CENG 383

Real-Time Systems

Lecture 5

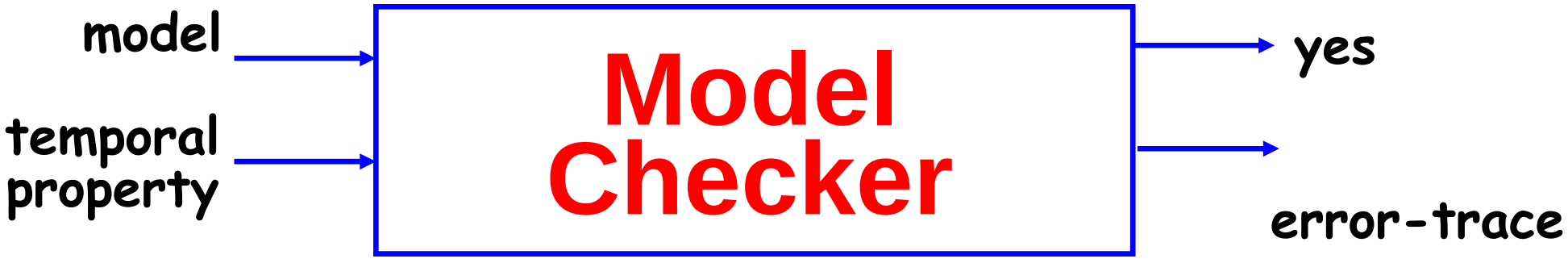
Formal Methods II

Model Checking

Asst. Prof. Tolga Ayav, Ph.D.

Department of Computer Engineering

İzmir Institute of Technology



Advantages

Automated formal verification, Effective debugging tool

Moderate industrial success

In-house groups: Intel, Microsoft, Lucent, Motorola…

Commercial model checkers: FormalCheck by Cadence

Obstacles

Scalability is still a problem (about 500 state vars)
Effective use requires great expertise

Still, a great success story for CS theory impacting practice, and a vibrant area of research

UPPAAL: www.uppaal.com

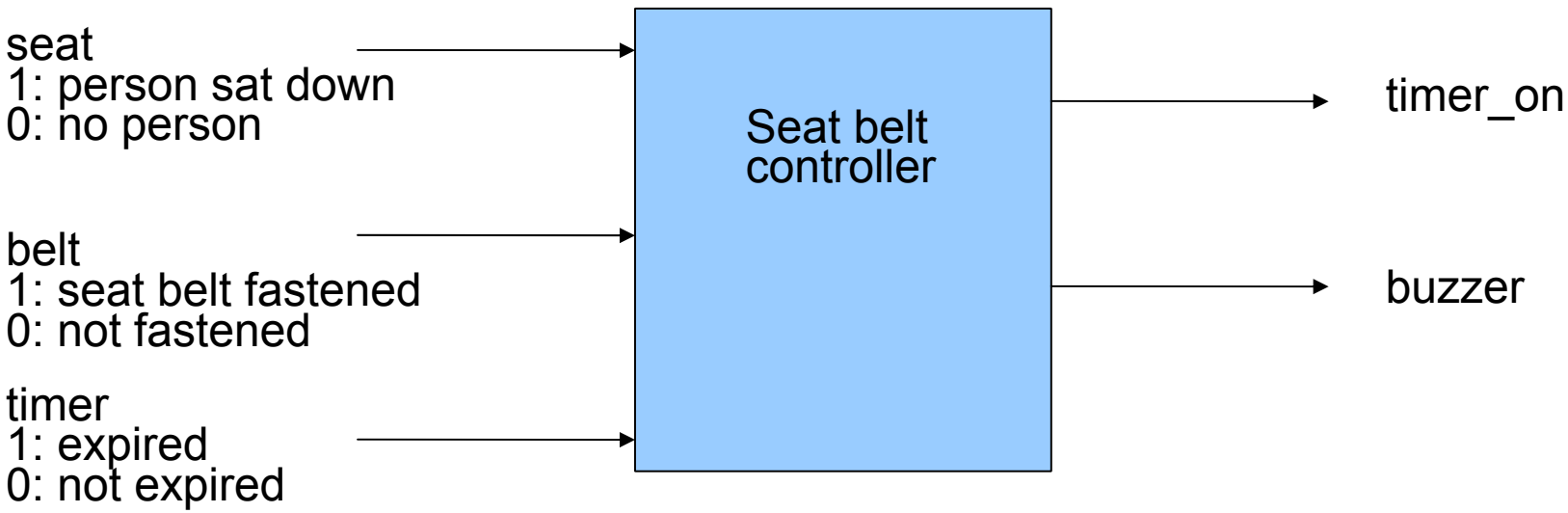
- developed jointly by Uppsala university and Aalorg university

