

GRAPH MATRICES AND APPLICATIONS

Problem with Pictorial Graphs

- Graphs were introduced as an abstraction of software structure.
- Whenever a graph is used as a model, sooner or later we trace paths through it- to find a set of covering paths, a set of values that will sensitize paths, the logic function that controls the flow, the processing time of the routine, the equations that define the domain, or whether a state is reachable or not.
- Path is not easy, and it's subject to error. You can miss a link here and there or cover some links twice.
- One solution to this problem is to represent the graph as a matrix and to use matrix operations equivalent to path tracing. These methods are more methodical and mechanical and don't depend on your ability to see a path they are more reliable.

Tool Building

- ☐ If you build test tools or want to know how they work, sooner or later you will be implementing or investigating analysis routines based on these methods.
- ☐ It is hard to build algorithms over visual graphs so the properties or graph matrices are fundamental to tool building.

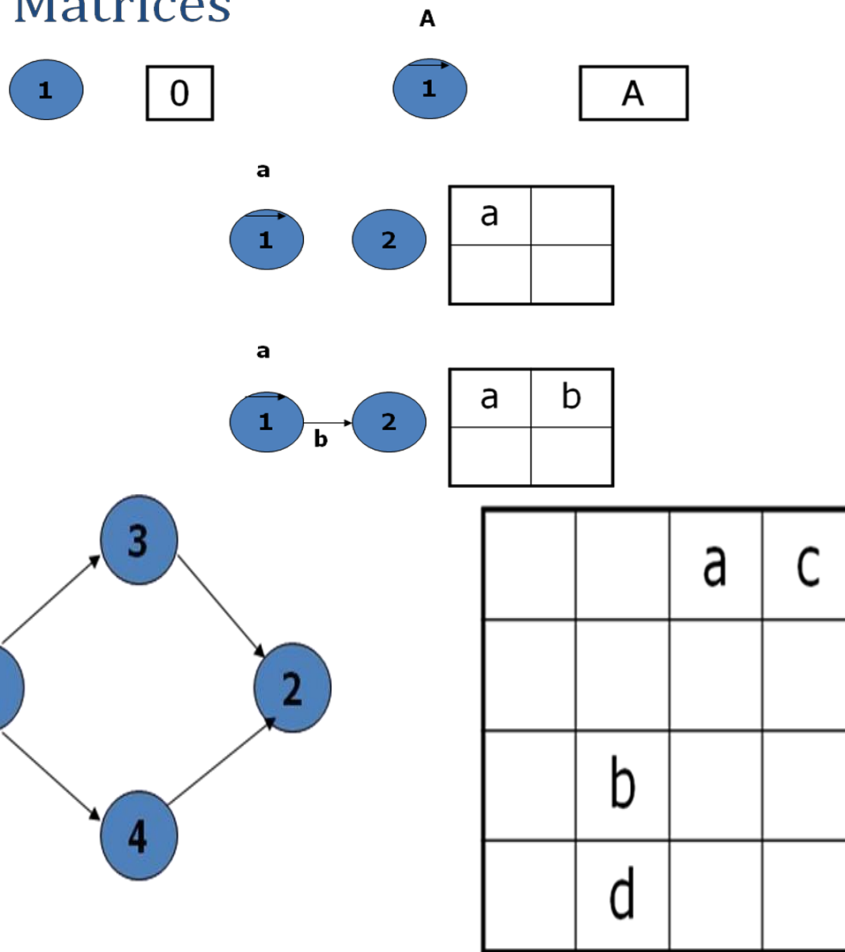
The Basic Algorithms

- ☐ The basic tool kit consists of:
- ☐ Matrix multiplication, which is used to get the path expression from every node to every other node.
- ☐ A partitioning algorithm for converting graphs with loops into loop free graphs or equivalence classes.
- ☐ A collapsing process which gets the path expression from any node to any other node.

