation Cost + Consequential Cost)

15. **List out the different consequences of bugs?**

1) Mild 2) Moderate 3) Annoying 4) Disturbing 5) Serious 6) Very Serious 7) Extreme 8) Intolerable 9) Catastrophic 10) Infectious

16. List the terminologies used in Strategies of data flow testing?

1) Definition clear path 2) Definition du path 3) Loop free path segment

17. **Differentiate between Modularity Vs Efficiency?**

Modularity Versus Efficiency: A module is a discrete, well-defined, small component of a system. Smaller the modules, difficult to integrate; larger the modules, difficult to understand. Both tests and systems can be modular. Testing can and should likewise be organised into modular components. Small, independent test cases can be designed to test independent modules.

18. **Differentiate between Builder Vs Buyer?**

Builder Versus Buyer: Most software is written and used by the same organization. Unfortunately, this situation is dishonest because it clouds accountability. If there is no separation between builder and buyer, there can be no accountability.

The different roles / users in a system include:

i) **Builder:** Who designs the system and is accountable to the buyer.

ii) **Buyer:** Who pays for the system in the hope of profits from providing services.

iii) **User:** Ultimate beneficiary or victim of the system. The user's interests are also guarded by.

iv) **Tester:** Who is dedicated to the builder's destruction.

v) **Operator:** Who has to live with the builders' mistakes, the buyers' murky (unclear) specifications, testers' oversights and the users' complaints.

19. Define path testing?

Path Testing is the name given to a family of test techniques based on judiciously selecting a set of test paths through the program.

20. **Define path segment and process block?**

A path segment is succession of consecutive links that belongs to some path. The Length of path measured by the number of links in it and not by the number of the instructions or statements executed along that path.

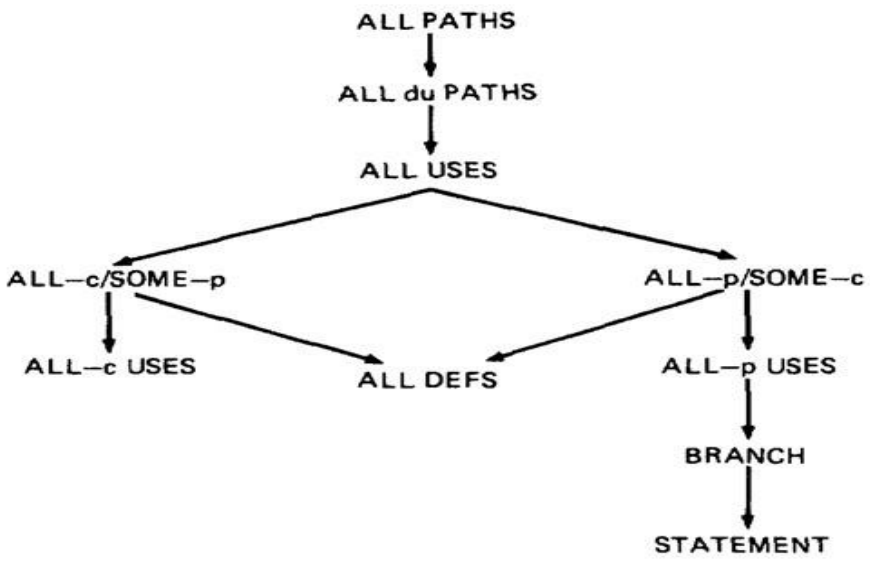
21. Define Definition-clear path segment?

Definition-Clear Path Segment, with respect to variable X, is a connected sequence of links such that X is (possibly) defined on the first link and not redefined or killed on any subsequent link of that path segment

22.Define du path?

A **du path** from node i to k is a path segment such that if the last link has a computational use of X, then the path is simple and definition-clear; if the penultimate (last but one) node is j - that is, the path is (i,p,q,...,r,s,t,j,k) and link (j,k) has a predicate use - then the path from i to j is both loop-free and definition-clear.

23. Depict the pictorial representation of ordering the strategies?



24. Define slice and dice?

A (static) program **slice** is a part of a program (e.g., a selected set of statements) defined with respect to a given variable X (where X is a simple variable or a data vector) and a statement i: it is the set of all statements that could (potentially, under static analysis) affect the value of X at statement i - where the influence of a faulty statement could result from an improper computational use or predicate use of some other variables at prior statements.

A program **dice** is a part of a slice in which all statements which are known to be correct have been removed.