- UPPsala + AALborg = UPPAAL
  - SWEDEN + DENMARK = SWEDEN
  - SWEDEN + DENMARK = DENMARK

State Machine Example

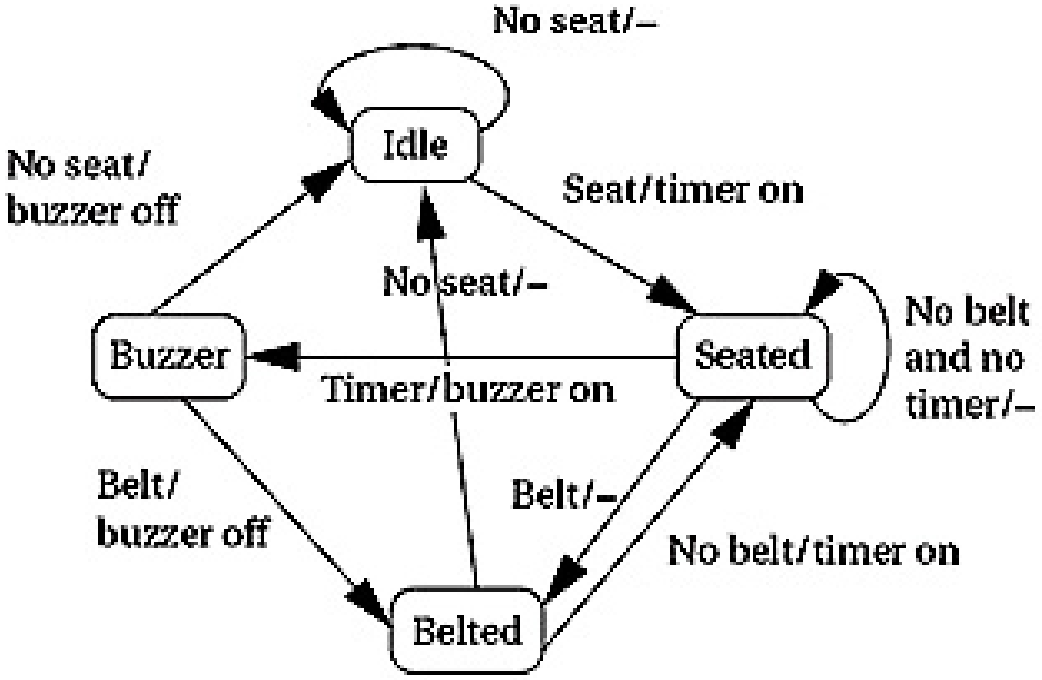
Design a Simple Seat Belt Controller:

The controller's job is to turn on a buzzer if a person sits in a seat and does not fasten the seat belt within a fixed amount of time. This system has three inputs and one output.
The inputs are a sensor for the seat to know when a person has sat down, a seat belt sensor that tells when the belt is fastened, and a timer that goes off when the required time interval has elapsed. The output is the buzzer

