Internship Report – Month 4

Title: Introduction to UPPAAL-SMC and Probabilistic Simulation
Software Used: UPPAAL 5.0.0 (with SMC support)

General Objective for Month 4:

The focus of this month was on learning to use **UPPAAL-SMC** for modeling and simulating **real-time systems with probabilistic and stochastic behaviors**. A practical traffic light system with multiple cars was modeled and analyzed using probabilistic constructs.

Weekly Training Summary (based on official schedule):

Week 1:

Introduction to UPPAAL-SMC and probabilistic modeling

- Introduction to the differences between UPPAAL and UPPAAL-SMC
- Learning about random distributions (e.g., exponential)
- Using stochastic clocks and `rate()` expressions
- Modeling failure and recovery states in traffic lights

Week 2:

Simulating probabilistic systems using UPPAAL-SMC

- Designing templates for CarA and CarB
- Defining a centralized traffic light controller
- Synchronizing processes using broadcast channels (`greenA!`, `carA_arrival[id]?`)
- Scaling the model using parameter `id` to represent multiple instances

Week 3:

Analyzing stochastic and non-deterministic behaviors

- Implementing transitions for failure (`Broken`) and self-repair (`Recovery`)
- Adding probabilistic behavior using `rate()` on transitions
- Queue comparison logic to prioritize one direction
- Designing guards and updates to avoid deadlocks

Week 4:

Advanced practical exercises using UPPAAL-SMC

- Final model implementation including:
  - Two cars on route A and two cars on route B
  - Intelligent traffic light with dynamic decision logic
  - Stochastic transitions representing real-world timing uncertainties

Running simulations and analyzing results using SMC queries like:

smc
CopyEdit

```
Pr[<=60](<> CarA(0).Pass)
E[<=100](max: queueB)
Pr[<=50](<> light.Broken)
```

Skills and Competencies Gained:

- Mastery of modeling real-world systems with probabilistic behavior
- Understanding behavior under faults, competition, and nondeterminism
- Proficient use of UPPAAL-SMC syntax, including:

    - `rate()`, `broadcast chan[]`, `bool[]`, `int id`, and system instantiation
- Ability to formulate, execute, and interpret advanced SMC queries

Technical Challenges Faced:

- Errors caused by incorrect use of parameterized processes and synchronizations
- Deadlock risks due to missing or improperly defined guards
- Syntax issues such as "unexpected T-ASSIGNMENT" or "No such process" due to declaration mismatches

Conclusion:

By the end of Month 4, I had developed a complete, functional, and analyzable model for a probabilistic traffic light system. This phase significantly strengthened my ability to use UPPAAL-SMC for complex, non-deterministic systems and laid a solid foundation for upcoming phases involving autonomous vehicles and real-world applications