The Matrix of a Graph

- A graph matrix is a square array with one row and one column for every node in the graph.
- Each row-column combination corresponds to a relation between the node corresponding to the row and the node corresponding to the column.
- The relation for example, could be as simple as the link name, if there is a link between the nodes.
- Some of the things to be observed:
 - The size of the matrix equals the number of nodes.
 - There is a place to put every possible direct connection or link between any and any other node.
 - The entry at a row and column intersection is the link weight of the link that connects the two nodes in that direction.
- ☐ A connection from node i to j does not imply a connection from node j to node i.
- If there are several links between two nodes, then the entry is a sum; the "+" sign denotes parallel links as usual.

Some Graphs and their Matrices



A simple weight

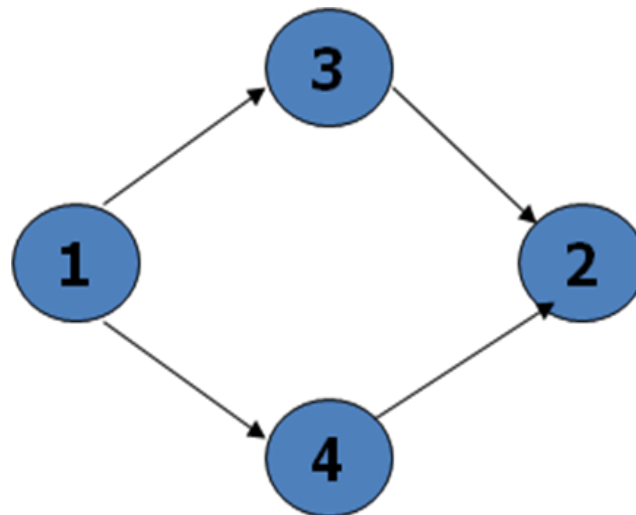
- A simplest weight we can use is to note that there is or isn't a connection. Let "1" mean that there is a connection and "0" mean that there isn't.
- The arithmetic rules are:
 - $1+1=1$ $1*1=1$
 - $1+0=1$ $1*0=0$
 - $0+0=0$ $0*0=0$
- A matrix defined like this is called connection matrix.

Connection matrix

- The connection matrix is obtained by replacing each entry with 1 if there is a link and 0 if there isn't.
- As usual we don't write down 0 entries to reduce the clutter.

		a	c
	b		
	d		

		1	1
	1		
	1		



Connection Matrix-continued

- ☐ Each row of a matrix denotes the out links of the node corresponding to that row.
- ☐ Each column denotes the in links corresponding to that node.
- ☐ A branch is a node with more than one nonzero entry in its row.
- ☐ A junction is node with more than one nonzero entry in its column.
- ☐ A self loop is an entry along the diagonal.

Cyclomatic Complexity

- The cyclomatic complexity obtained by subtracting 1 from the total number of entries in each row and ignoring rows with no entries, we obtain the equivalent number of decisions for each row. Adding these values and then adding 1 to the sum yields the graph's cyclomatic complexity.

		1	1	$2-1=1$
	1			$1-1=0$
	1			$1-1=0$

$1+1=2$ (cyclomatic complexity)

Relations

- ☐ A relation is a property that exists between two objects of interest.
- ☐ For example,
 - “Node a is connected to node b” or aRb where “R” means “is connected to”.
 - “ $a \geq b$ ” or aRb where “R” means greater than or equal”.
- ☐ A graph consists of set of abstract objects called nodes and a relation R between the nodes.
- ☐ If aRb , which is to say that a has the relation R to b, it is denoted by a link from a to b.
- ☐ For some relations we can associate properties called as link weights.

Transitive Relations

- ☐ A relation is transitive if aRb and bRc implies aRc .
- ☐ Most relations used in testing are transitive.
- ☐ Examples of transitive relations include: is connected to, is greater than or equal to, is less than or equal to, is a relative of, is faster than, is slower than, takes more time than, is a subset of, includes, shadows, is the boss of.
- ☐ Examples of intransitive relations include: is acquainted with, is a friend of, is a neighbor of, is lied to, has a du chain between.