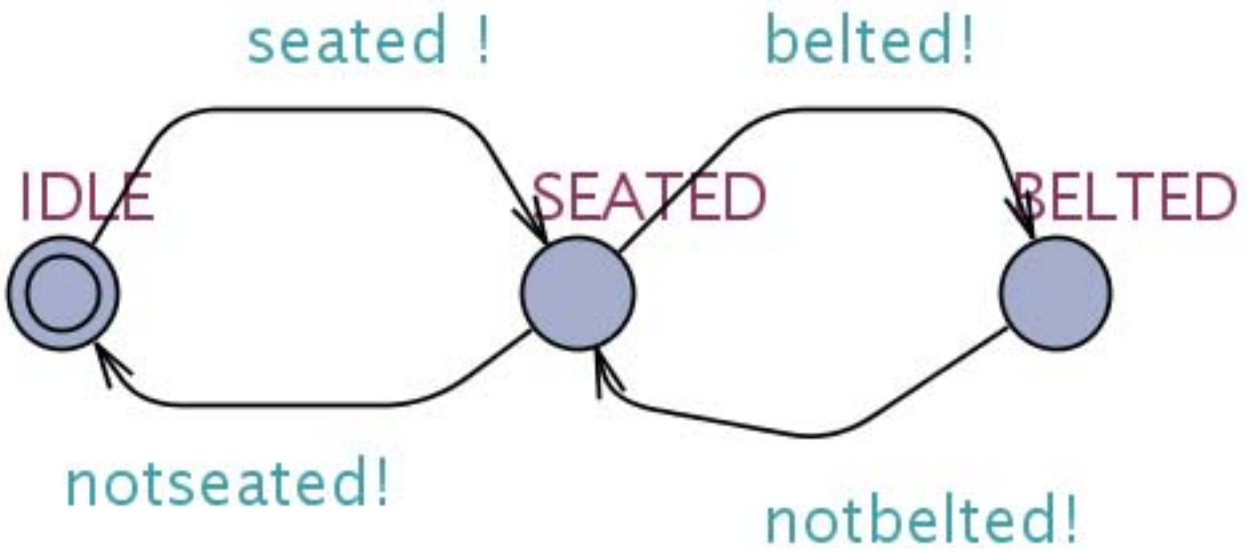


FSM for the Seat Belt Controller

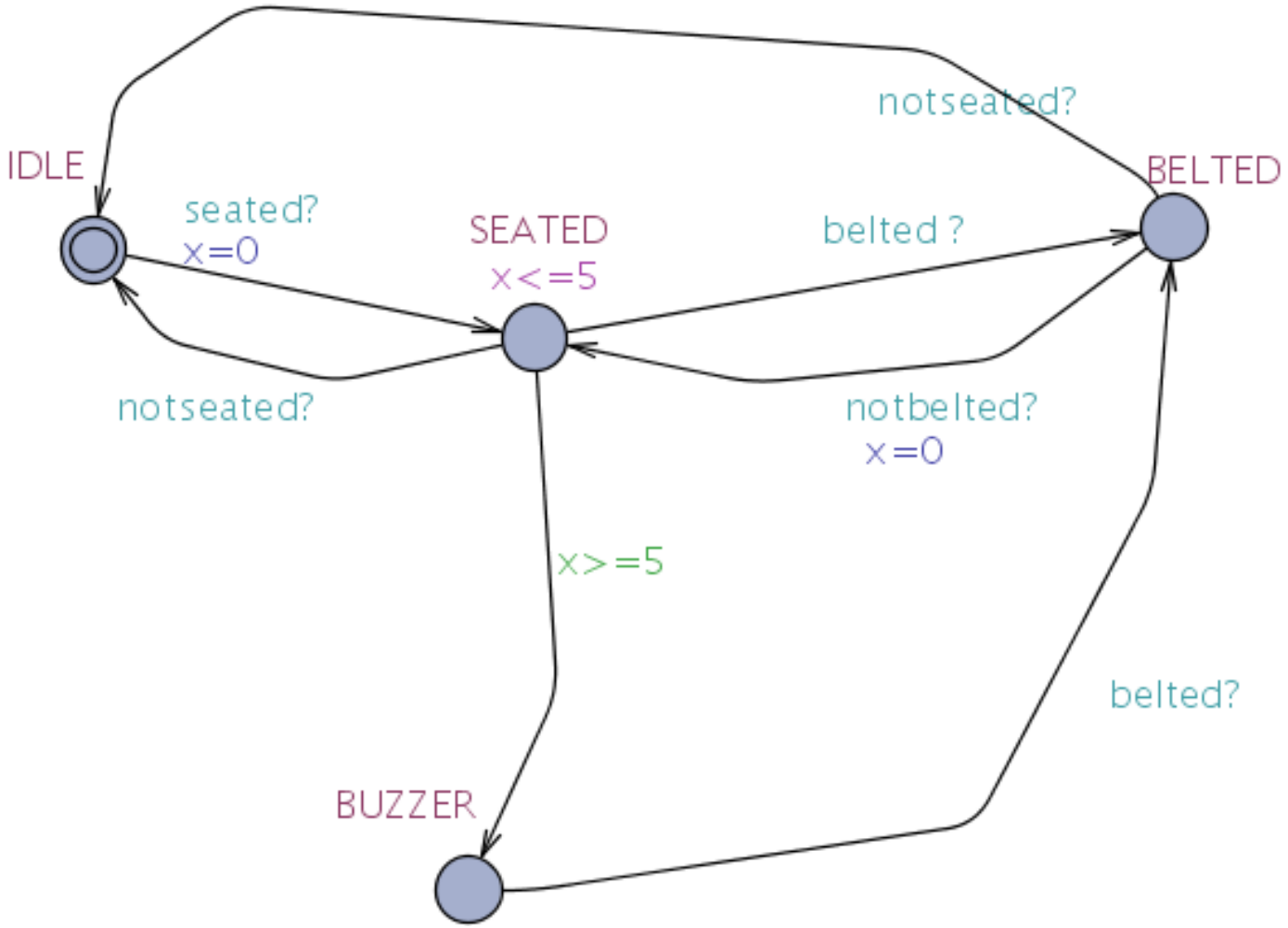
Inputs/outputs
(– = no action)



