

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-3 DOMAIN TESTING

2 MARKS QUESTIONS

1. Define Domain?

Domain: In mathematics, domain is a set of possible values of an independent variable or the variables of a function.

2. Define Domain closure?

A domain boundary is **closed** with respect to a domain if the points on the boundary belong to the domain. If the boundary points belong to some other domain, the boundary is said to be **open**.

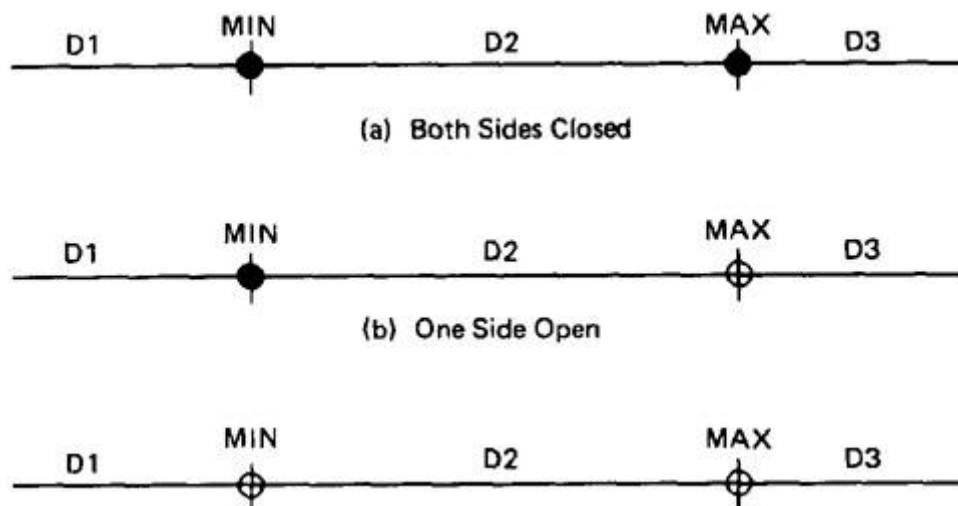


Figure: Open and Closed Domains.

3. Why loop free software creates trouble for domain testing?

Loops are problematic for domain testing. The trouble with loops is that each iteration can result in a different predicate expression (after interpretation), which means a possible domain boundary change.

4. What are the properties of nice domain?

Properties of nice domains are: Linear, Complete, Systematic, Orthogonal, Consistently closed, Convex and Simply connected.

5. Write about Linear and non linear boundaries?

LINEAR AND NON LINEAR BOUNDARIES: Nice domain boundaries are defined by linear inequalities or equations. The impact on testing stems from the fact that it takes only two points to determine a straight line and three points to determine a plane and in general $n+1$ point to determine a n -dimensional hyper plane.

In practice more than 99.99% of all boundary predicates are either linear or can be linearized by simple variable transformations.

6. When can we say that the boundary is complete and incomplete?

Nice domain boundaries are complete in that they span the number space from plus to minus infinity in all dimensions.

The advantage of complete boundaries is that one set of tests is needed to confirm the boundary no matter how many domains it bounds.

If the boundary is chopped up and has holes in it, then every segment of that boundary must be tested for every domain it bounds.

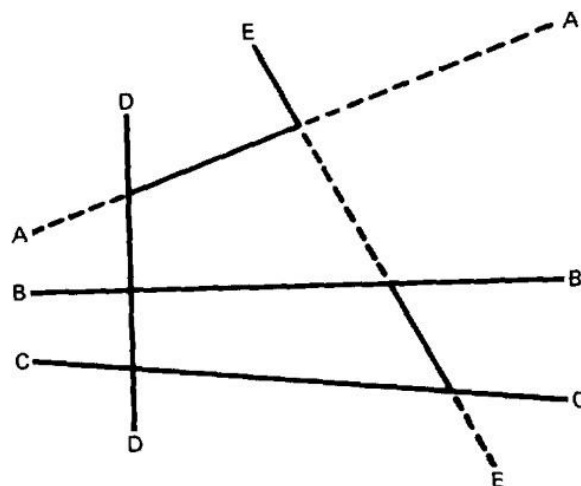


Figure: Incomplete Domain Boundaries.

