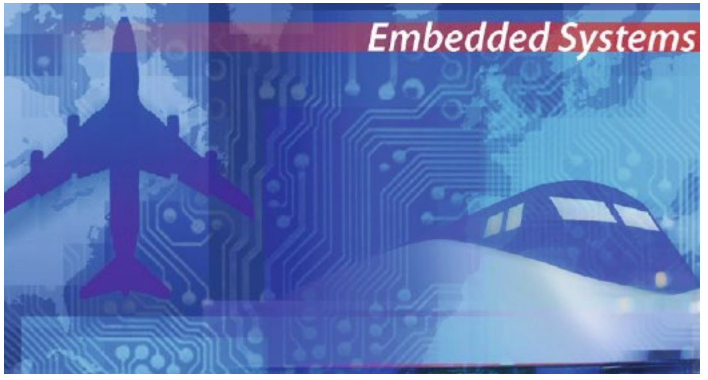
Sharif University of Technology
Department of Computer Science and Engineering

Lec. 2:

Automata-Based Programming: An Example



M. Ansari

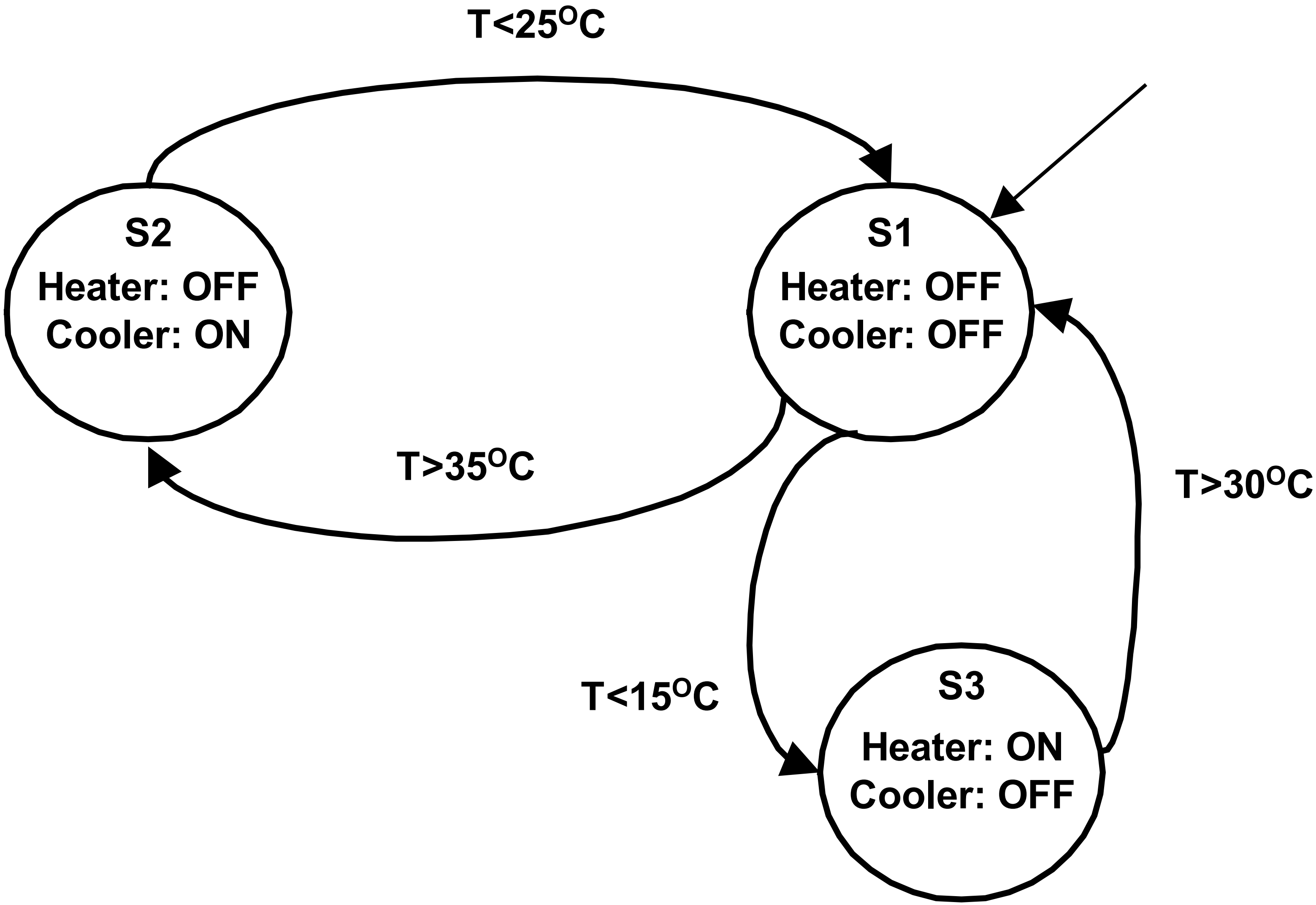
Spring 2025

Reactive Systems

- Typically ES are reactive systems.

"A reactive system is in continual interaction with its environment and executes at a pace determined by that environment."