25. What is the practical suggestion in path testing?

1. Draw the control flow graph on a single sheet of paper.
2. Make several copies - as many as you will need for coverage (C1+C2) and several more.
3. Use a yellow highlighting marker to trace paths. Copy the paths onto master sheets.
4. Continue tracing paths until all lines on the master sheet are covered, indicating that you appear to have achieved C1+C2.
5. As you trace the paths, create a table that shows the paths, the coverage status of each process, and each decision.

26. What is a path selection criterion?

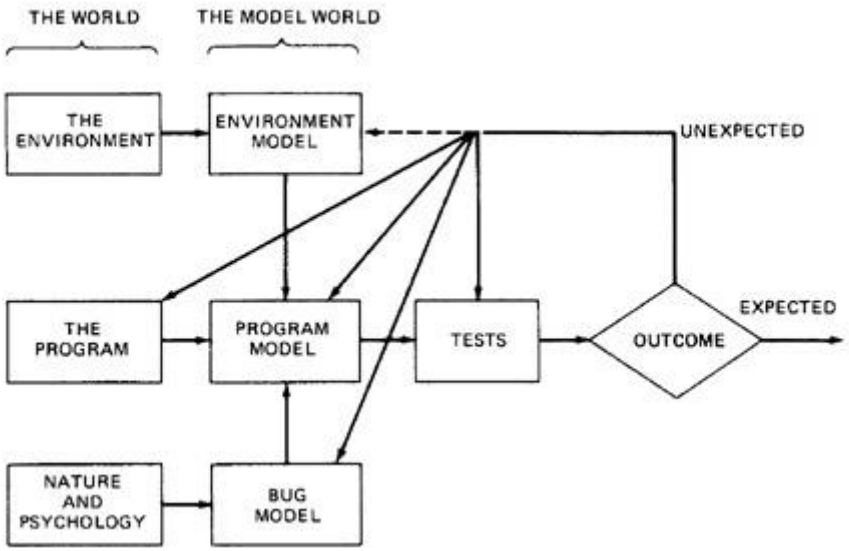
Three different testing criteria or strategies out of a potentially infinite family of strategies.

0. **Path Testing (P_{inf})**:
   Execute all possible control flow paths through the program: typically, this is restricted to all possible entry/exit paths through the program.
   If we achieve this prescription, we are said to have achieved 100% path coverage. This is the strongest criterion in the path testing strategy family: it is generally impossible to achieve.
1. **Statement Testing (P_1)**:
   Execute all statements in the program at least once under some test. If we do enough tests to achieve this, we are said to have achieved 100% statement coverage.
   An alternate equivalent characterization is to say that we have achieved 100% node coverage. We denote this by C1.

This is the weakest criterion in the family: testing less than this for new software is unconscionable (unprincipled or can not be accepted) and should be criminalized.

2. **Branch Testing (P_2)**:
   Execute enough tests to assure that every branch alternative has been exercised at least once under some test.
   If we do enough tests to achieve this prescription, then we have achieved 100% branch coverage.
   An alternative characterization is to say that we have achieved 100% link coverage.
   For structured software, branch testing and therefore branch coverage strictly includes statement coverage.
   We denote branch coverage by C2.

27. Depict the pictorial representation of model for a testing?



28. Define Benign bug hypothesis and bug locality hypothesis?

Benign Bug Hypothesis: The belief that bugs are nice, tame and logical. (Benign: Not Dangerous)

Bug Locality Hypothesis: The belief that a bug discovered with in a component effects only that component's behaviour.

29. Define Lingua Salvator Est and Silver Bullets?

Lingua Salvator Est: The belief that the language syntax and semantics (e.g. Structured Coding, Strong typing, etc) eliminates most bugs.

Silver Bullets: The mistaken belief that X (Language, Design method, representation, environment) grants immunity from bugs.

30. Define Sadism Suffices and Angelic Testers?

Sadism Suffices: The common belief (especially by independent tester) that a sadistic streak, low cunning, and intuition are sufficient to eliminate most bugs. Tough bugs need methodology and techniques.

Angelic Testers: The belief that testers are better at test design than programmers is at code design.

31. Importance of bugs depends on?

IMPORTANCE OF BUGS: The importance of bugs depends on frequency, correction cost, installation cost, and consequences.

1. **Frequency:** How often does that kind of bug occur? Pay more attention to the more frequent bug types.