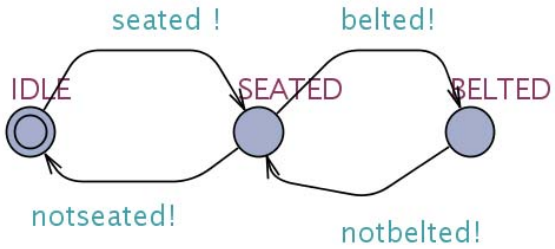
Timed Automaton for Driver



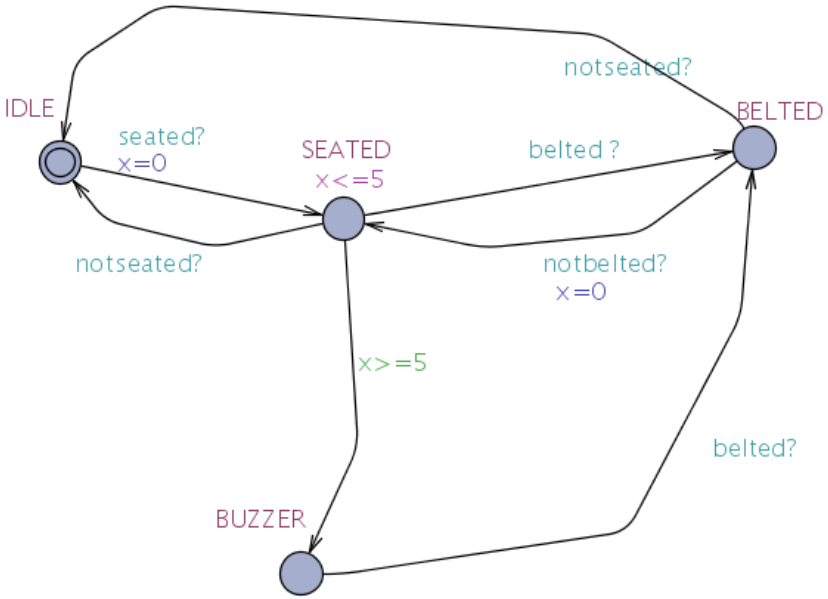
Timed Automaton of Car's Seatbelt Controller