Reflexive Relations

- ☐ A relation R is reflexive if, for every a, aRa .
- ☐ A reflexive relation is equivalent to a self loop at every node.
- ☐ Examples of reflexive relations include: equals, is acquainted with, is a relative of.
- ☐ Examples of irreflexive relations include: not equals, is a friend of, is on top of, is under.

Symmetric Relations

- ☐ A relation R is symmetric if for every a and b, aRb implies bRa .
- ☐ A symmetric relation mean that if there is a link from a to b then there is also a link from b to a.
- ☐ A graph whose relations are not symmetric are called directedgraph.

- ☐ A graph over a symmetric relation is called an undirected graph.
- ☐ The matrix of an undirected graph is symmetric ($a_{ij}=a_{ji}$) for all i,j)

Antisymmetric Relations

- ☐ A relation R is antisymmetric if for every a and b , if aRb and bRa , then $a=b$, or they are the same elements.
- ☐ Examples of antisymmetric relations: is greater than or equal to, is a subset of, time.
- ☐ Examples of nonantisymmetric relations: is connected to, can be reached from, is greater than, is a relative of, is a friend of

equivalence Relations

- ☐ An equivalence relation is a relation that satisfies the reflexive, transitive, and symmetric properties.
- ☐ Equality is the most familiar example of an equivalence relation.
- ☐ If a set of objects satisfy an equivalence relation, we say that they form an equivalence class over that relation.
- ☐ The importance of equivalence classes and relations is that any member of the equivalence class is, with respect to the relation, equivalent to any other member of that class.
- ☐ The idea behind partition testing strategies such as domain testing and path testing, is that we can partition the input space into equivalence classes.
- ☐ Testing any member of the equivalence class is as effective as testing them all.

Partial Ordering Relations

- ☐ A partial ordering relation satisfies the reflexive, transitive, and antisymmetric properties.
- ☐ Partial ordered graphs have several important properties: they are loop free, there is at least one maximum element, and there is at least one minimum element.

The Powers of a Matrix

- Each entry in the graph's matrix expresses a relation between the pair of nodes that corresponds to that entry.
- ☐ Squaring the matrix yields a new matrix that expresses the relation between each pair of nodes via one intermediate node under the assumption that the relation is transitive.
- ☐ The square of the matrix represents all path segments twolinks long.
- ☐ The third power represents all path segments three links long.

Matrix Powers and Products

- Given a matrix whose entries are a_{ij} , the square of that matrix is obtained by replacing every entry with
 - n
 - $a_{ij} = \sum_{k=1}^n a_{ik} a_{kj}$
 - $k=1$
- more generally, given two matrices A and B with entries a_{ik} and b_{kj} , respectively, their product is a new matrix C, whose entries are c_{ij} , where:
 - n
 - $C_{ij} = \sum_{k=1}^n a_{ik} b_{kj}$
 - $k=1$

Partitioning Algorithm

- Consider any graph over a transitive relation. The graph may have loops.
- We would like to partition the graph by grouping nodes in such a way that every loop is contained within one group or another.
- Such a graph is partially ordered.
- There are many used for an algorithm that does that:
- We might want to embed the loops within a subroutine so as to have a resulting graph which is loop free at the top level.
- Many graphs with loops are easy to analyze if you know where to break the loops.
- While you and I can recognize loops, it's much harder to program a tool to do it unless you have a solid algorithm on which to base the tool.

Node Reduction Algorithm (General)

- The matrix powers usually tell us more than we want to know about most graphs.
 - In the context of testing, we usually interested in establishing a relation between two nodes- typically the entry and exit nodes.
 - In a debugging context it is unlikely that we would want to know the path expression between every node and every other node.
 - The advantage of matrix reduction method is that it is more methodical than the graphical method called as node by node removal algorithm.
1. Select a node for removal; replace the node by equivalent links that bypass that node and add those links to the links they parallel.
 2. Combine the parallel terms and simplify as you can.
 3. Observe loop terms and adjust the out links of every node that had a self loop to account for the effect of the loop.
 4. The result is a matrix whose size has been reduced by 1. Continue until only the two nodes of interest exist.