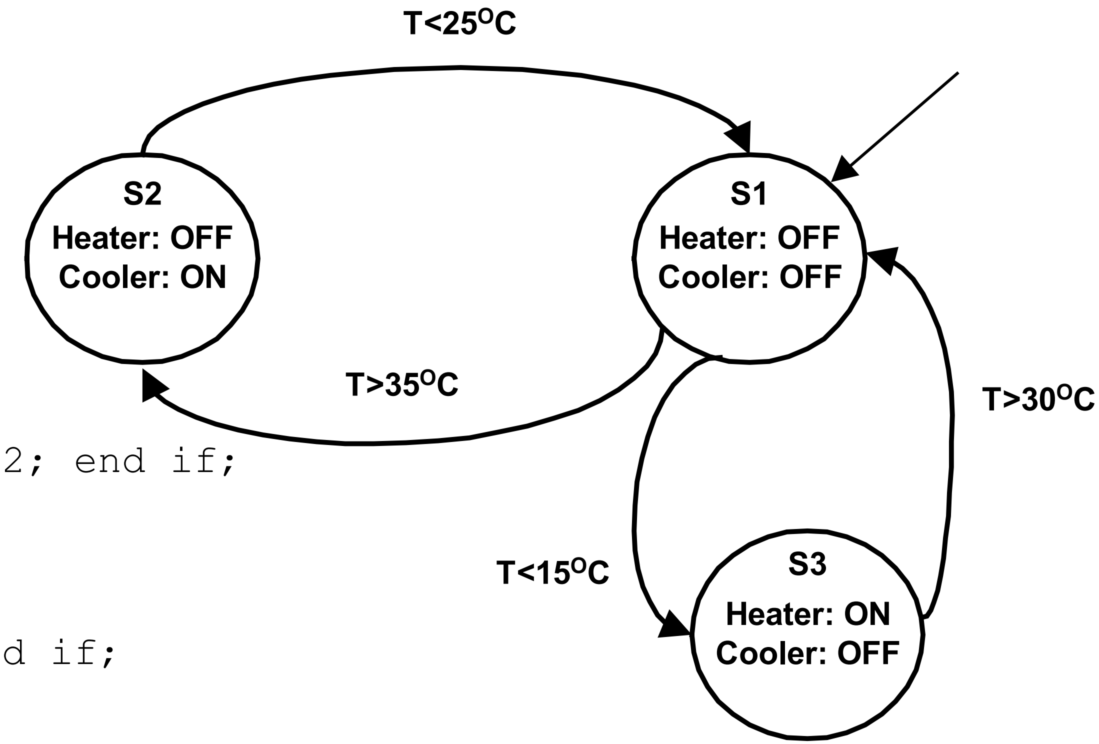
Example: Air Conditioning



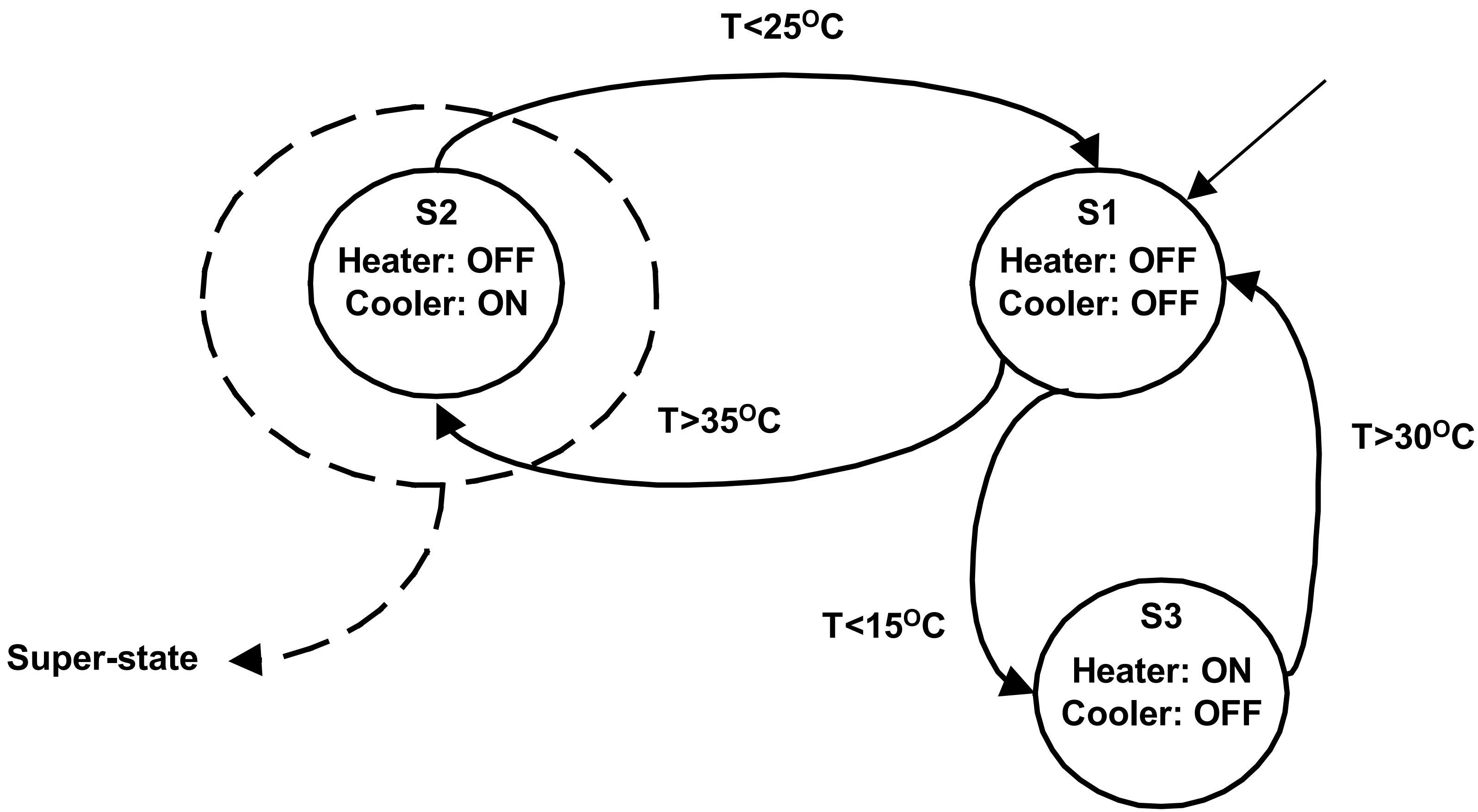
Example: Embedded Software

```
State PS=S1,NS;
Event e;

while(1){
  case(PS){
    S1:
      Turn_off(Heater);
      Turn_off(Cooler);
      e=Wait_for_event();
      if(e=='T<15') NS=S3;
      else if(e=='T>35') NS=S2; end if;
    S2:
      Turn_off(Heater);
      Turn_on(Cooler);
      e=Wait_for_event();
      if(e=='T<25') NS=S1; end if;
    S3:
      Turn_on(Heater);
      Turn_off(Cooler);
      e=Wait_for_event();
      if(e=='T>30') NS=S1; end if;
  }
  PS=NS;
}
```

