Seminar talk outline

00:00
“Modern software systems are becoming increasingly complex. With their growing size, testing is becoming both time consuming ad requires significant effort. Even automated test generation tools let some bugs pass through. We need a way that takes all the programming language constructs away from testing the functionality of the systems.”

"This is where software verification comes into picture.”

01:00
Explanation of Peterson’s algorithm.

01:45
Note on the properties specified by the algorithm as well as the assumptions we have made regarding the conditions in which these conditions should hold.

3:00
What is model checking and what is deduction.

4:00
“Both Model checking and Deduction requires a specification of the system against which the clauses are checked. So let’s begin by writing a specification for Peterson’s algorithm.”

5:00
Notes on how the specification is written and what the different components are.

10:00
How does the tla+ model checker work.

Presenting the results from model checking Petersons algorithm.

11:00
Present an argument on why model checking might fail in some cases. And present a case for Proof system.

12:00
Talking about the architecture of the proof system.

12:30
Setting the outlook for the proof of Mutual exclusion.

14:00
Explaining how proof system works by expanding the init clause and proving invariant true.

16:30
Note on how the invariant proof flows in the intermediate steps.

18:00
Introduce the use of temporal logic (PTL) ad briefly present the operators.

19:00
A side on how the proof looks in the toolbox.

20:00
Conclude by presenting Pros and cons of using tla+.