Automation and testing tools:

Automation Testing is a software testing technique that performs using special automated testing software tools to execute a test case suite. On the contrary, Manual Testing is performed by a human sitting in front of a computer carefully executing the test steps.

The automation testing software can also enter test data into the System Under Test, compare expected and actual results and generate detailed test reports. Software Test Automation demands considerable investments of money and resources.

Successive development cycles will require execution of same test suite repeatedly. Using a test automation tool, it's possible to record this test suite and re-play it as required. Once the test suite is automated, no human intervention is required. This improved ROI (Return on Investment) of Test Automation. The goal of Automation is to reduce the number of test cases to be run manually and not to eliminate Manual Testing altogether.

Need for Automation:

Automation testing is the application of tools and technology to testing software with the goal of reducing testing efforts, delivering capability faster and more affordably. It helps in building better quality software with less effort.

By reducing the risk of human error in common daily tasks, automation gives you predictable and repeatable processes for managing configurations to improve consistency, speed changes and increase uptime.

The benefits of automation include improved accuracy, reliability, productivity as well as improved employee morale. The main objective of process automation is to improve a company's workflows. With automation, we can reduce costs, time, and waste as well as increase productivity, reduce mistakes, and control all the processes of the business in real time.

Automation brings in necessary agility to testing and helps it to respond faster and more effectively to changes. Continuous testing: Agility demands early and continuous testing. Test coverage extends to not only the newly added code but also the code from previous iterations.

Categorization of testing tools:

Software testing tools are required for the betterment of the application or software.

That's why we have so many tools available in the market where some are open-source and paid tools.

The significant difference between open-source and the paid tool is that the open-source tools have limited features, whereas paid tool or commercial tools have no limitation for the features. The selection of tools depends on the user's requirements, whether it is paid or free.

The software testing tools can be categorized, depending on the licensing (paid or commercial, open-source), technology usage, type of testing, and so on.

With the help of testing tools, we can improve our software performance, deliver a high-quality product, and reduce the duration of testing, which is spent on manual efforts.

The software testing tools can be divided into the following:

- Test management tool
- Bug tracking tool
- Automated testing tool
- Performance testing tool
- Cross-browser testing tool
- Integration testing tool
- Unit testing tool
- Mobile/android testing tool

- GUI testing tool
- Security testing tool

Test Management tool:

Test management tools are used to keep track of all the testing activity, fast data analysis, manage manual and automation test cases, various environments, and plan and maintain manual testing as well.

Bug tracking tool:

The defect tracking tool is used to keep track of the bug fixes and ensure the delivery of a quality product. This tool can help us to find the bugs in the testing stage so that we can get the defect-free data in the production server. With the help of these tools, the end-users can allow reporting the bugs and issues directly on their applications.

Automation testing tool:

This type of tool is used to enhance the productivity of the product and improve the accuracy. We can reduce the time and cost of the application by writing some test scripts in any programming language.

Performance testing tool

Performance or Load testing tools are used to check the load, stability, and scalability of the application. When n-number of the users using the application at the same time, and if the application gets crashed because of the immense load, to get through this type of issue, we need load testing tools.

Test Automation is the best way to increase the effectiveness, test coverage, and execution speed in software testing. Automated software testing is important due to the following reasons:

- Manual Testing of all workflows, all fields, all negative scenarios is time and money consuming.
- It is difficult to test for multilingual sites manually.
- Test Automation in software testing does not require Human intervention. You can run automated test unattended (overnight).
- Test Automation increases the speed of test execution.
- Automation helps increase Test Coverage.
- Manual Testing can become boring and hence error-prone.

Test cases to be automated can be selected using the following criterion to increase the automation ROI:

- High Risk – Business Critical test cases
- Test cases that are repeatedly executed
- Test Cases that are very tedious or difficult to perform manually
- Test Cases which are time-consuming

The following category of test cases are not suitable for automation:

- Test Cases that are newly designed and not executed manually at least once.
- Test Cases for which the requirements are frequently changing.
- Test cases which are executed on an ad-hoc basis.

