Software Testing Methodologies

BITS Pilani

Prashant Joshi

Module 2: Agenda

Module 2: Mathematics & Formal Methods

- **Topic 2.1** — Permutations & Combinations
- **Topic 2.2** — Propositional Logic
- **Topic 2.3** — Discrete Math
- **Topic 2.4** — Graph Theory

Math for Test Engineers

- For Test Engineers – Know our focus
- Testing is a craft; math are the craftsman's tools
- Bring *Rigor*, *Precision* and *Efficiency*
- Our treatment of math
  - Largely informal – What is required for Test Engineers and not for mathematicians
  - Our focus is discrete mathematics
- Aim is
  - To make test engineers better at their craft

BITS Pilani
Pilani Campus

Topic 2.1: Permutations & Combinations

Permutation & Combination

- Selecting several things out of a larger group

- Two aspects to look at
  - Order
  - Repetition

Combination

- Order does not matter
- Example: Fruits in a fruit salad. It does not matter in which the fruits are put into the salad. It could be Apple, Banana and Strawberry or any other order

Permutation

- Order does matter
- Example: A lock which opens with a sequence of digits. We call it the combination lock. It is indeed a permutation lock
    - Sequence 437 (347 will never work!)

Indeed a permutation lock

Permutation

- Repetition – Yes & No
- Example
  - Repetition allowed: Digits in the permutation lock may repeat like “333” or “557”
  - Repetition now allowed: First three standings in a running race

Permutation

- Choosing r things out of n

Repetition

- n possibilities for each of the r choices

$$n^r$$

^nC_r

Without Repetition

- Possibilities reduce with every selection

$$\frac{n!}{(n-r)!}$$

Combination

Choosing r things out of n

Repetition allowed

$$\frac{(n+r-1)!}{r!\,(n-1)!}$$

Without Repetition

$$\frac{n!}{r!\,(n-r)!}$$

Software Testing Methodologies

BITS Pilani

Prashant Joshi

innovate achieve lead

BITS Pilani
Pilani Campus

Topic 2.2: Propositional Logic

Propositional Logic

- A proposition is a sentence that is either TRUE or FALSE
- Given a proposition, it is always possible to tell if it is T or F
- Propositional logic has operations, expressions, and identities
- Logical Operators
- Logical Expressions
- Logical Equivalence

Propositional Logic

p	q	p^q (AND)	pvq (OR)	~p (NOT)
T	T	T	T	F
T	F	F	T	F
F	T	F	T	T
F	F	F	F	T

p	q	p+q (EX-OR)	p→q (IF-THEN)
T	T	F	T
T	F	T	F
F	T	T	T
F	F	F	T

BITS Pilani

Software Testing Methodologies

Prashant Joshi

innovate achieve lead

BITS Pilani
Pilani Campus

Topic 2.3: Discrete Math

Propositional Logic

- A proposition is a sentence that is either TRUE or FALSE
- Given a proposition, it is always possible to tell if it is T or F
- Propositional logic has operations, expressions, and identities
- Logical Operators
- Logical Expressions
- Logical Equivalence

Propositional Logic

p	q	p^q (AND)	pvq (OR)	~p (NOT)
T	T	T	T	F
T	F	F	T	F
F	T	F	T	T
F	F	F	F	T

p	q	p+q (EX-OR)	p→q (IF-THEN)
T	T	F	T
T	F	T	F
F	T	T	T
F	F	F	T

innovate achieve lead

BITS Pilani
Pilani Campus

Topic 2.3: Discrete Math

Set Theory

Collection of things which have a common property

- Things that one wears (Specific activity wear)
- Sports kit for badminton
- Months in a year or months with 31 days