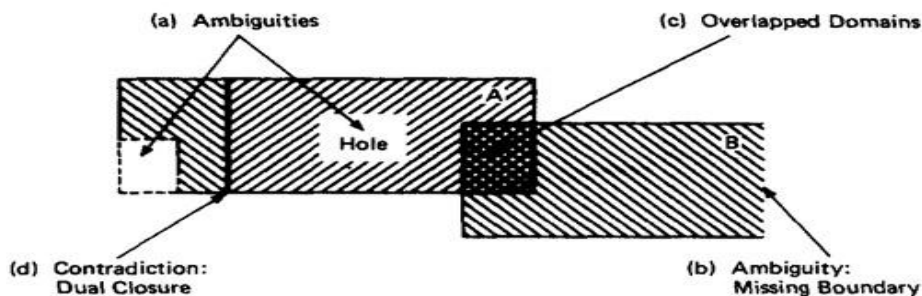
7. Define systematic boundary?

Systematic boundary means that boundary inequalities related by a simple function such as a constant.

$$\begin{array}{l} f_1(X) \geq k_1 \text{ or } f_1(X) \geq g(1,c) \\ f_1(X) \geq k_2 \quad f_2(X) \geq g(2,c) \\ \dots\dots\dots \quad \dots\dots\dots \\ f_i(X) \geq k_i \quad f_i(X) \geq g(i,c) \end{array}$$

where f_i is an arbitrary linear function, X is the input vector, k_i and c are constants, and $g(i,c)$ is a decent function over i and c that yields a constant, such as $k + ic$.

8. Write and depict the pictorial representation of ambiguity and contradictions?



Domain ambiguities are holes in the input space.

The holes may lie within the domains or in cracks between domains.

Two kinds of contradictions are possible: overlapped domain specifications and overlapped closure specifications

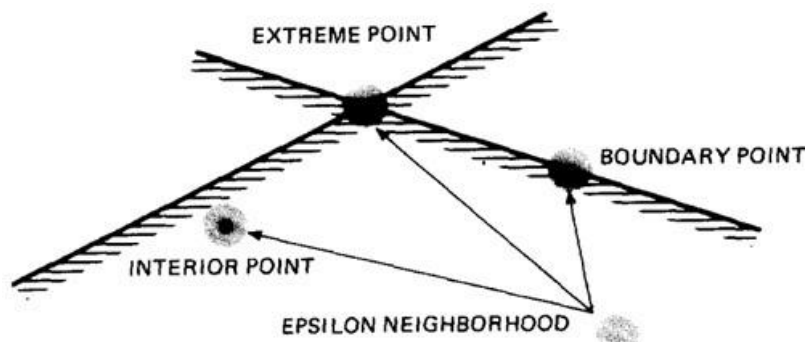
Figure shows overlapped domains and Figure shows dual closure assignment.

9. Write short notes on domain testing strategy?

- Domains are defined by their boundaries; therefore, domain testing concentrates test points on or near boundaries.
- Classify what can go wrong with boundaries, then define a test strategy for each case. Pick enough points to test for all recognized kinds of boundary errors.
- Because every boundary serves at least two different domains, test points used to check one domain can also be used to check adjacent domains. Remove redundant test points.
- Run the tests and by post-test analysis (the tedious part) determine if any boundaries are faulty and if so, how.
- Run enough tests to verify every boundary of every domain.

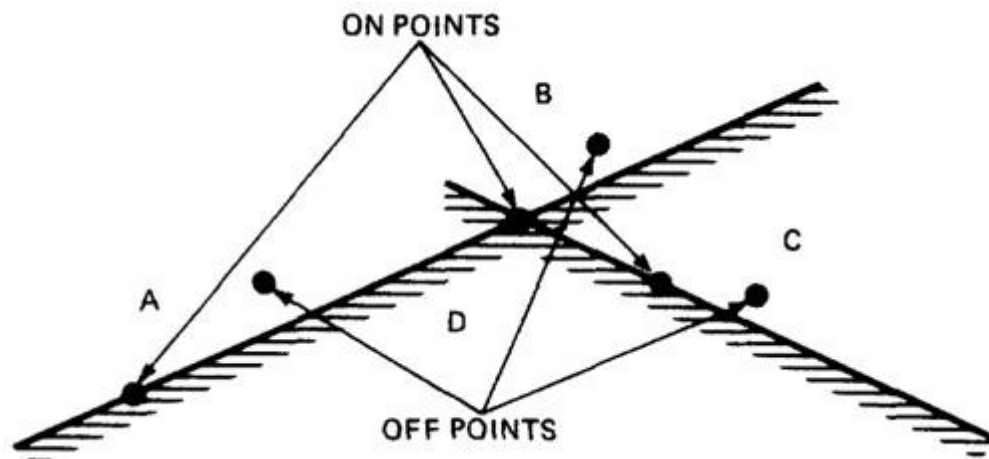
10. Define interior point, exterior point and boundary point?

