SNS COLLEGE OF ENGINEERING

Kurumbapalayam(Po), Coimbatore – 641 107

Accredited by NAAC-UGC with 'A' Grade

Approved by AICTE, Recognized by UGC & Affiliated to Anna University, Chennai

Department of Artificial Intelligence and Data Science

23ITT203 Object Oriented Software Engineering

Model Checking

What is Model Checking?

Definition:

- Model checking is an automated technique used to check if a software system behaves correctly by exploring all possible states the system can be in.
- It answers the question: "Does my software always behave correctly, no matter what inputs or situations occur?"

How Model Checking Works

- **Create a model of your software**

(e.g., a flowchart, a diagram, or a state machine that represents how your system behaves)

- **Define properties or rules to check**

(e.g., “The system should never crash”, or “The door should never open while the elevator is moving”)

- **Use a model checker**

(a software tool that automatically checks all possible behaviors against those rules)

- **If a rule is violated, the model checker shows a counterexample (an error path).**

Real-Life Example: Traffic Light System

Scenario:

You are building software for a **traffic light** that cycles like this:

- Red → Green → Yellow → Red...

You want to check:

- The light **never turns green right after yellow**
- The light **never skips a color**

How Model Checking Applies

Model: You model the traffic light as a **state machine**:

States: Red, Green, Yellow

Transitions: Red → Green, Green → Yellow, Yellow → Red

Rules to check:

It must always go Red → Green → Yellow → Red

It must never go Red → Yellow (skip green)

Model Checker: The tool checks **all possible sequences** of state changes, and if it finds any incorrect path (like Red → Yellow), it shows that as a problem.

In Object-Oriented Software Engineering

In OOP, systems have:

Classes and **objects**

State changes (when an object variable changes)

Interactions between objects (e.g., user login, transaction, notification)

Model checking helps by:

Modeling object behaviors as **state machines**

Checking **if methods and object states follow correct rules**

Verifying **safety properties** like “user must be logged in before accessing data”

Benefits of Model Checking

Benefit	Explanation
✔ Catches bugs early	Especially those that occur in rare conditions
✔ Automatic	You don't have to test every input manually
✔ Gives counterexamples	Shows exactly how the error happens
✔ Works well for critical systems	Banking, aviation, medical software

Limitations of Model Checking

Limitation	Description
! State explosion	Too many states can make it slow or impossible
! Modeling is required	You must first build an accurate model
! Doesn’t check performance	Only checks logical correctness, not speed or UI


thank
you