Listing Elements

$$Y = \{April, June, September, November\}$$

Decision rule

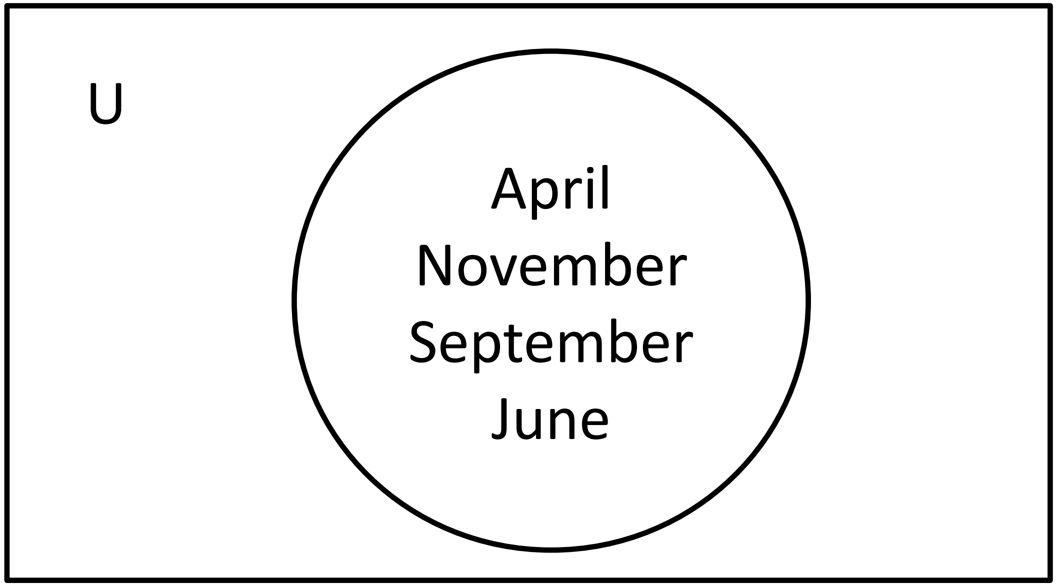
$$Y = \{year: 1800 \leq year \leq 2014\}$$

Decision rule

$$S = \{sales: the\ 15\%\ commission\ rate\ applies\ to\ the\ sale\}$$

Venn Diagrams

- Picture(s) for Sets
- A set of depicted as a circle with interior of the circle corresponds to the elements of the set



Venn diagram of 30 day month

Set Operations

- Union is the set $A \cup B = \{x : x \in A \vee x \in B\}$
- Intersection is the set $A \cap B = \{x : x \in A \wedge x \in B\}$
- Complement of A is the set $A' = \{x : x \notin A\}$
- Relative complement of B WRT A is the set

  $A - B = \{x : x \in A \wedge x \notin B\}$
- Symmetric difference of A and B is the set

  $A \oplus B = \{x : x \in A \oplus x \in B\}$

Refer to Venn Diagrams of basic sets in T1

Set Operations

- Unordered pair (a, b)
- Ordered pair $< a, b >$

What is the difference?

- Cartesian Product

$$A \times B = \{< x, y >: x \in A \wedge y \in B\}$$

Set Relations

- Subset
  - A is subset of B if and only if all elements of A are also in B
- Proper subset
  - A is a proper subset of B if and only if there is at least one element in B which is not in A
- Equal Sets
  - Each is a subset of the other

Look up the notations in the book T1 Chapter 3

Set Partitions

$$n^r$$

Software Testing Methodologies

BITS Pilani

Prashant Joshi

Topic 2.4: Graph Theory

Graph

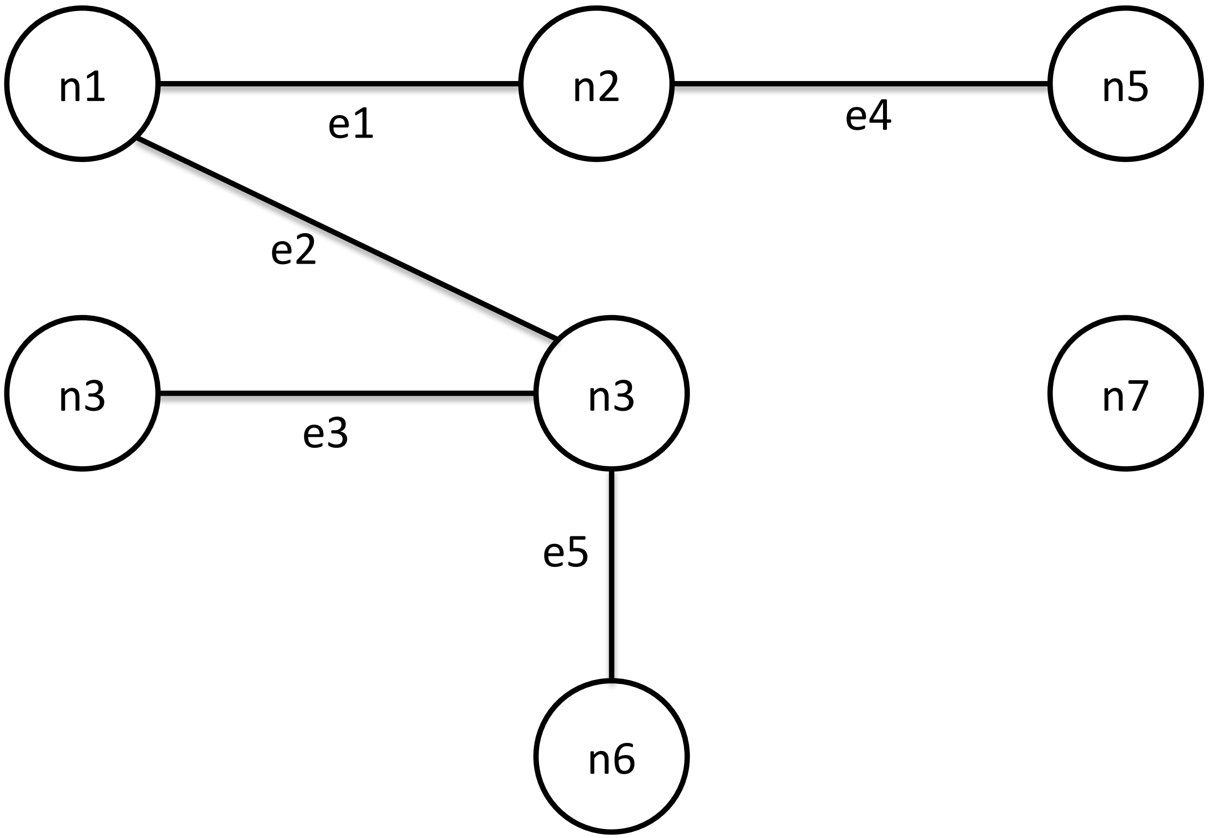
- A graph (also known as linear Graph) is an abstract mathematical structure defined from two sets – set of nodes and set of edges that form connections between nodes
- Example: Computer Network
- Definition
- *A Graph G = (V,E) is composed of a finite (and nonempty) set V of nodes and a set of E of unordered pairs of nodes*

