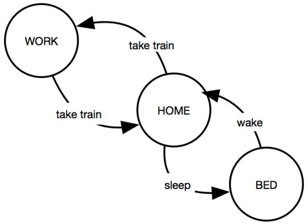
Automation

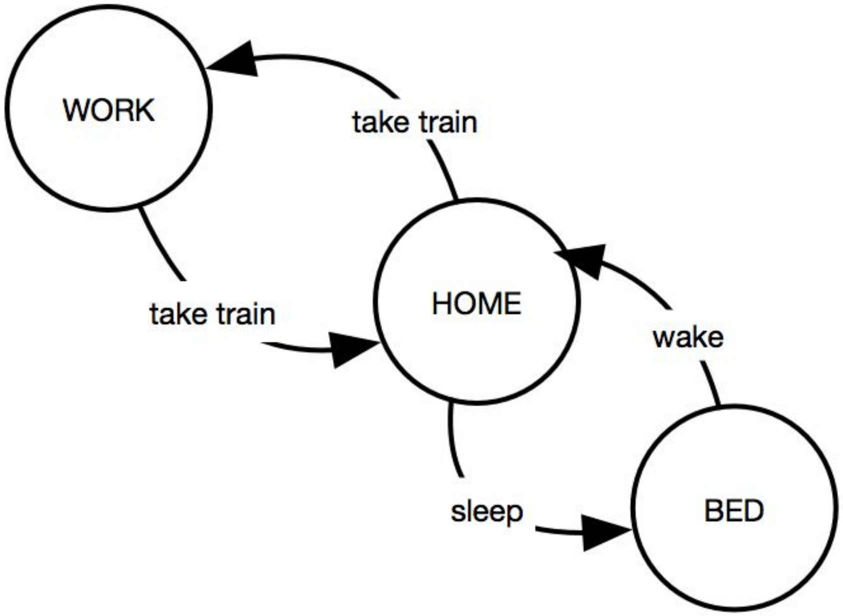
State Machines

Prof. Francesco Maurelli



f.maurelli@jacobs-university.de

State Machines



State Machines

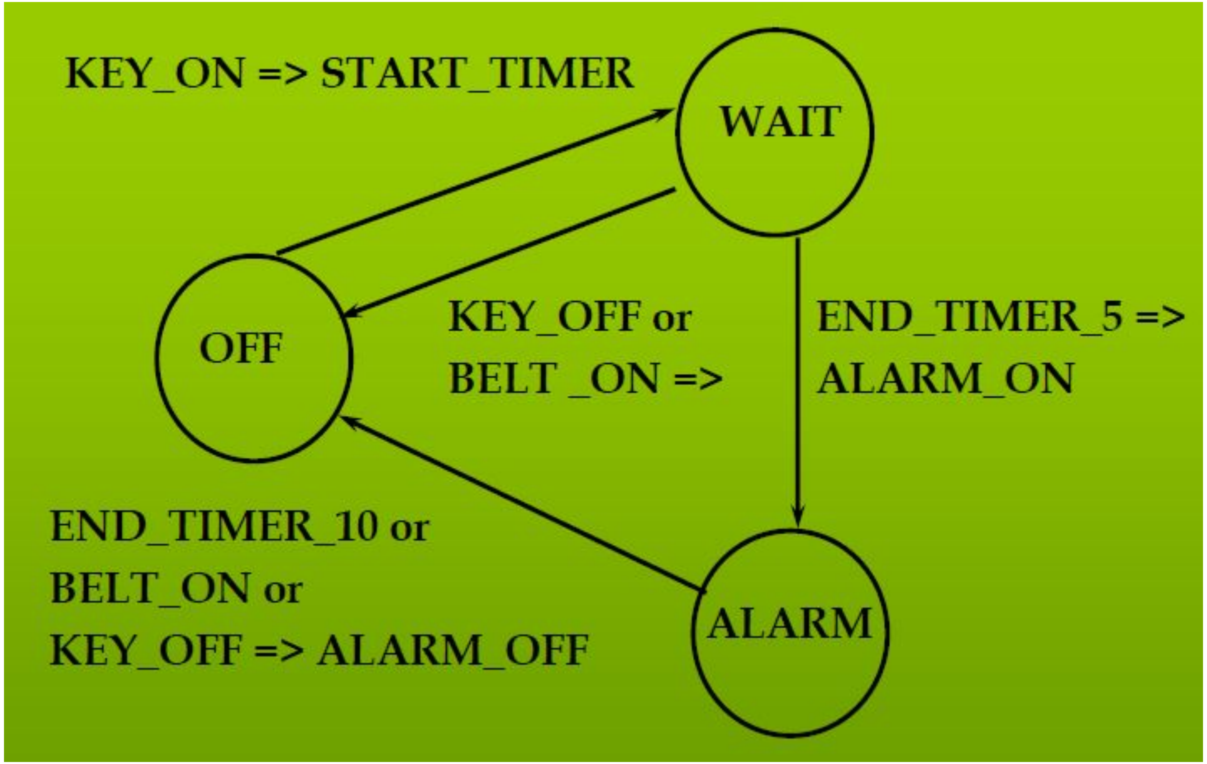
- Functional decomposition into states of operation
- Typical domains of application:
  - control functions
  - protocols (telecom, computers, ...)

