

# 9 Lecture - CS402

## Important Subjective

### 1. What is the difference between a GTG and a finite state machine?

Answer: GTGs can capture more complex system interactions than finite state machines. While finite state machines are limited to simple systems with a fixed number of states and transitions, GTGs can represent systems with multiple components and more complex interactions between them.

### How can GTGs be used for software design?

Answer: GTGs can be used to model the behavior of a system, which can help in designing the software that implements the system. GTGs can also be used to identify potential errors or edge cases in the system design.

### What is a state in a GTG?

Answer: A state in a GTG represents a particular configuration or condition of the system being modeled. For example, in a vending machine, a state might represent the machine being idle, dispensing a product, or out of change.

### What is a transition in a GTG?

Answer: A transition in a GTG represents a change in the system from one state to another. For example, in a vending machine, a transition might represent the machine dispensing a product or returning change to the user.

### How can GTGs be used for software testing?

Answer: GTGs can be used to generate test cases that cover all possible system states and transitions. By analyzing the behavior of the system using the GTG, testers can identify all possible scenarios that need to be tested.

### How can GTGs be used for requirements analysis?

Answer: GTGs can be used to capture the desired behavior of the system and ensure that all requirements are met. By analyzing the behavior of the system using the GTG, requirements can be refined and validated.

### What are some limitations of using GTGs?

Answer: GTGs can be difficult to create, especially for large or complex systems. Additionally, GTGs may not be suitable for modeling very simple systems or systems with highly variable behavior.

### What is the difference between a GTG and a UML state machine diagram?

Answer: UML state machine diagrams are a graphical modeling language used to represent the behavior of an object or system. While they are similar to GTGs in that they represent states and transitions, UML state machine diagrams are more formal and are often used in object-oriented design.

### How can GTGs be used to identify errors or edge cases in a system?

Answer: By analyzing the behavior of the system using the GTG, potential errors or edge cases

can be identified. For example, if a particular state or transition is not covered by the GTG, it may indicate that the system is not handling that scenario correctly.

**What is the benefit of using a graphical representation, such as a GTG, for modeling a system?**

Answer: A graphical representation makes it easier to understand and communicate the behavior of a system. By visually representing the states and transitions, it is easier to identify potential errors, edge cases, or areas where the system can be improved.