Property Verification



||



We can check if TA1 || TA2 satisfy the following CTL properties:

A[] not (car.SEATED && x>5)

A[]not deadlock

(driver.SEATED && car.SEATED && x>5)-->(driver.SEATED && car.BUZZER)

E<>car.BUZZER

A[](driver.SEATED && (car.SEATED || car.BUZZER))

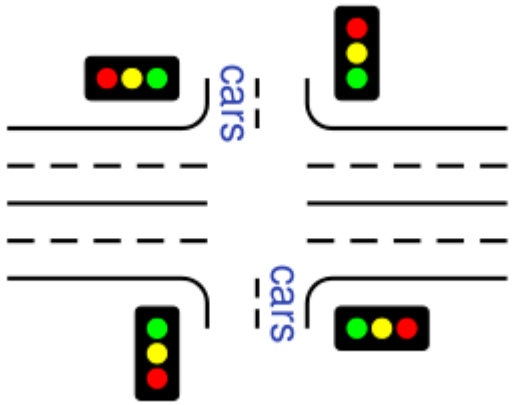
|| (car.IDLE&&driver.IDLE) || (car.BELTED&&driver.BELTED)

-

A[](driver.SEATED && ((car.SEATED&&x<=5) || (car.BUZZER&&x>=5)))

|| (car.IDLE&&driver.IDLE) || (car.BELTED&&driver.BELTED)