1. An **interior point** is a point in the domain such that all points within an arbitrarily small distance (called an epsilon neighbourhood) are also in the domain.
2. A **boundary point** is one such that within an epsilon neighbourhood there are points both in the domain and not in the domain.
3. An **extreme point** is a point that does not lie between any two other arbitrary but distinct points of a (convex) domain.



11. Define on point and off point?

1. An **on point** is a point on the boundary.
2. If the domain boundary is closed, an **off point** is a point near the boundary but in the adjacent domain.
3. If the boundary is open, an off point is a point near the boundary but in the domain being tested. the acronym COOOOI: Closed Off Outside, Open off Inside.



12. List some generic domain bugs with pictorial representations?

Closure bug, shifted boundaries, tilted boundaries, extra boundary, missing boundary.

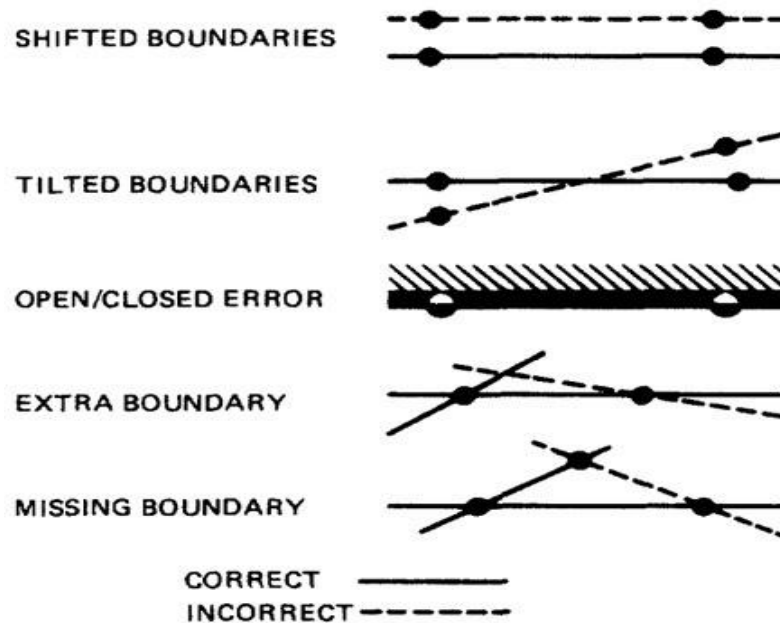


Figure: Generic Domain Bugs

13. When can we say that closure bug occurs in 2D bugs?

Closure Bug: Figure shows a faulty closure, such as might be caused by using a wrong operator (for example, $x \geq k$ when $x > k$ was intended, or vice versa). The two on points detect this bug because those values will get B rather than A processing.

14. When can we say that closure bug occurs in 2D bugs?

Shifted Boundary: In Figure the bug is a shift up, which converts part of domain B into A processing, denoted by A'. This result is caused by an incorrect constant in a predicate, such as $x + y \geq 17$ when $x + y \geq 7$ was intended. The off point (closed off outside) catches this bug. Figure 4.15c shows a shift down that is caught by the two on points.