Automated Testing Process:

Following steps are followed in an Automation Process

Step 1) Test Tool Selection

Step 2) Define scope of Automation

Step 3) Planning, Design and Development

Step 4) Test Execution

Step 5) Maintenance

Automation testing tool:

The automation testing is used to change the manual test cases into a test script with the help of some automation tools.

We have various types of automation testing tools available in the market. Some of the most commonly used automation testing tools are as follows:

- **Selenium**
- **Watir**
- **QTP**
- **Telerik Studio**
- **Testim**
- **Applitools**

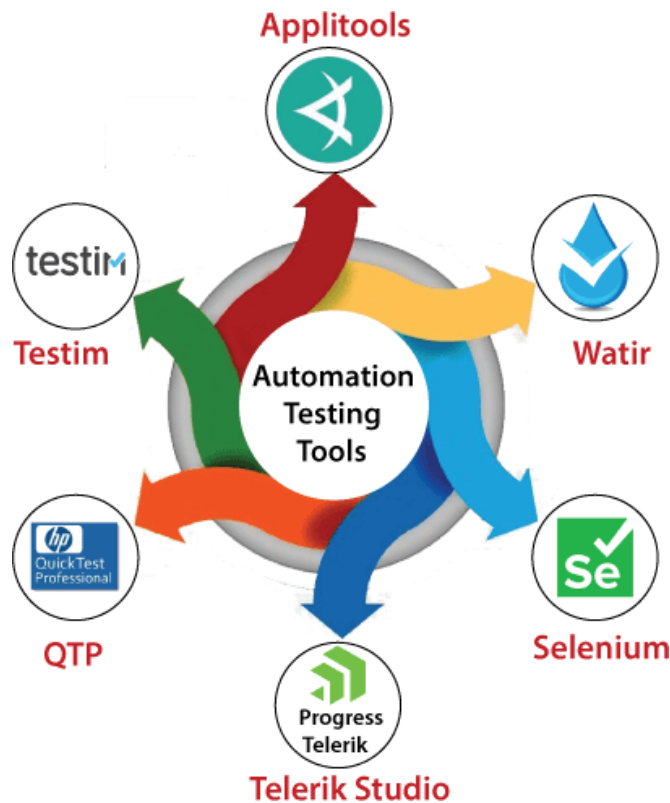


Fig. 5.1.Automation testing tools

Selenium:

It is an open-source and most commonly used tool in automation testing. This tool is used to test web-based applications with the help of test scripts, and these scripts can be written in any programming language such as java, python, C#, Php, and so on.

Features of Selenium:

This tool is most popular because of its various features. Following are standard features of Selenium:

- Selenium supports only web-based application, which means that the application can be opened by the browser or the URL like Gmail, Amazon, etc.
- Selenium does not support a stand-alone application, which means that the application is not opened in the browser or URL like Notepad, MS-Word, Calculator, etc.
- Selenium web driver is the latest tool in the selenium community that removes all the drawbacks of the previous tool (selenium-IDE).
- Selenium web-driver is powerful because it supports multiple programming languages, various browsers, and different operating systems and also supports mobile applications like iPhone, Android.

Performance testing tools (Load testing tools):

When we have to measure the load, stability, response time of the application, we required some performance (load) testing tools, which help us to test the performance of the software or an application. Performance testing tools can be open-source and commercial.

We have various types of performance testing tools available in the market; some of the most used performance (load) testing tools are as follows:

- **Apache JMeter**
- **LoadRunner[HP]**
- **LoadNinja**
- **WebLOAD**
- **LoadComplete**