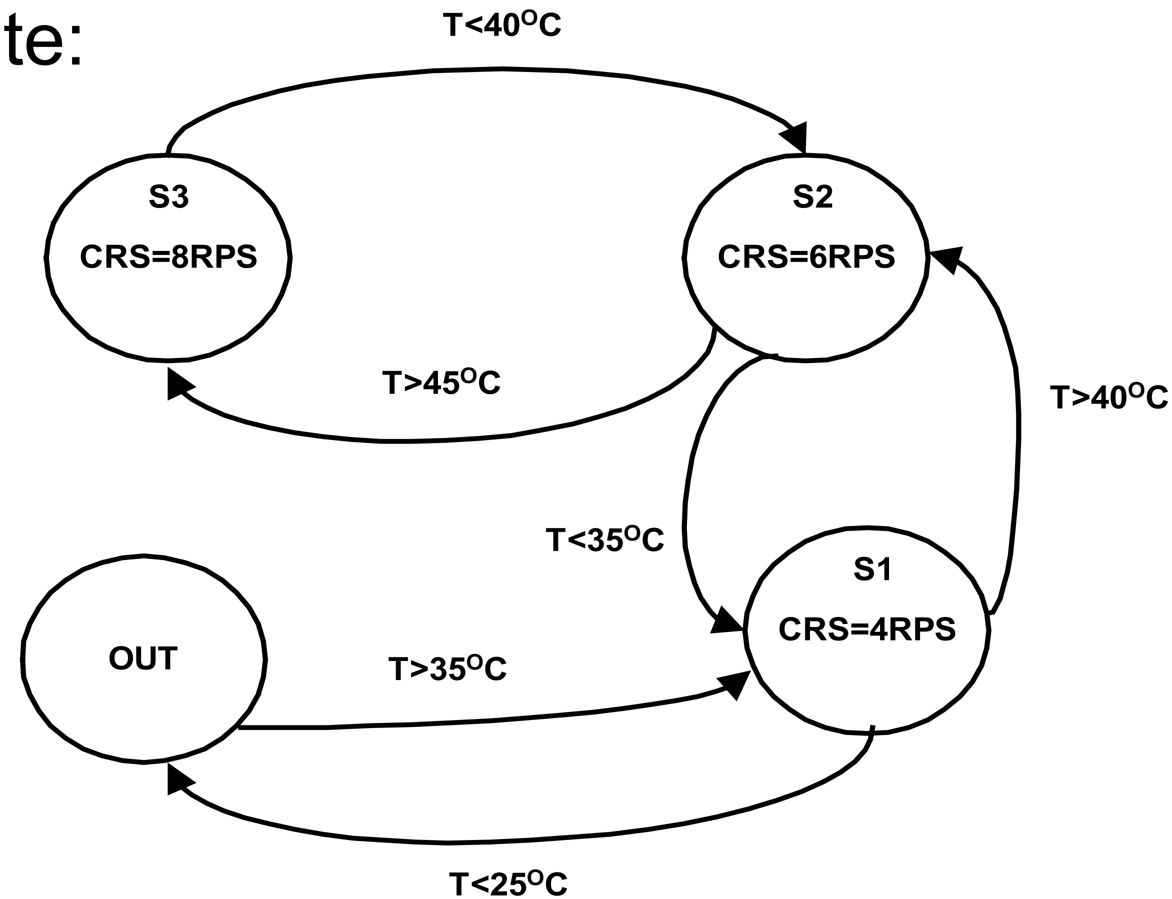


Example: Super-state



Example: Super-state (Cont.)