$$V = \{n_1, n_2, n_3 ..., n_m)$$

$$E = \{e_1, e_2, e_3 ..., e_p)$$

A Graph

- Nodes and Edges Sets
- Connection between nodes

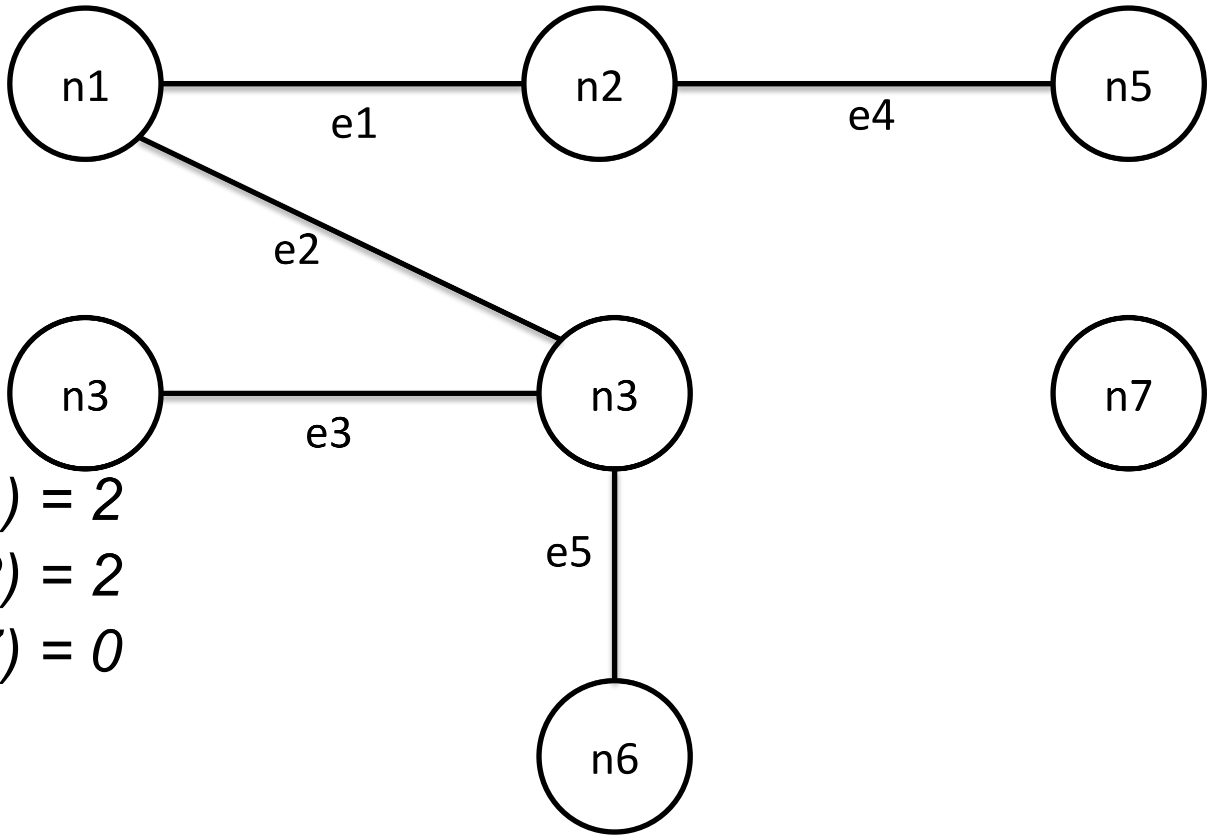


Use of representation

- Nodes as program statements
- Edges
    - Flow of control
    - Define/use relationships