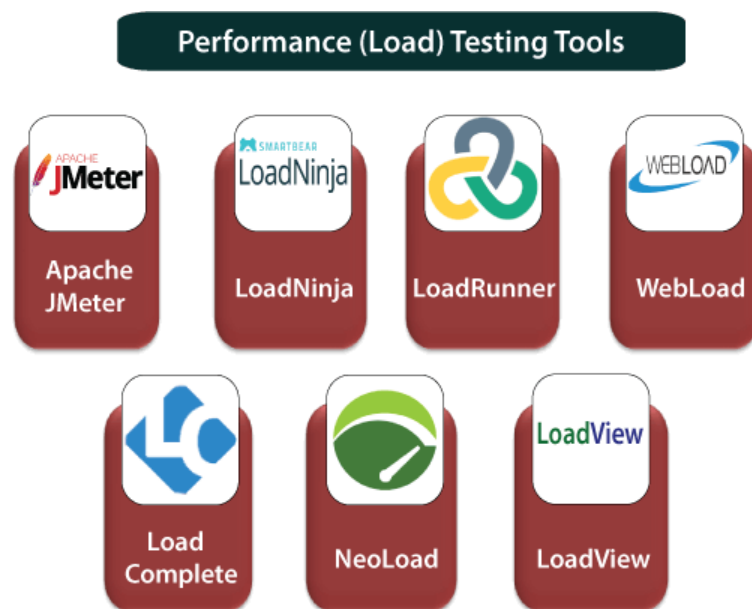


Fig. 5.2.Performance testing tools

LoadRunner:

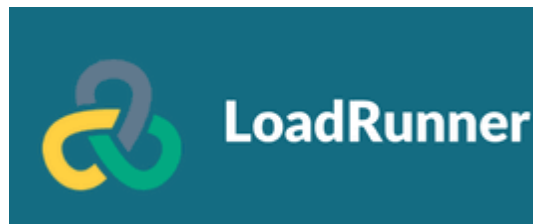
It is one of the most powerful tools of performance testing, which is used to support the performance testing for the extensive range of protocols, number of technologies, and application environments.

It quickly identifies the most common causes of performance issues. And also accurately predict the application scalability and capacity.

Feature of LoadRunner

- It will support XML; that's why we can easily view and handle the XML data within the test scripts.

- It supports a large range of applications, which will reduce the time to understand and explain the reports.
- With the help of this tool, we can get detailed performance test reports.
- It will reduce the cost of distributed load testing.
- It will provide the operational tool for deployment tracking.
- This tool is used to reduce the cost of software and hardware.



Apache JMeter:

It is used to test the performance of both static and dynamic resources and dynamic web applications. This tool is completely designed on the JAVA application to load the functional test behavior and measure the performance of the application. It is an open-source tool that facilitates users or developers to use the source code for the development of other applications.

It can be used to reproduce the huge load on a server, object, or network, group of servers to test its power, or to explore the complete performance in multiple load types. Previously it was used to test the web application, but now it is expanded to other test functions also.

Features of JMeter:

Below are some essential elements of JMeter:

- This tool supports a user-friendly GUI, which is interactive and straightforward.
- JMeter maintains multiple testing approaches, such as functional, distributed, and load testing.
- It is incredibly extensible to load the performance test in the multiple types of servers such as database server: LDAP, JMS, JDBC, web server: SOAP, HTTPS, HTTP, and mail server: POP3
- It is a platform-independent because it is designed with the help of JAVA, so it can run on any platform, which accepts a JVM like Window, Mac, and Linux, etc.

JUnit: Open Source Java Unit Testing Tool

JUnit is a unit testing framework for Java programming language. It plays a crucial role test-driven development, and is a family of unit testing frameworks collectively known as xUnit.

JUnit promotes the idea of "first testing then coding", which emphasizes on setting up the test data for a piece of code that can be tested first and then implemented. This approach is like "test a little, code a little, test a little, code a little." It increases the productivity of the programmer and the stability of program code, which in turn reduces the stress on the programmer and the time spent on debugging.

Features of JUnit:

- JUnit is an open source framework, which is used for writing and running tests.
- Provides annotations to identify test methods.
- Provides assertions for testing expected results.
- Provides test runners for running tests.
- JUnit tests allow you to write codes faster, which increases quality.
- JUnit is elegantly simple. It is less complex and takes less time.
- JUnit tests can be run automatically and they check their own results and provide immediate feedback. There's no need to manually comb through a report of test results.
- JUnit tests can be organized into test suites containing test cases and even other test suites.
- JUnit shows test progress in a bar that is green if the test is running smoothly, and it turns red when a test fails.