Super-state:

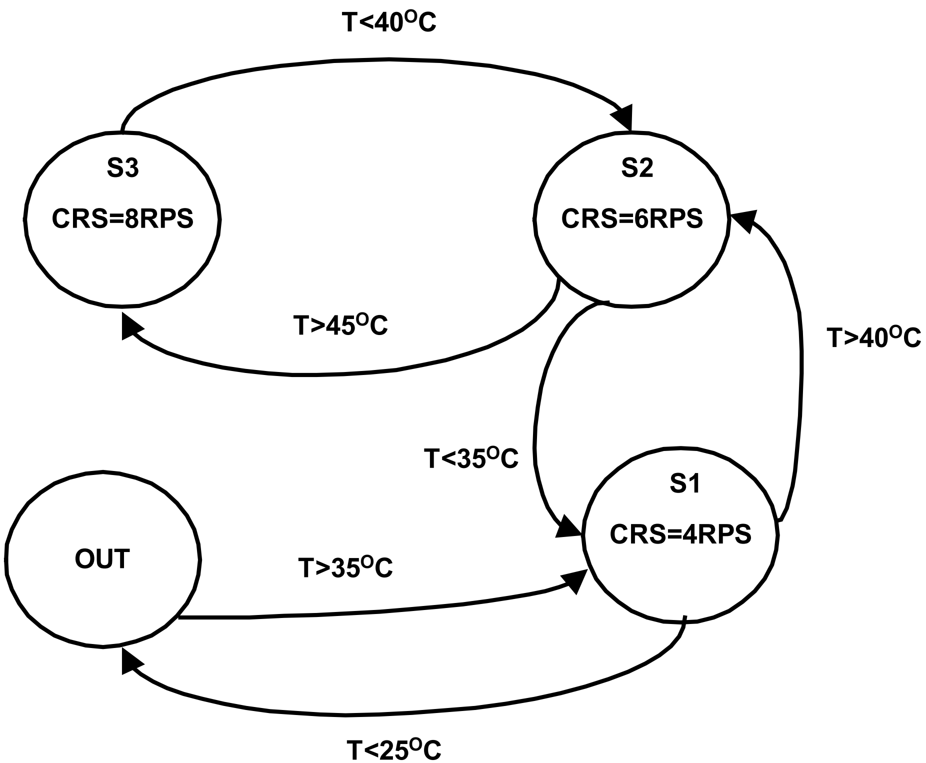


CRS: Cooler Rotational Speed

Example: Embedded Software

```
State PS=S1,NS;
State S2_PS,S2_NS;
Event e;

while(1){
  case(PS){
    S1: ...
    S2:
      Turn_off(Heater);
      Turn_on(Cooler);
      S2_PS=S1;
      while(S2_PS != OUT){
        case(S2_PS){
          S1:
            CRS(4);
            e=Wait_for_event();
            if(e=='T<25') S2_NS=OUT;
            else if(e=='T>40') S2_NS=S2; end if;
          S2:
            CRS(6);
            e=Wait_for_event();
            if(e=='T<35') S2_NS=S1;
            else if(e=='T>45') S2_NS=S3; end if;
          S3:
            CRS(8);
            e=Wait_for_event();
            if(e=='T<40') S2_NS=S2; end if;
        }
        S2_PS=S2_NS;
      }
      if(e=='T<25') NS=S1; end if;
    S3: ...
  }
  PS=NS;
}
```



CRS: Cooler Rotational Speed

Assignment

- Simulate the air conditioning example
  - Use software programming languages, e.g. C, C++, Java, Python, MATLAB, etc.

Advantages of this paradigm

- Some of the advantages:
  - Suitable for reactive systems
  - Hierarchical (e.g. Super-states)
    - Human beings are not capable of comprehending systems with more than 3~5 objects.
  - Verification
    - Each automata is simple and easy to understand
    - Each automata has to comply with the super-state that it belongs to.
  - Automatic code generation

TrueTime Toolbox

- Matlab/Simulink-based simulator
- Co-simulation of embedded systems and electromechanical components.
- Supports
  - DVS
  - Networking protocols (CAN, TTP)
  - Wireless networks (ZigBee)

Assignment

- Run the example ‘Mobile Motes’ of the TrueTime Reference Manual.
- Please write a report about this experiment.