Degree of a Node

- The degree of a node in a graph is the number of edges that have that node as an endpoint *deg(n)*



deg(n1) = 2
deg(n2) = 2
deg(n7) = 0

Use of Degree of Node

- Indicates Popularity
- Social scientists
    - Social interactions
    - Friendship/communicates with
- Example:
    - Graph with nodes are objects and edges are messages; degree can represent the extent of integration testing that is appropriate for the object

Incidence Matrix

- The incidence matrix is a graph G=(V,E) with m nodes and n edges is a m x n matrix, where the element in row i, column j is a 1 if and only if node i is an endpoint of edge j; otherwise the element is 0

	e1	***e2***	***e3***	***e4***	***e5***
n1	1	1	0	0	0
n2	1	0	0	1	0
n3	0	0	1	0	0
n4	0	1	1	0	1
n5	0	0	0	1	0
n6	0	0	0	0	1
n7	0	0	0	0	0

Use of this representation

- Degree of node is zero
- Unreachable node

Adjacency Matrix

- Deals with connections
- The adjacency matrix of a Graph G=(V,E) with m nodes is an m x m matrix, where the element in row i, column j is 1 if and only if an edge exists between node i and node j; otherwise, the element is 0

	n1	***n2***	***n3***	***n4***	***n5***	***N6***	***n7***
n1	0	1	0	1	0	0	0
n2	1	0	0	0	1	0	0
n3	0	0	0	1	0	0	0
n4	1	0	1	0	0	1	0
n5	0	1	0	0	0	0	0
n6	0	0	0	1	0	0	0
n7	0	0	0	0	0	0	0

Use of this representation

- Deals with connections
- Useful for later graph theory concepts example: paths

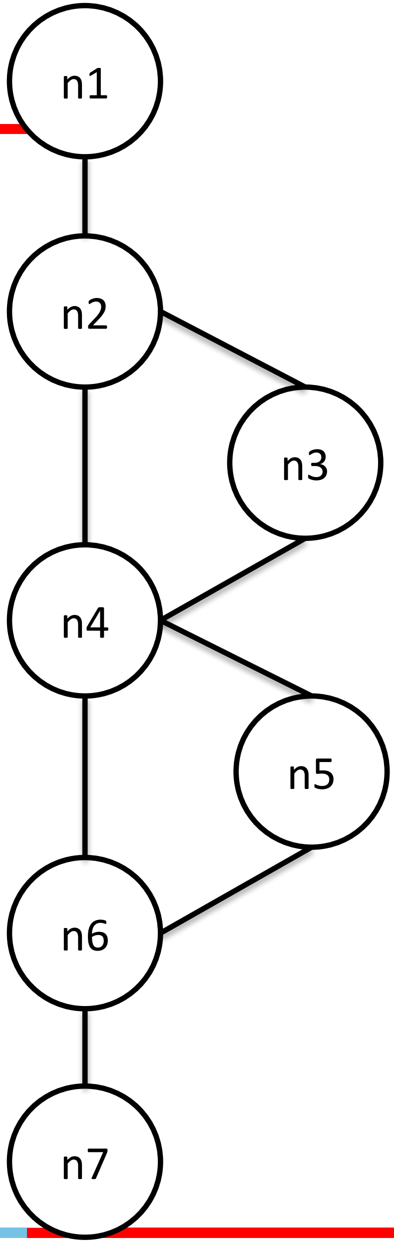
Paths

- A path is a sequence of edges such that for any adjacent pair of edges e_i, e_j in the sequence, the edges share a common (node) endpoint

Path	Node Sequence	Edge Sequence
Between n1 and n5	n1, n2, n5	e1, e4
Between n6 and n5	n6, n4, n1, n2, n5	e5, e2, e1, e4
Between n3 and n2	n3, n4, n1, n2	e3, e2, e1

Graph

- n1 represents a series of statements
- n7 also represents a series of statement
- Is this the correct representation?
- How does this help us get to testing?
- What does this help in?



BITS Pilani

Software Testing Methodologies

Prashant Joshi