- Finite-State Machines or Finite-State *Automata* (*Automaton*)

State Machines / example seat belt alarm

- *If the driver*
    - *turns on the key, and*
    - *does not fasten the seat belt within 5 seconds*
- *then an alarm beeps*
    - *for 5 seconds, or*
    - *until the driver fastens the seat belt, or*
    - *until the driver turns off the key*

State Machines / example seat belt alarm



State Machines - formalism

– FSM = (I, O, S, r, δ, λ)
– I = { KEY_ON, KEY_OFF, BELT_ON, END_TIMER_5, END_TIMER_10 }
– O = { START_TIMER, ALARM_ON, ALARM_OFF }
– S = { OFF, WAIT, ALARM }
– r = OFF

$\delta : 2^I \times S \rightarrow S$
e.g. ({ KEY_OFF }, WAIT) = OFF

$\lambda: 2^I \times S \rightarrow 2^O$
e.g. ({ KEY_ON }, OFF) = { START_TIMER }

Vending Machine Example

- A vending machine accepts nickels, dimes, and quarters.
- Only 1 drink
- Cost is 30 ¢
- If >30¢ is deposited, change is immediately returned.
- When there is 30¢ deposited, if the “dispense” button is pressed, the machine drops a drink.
- It can then accept a new payment

Vending Machine Example

- Input symbol set:
  I = {nickel, dime, quarter, button}
- Output symbol set:
  O = {Æ, 5¢, 10¢, 15¢, 20¢, 25¢, coke}

- State set:
  S = {0, 5, 10, 15, 20, 25, 30}
  Representing how much money has been taken.

Vending Machine Example

Old state	Input	New state	Output
0	n	5	∅
0	d	10	∅
0	q	25	∅
0	b	0	∅
5	n	10	∅
5	d	15	∅
5	q	30	∅
5	b	5	∅

Old state	Input	New state	Output
10	n	15	∅
10	d	20	∅
10	q	30	5¢
10	b	10	∅
15	n	20	∅
15	d	25	∅
15	q	30	10¢
15	b	15	∅

Old state	Input	New state	Output
20	n	25	∅
20	d	30	∅
20	q	30	15¢
20	b	20	∅
25	n	30	∅
25	d	30	5¢
25	q	30	20¢
25	b	25	∅