15. When can we say that closure bug occurs in 2D bugs?

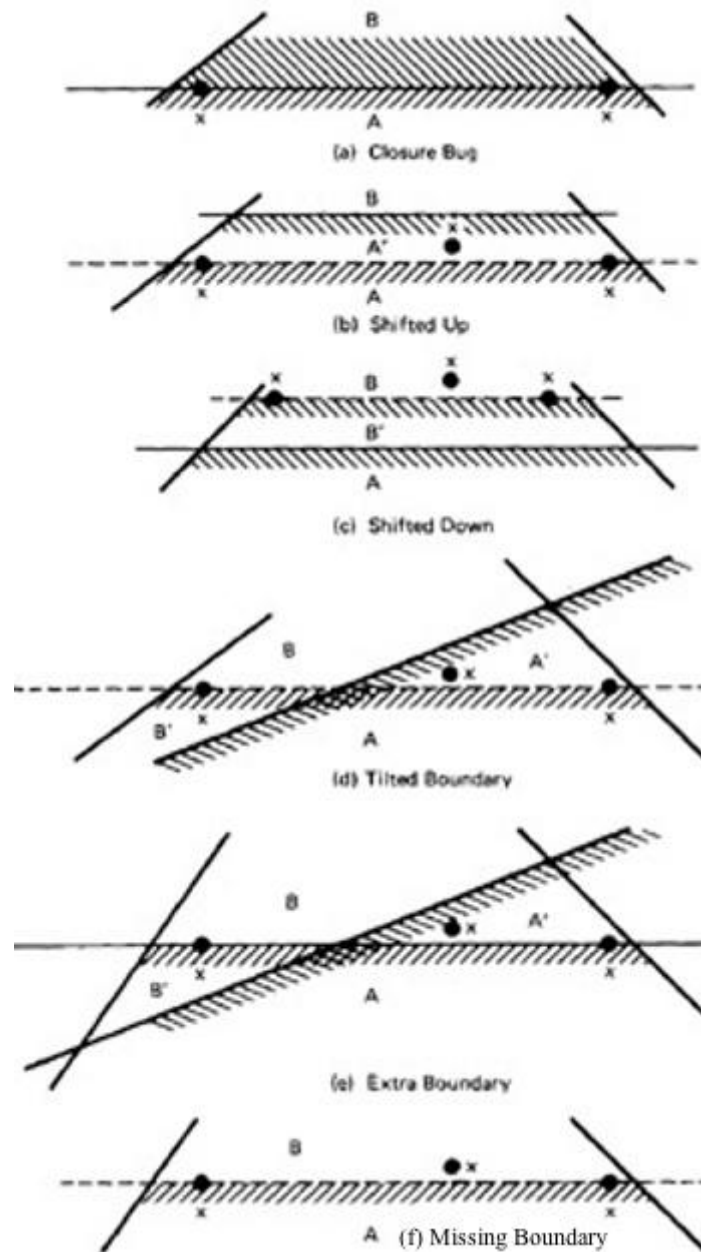
Tilted Boundary: A tilted boundary occurs when coefficients in the boundary inequality are wrong. For example, $3x + 7y > 17$ when $7x + 3y > 17$ was intended. Figure has a tilted boundary, which creates erroneous domain segments A' and B'. In this example the bug is caught by the left on point.

16. When can we say that closure bug occurs in 2D bugs?

Extra Boundary: An extra boundary is created by an extra predicate. An extra boundary will slice through many different domains and will therefore cause many test failures for the same bug. The extra boundary in Figure is caught by two on points, and depending on which way the extra boundary goes, possibly by the off point also.

17. When can we say that closure bug occurs in 2D bugs?

Missing Boundary: A missing boundary is created by leaving a boundary predicate out. A missing boundary will merge different domains and will cause many test failures although there is only one bug. A missing boundary, shown in Figure is caught by the two on points because the processing for A and B is the same - either A or B processing.



18. What is Interface Testing?

Interface Testing is performed to evaluate whether systems or components pass data and control correctly to one another. It is to verify if all the interactions between these modules are working properly and errors are handled properly.

19. Define range, domain and domain span?

The set of output values produced by a function is called the range of the function, in contrast with the domain, which is the set of input values over which the function is defined. For a single variable, the domain span is the set of numbers between (and including) the smallest value and the largest value.

20. What does interface testing requires?

Interface testing requires that we select the output values of the calling routine *i.e.* caller's range must be compatible with the called routine's domain.

An interface test consists of exploring the correctness of the following mappings

caller domain --> caller range (caller unit test)

caller range --> called domain (integration test)

Called domain --> called range (called unit test)