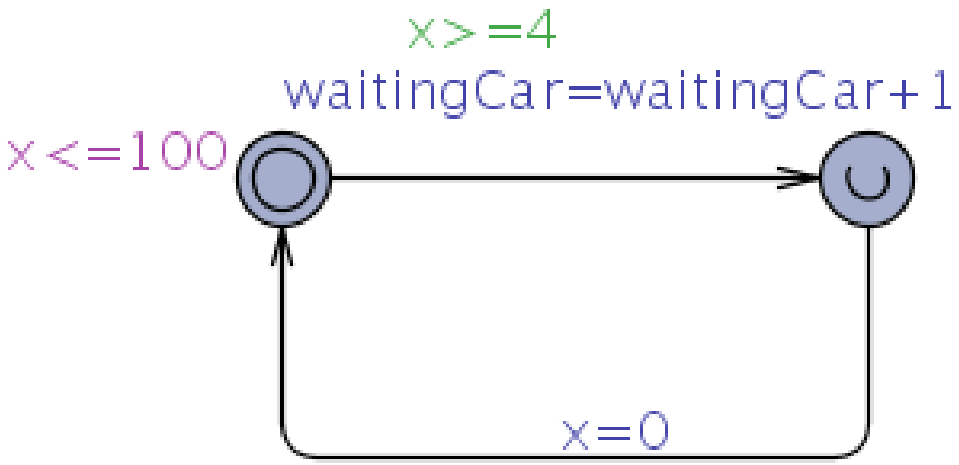
Case Study: Traffic Light Controller



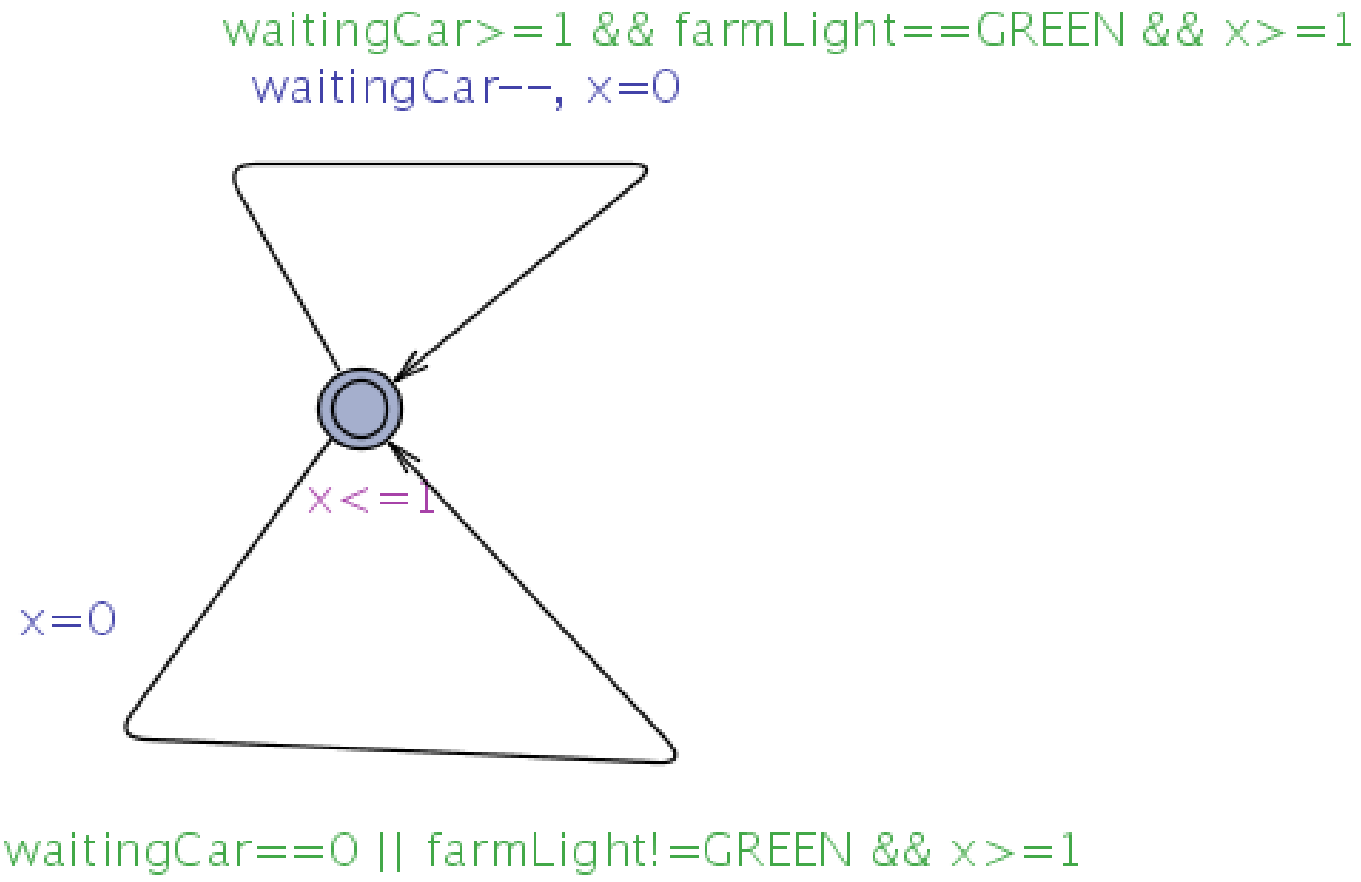
Traffic light controller controls a traffic light at the intersection of a busy highway and a farm road. Normally, the highway light is green but if a sensor detects a car on the farm cars road, the highway light turns yellow then red. The farm road light then turns green until there are no cars or after a long timeout. Then, the farm road light turns yellow then red, and the highway light returns to green. The inputs to the machine are the car sensor, a short timeout signal, and a long timeout signal. The outputs are a timer start signal and the colors of the highway and farm road lights.

Create a Timed Automata to model this intersection.
Draw your timed automata in UPPAAL.
Verify your model.

