Report for Month 2: Simulating and Analyzing Systems with UPPAAL

Introduction

The focus of Month 2 was on **simulating and analyzing concurrent systems** using UPPAAL. After the basic introduction in Month 1, this stage was about exploring advanced simulation tools, verifying system properties, and modeling multi-agent systems with synchronization.

Week 1: Exploring Advanced Simulation Methods

- Learned to use the **Symbolic Simulator** and the **Concrete Simulator**.

- Difference between them:

  - Symbolic Simulator → explores symbolic paths with timing constraints.

  - Concrete Simulator → runs specific traces with fixed timing values.

- Practiced with the simple models from Month 1 to observe possible system behaviors.

Week 2: Analyzing System Properties

- Introduced to **CTL (Computation Tree Logic)** properties in UPPAAL.

- Checked different types of properties:

  - **Reachability** (e.g., `E<> Car.Crossing`).

  - **Safety** (e.g., `A[] not deadlock`).

  - **Liveness** (e.g., `A<> Car.Exited`).

- Verified that the system was **safe, deadlock-free, and guaranteed to progress**.

Week 3: Practical Exercises for Concurrent Systems

- Modeled **two cars approaching an intersection**.

- Each car had states: *Idle*, *Approaching*, *Crossing*, *Exited*.

- Used **synchronization channels** (`car_here!`, `green?`) to coordinate between cars and the traffic light.

- This taught us how to model **multi-agent concurrent systems** in UPPAAL.

Week 4: Analyzing Complex Concurrent Systems

- Extended the models to handle more complex situations:

○ Multiple cars are arriving simultaneously.

○ Shared global variables, such as a `queue` to count waiting cars.

○ Advanced queries such as:

■ `E<> (CarA.Crossing && CarB.Crossing)` → both cars crossing simultaneously.

■ `A[] (CarA.Crossing imply not CarB.Crossing)` → ensuring no collision.

- Ensured that the system was **safe, fair, and deadlock-free**.

Conclusion

By the end of Month 2, we successfully:

1.   Gained hands-on experience with **UPPAAL simulators**.

2.   Verified **safety, liveness, and deadlock-freedom** using CTL queries.

3.   Modeled concurrent systems with multiple agents and synchronization.

4.   Built the foundation for more advanced **probabilistic simulations** in Month 4 with UPPAAL-SMC.