2. **Correction Cost:** What does it cost to correct the bug after it is found? The cost is the sum of 2 factors: (1) the cost of discovery (2) the cost of correction. These costs go up dramatically later in the development cycle when the bug is discovered. Correction cost also depends on system size.
3. **Installation Cost:** Installation cost depends on the number of installations: small for a single user program but more for distributed systems. Fixing one bug and distributing the fix could exceed the entire system's development cost.
4. **Consequences:** What are the consequences of the bug? Bug consequences can range from mild to catastrophic.

32. What are the different elements of a flow graph?

1) Process block 2) Decision 3) Case Statements 4) Junctions

33. Define Transaction?

A transaction is a unit of work seen from a system user's point of view.
A transaction consists of a sequence of operations, some of which are performed by a system, persons or devices that are outside of the system.

34. Write one example for a transaction?

A transaction for an online information retrieval system might consist of the following steps or tasks:

- Accept input (tentative birth)
- Validate input (birth)
- Transmit acknowledgement to requester
- Do input processing
- Search file
- Request directions from user
- Accept input
- Validate input
- Process request
- Update file
- Transmit output
- Record transaction in log and clean up (death)

35. Different types of births?

Births: There are three different possible interpretations of the decision symbol, or nodes with two or more out links. It can be a Decision, Biosis or Mitosis.

a. Decision: Here the transaction will take one alternative or the other alternative but not both.

b. Biosis: Here the incoming transaction gives birth to a new transaction, and both transaction continue on their separate paths, and the parent retains its identity.

c. Mitosis: Here the parent transaction is destroyed and two new transactions are created.

36. Different types of mergers?

Mergers: Transaction flow junction points are potentially as troublesome as transaction flow splits. There are three

types of junctions: (1) Ordinary Junction (2) Absorption (3) Conjugation

0. **Ordinary Junction:** An ordinary junction which is similar to the junction in a control flow graph. A transaction can arrive either on one link or the other.
1. **Absorption:** In absorption case, the predator transaction absorbs prey transaction. The prey gone but the predator retains its identity.
2. **Conjugation:** In conjugation case, the two parent transactions merge to form a new daughter. In keeping with the biological flavor this case is called as conjugation.

5. What are the four distinct states of a variable/object?

1. **K** :- undefined, previously killed, doesnot exist
2. **D** :- defined but not yet used for anything
3. **U** :- has been used for computation or in predicate
4. **A** :- anomalous

These capital letters (K, D, U, A) denote the state of the variable.

37. Depict the pictorial representation of unforgiving and forgiving data flow anomaly model?

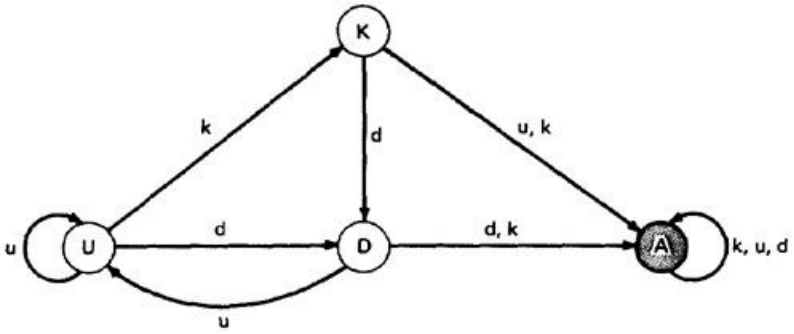


Figure Unforgiving Data Flow Anomaly State Graph

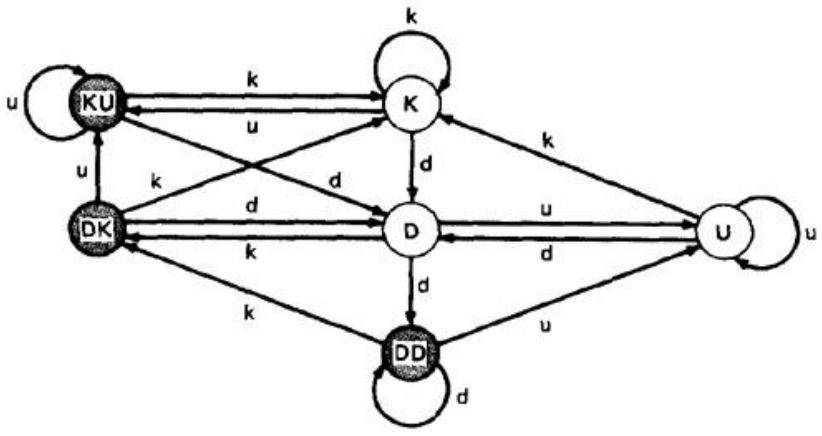


Figure Forgiving Data Flow Anomaly State Graph