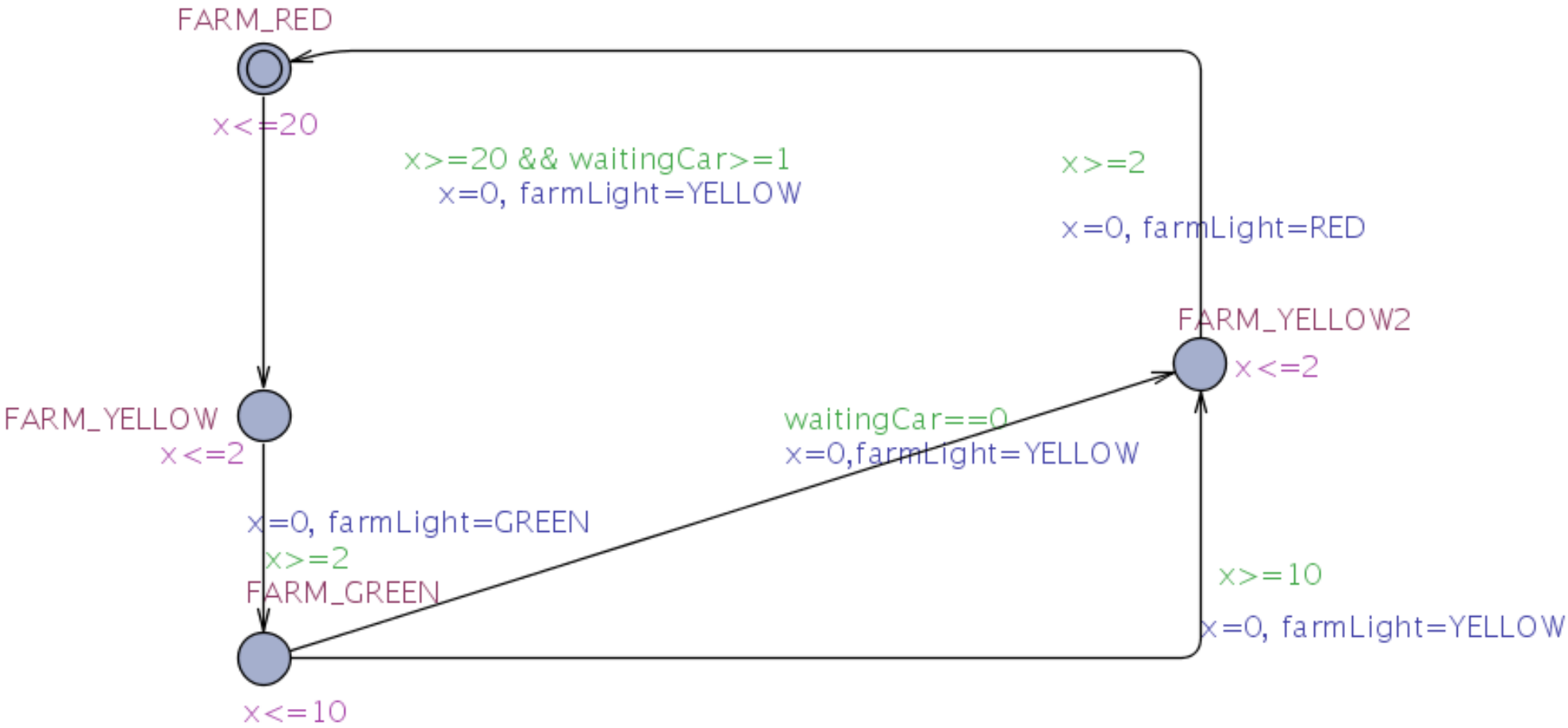
Automaton for Car Arrival at Farm Road



Automaton for Car Passing at Farm Road



Automaton for Traffic Lights at Farm Road



Simulation and Verification

- Simulate your model in UPPAAL model checker
- Write down the necessary properties.
- Verify the model using UPPAAL verification tool.
- Using verification, find an answer to the following question:
  - What is the maximum number of waiting cars on farm road?