Old state	Input	New state	Output
30	n	30	5¢
30	d	30	10¢
30	q	30	25¢
30	b	0	coke

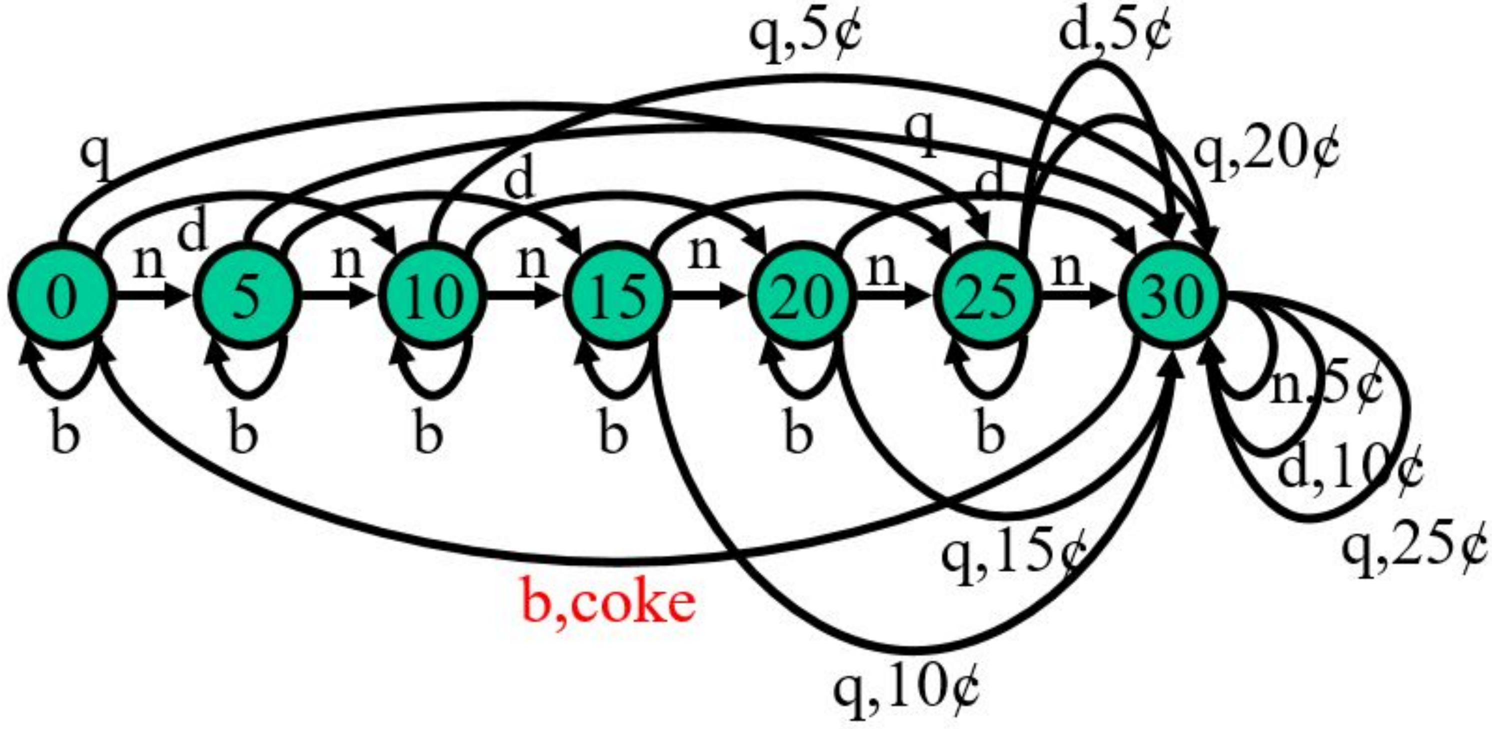
Quite long representation… anything more compact?

Vending Machine Example

Old state	Input Symbol n	d	q	b
0	5,∅	10,∅	25,∅	0,∅
5	10,∅	15,∅	30,∅	5,∅
10	15,∅	20,∅	30,5¢	10,∅
15	20,∅	25,∅	30,10¢	15,∅
20	25,∅	30,∅	30,15¢	20,∅
25	30,∅	30,5¢	30,20¢	25,∅
30	30,5¢	30,10¢	30,25¢	0,coke

Each pair shows new state, output symbol

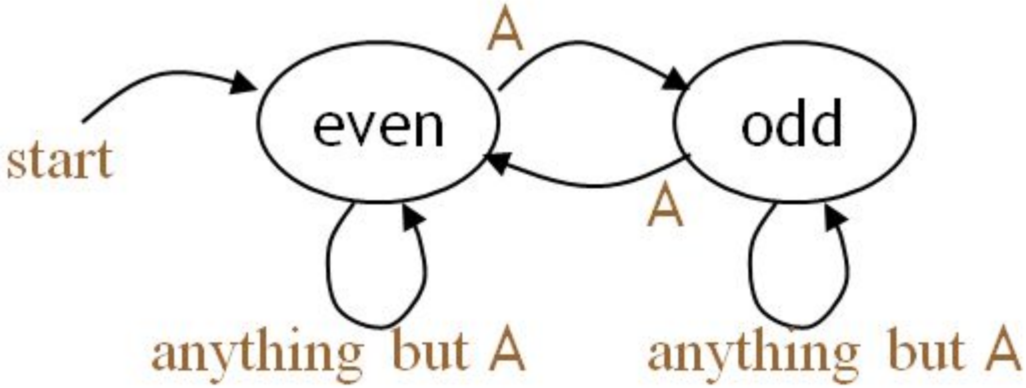
Vending Machine Example



String Example

Build a FSM to determine whether the number of As in a string is even or odd

String Example



Deterministic vs. Non-Deterministic

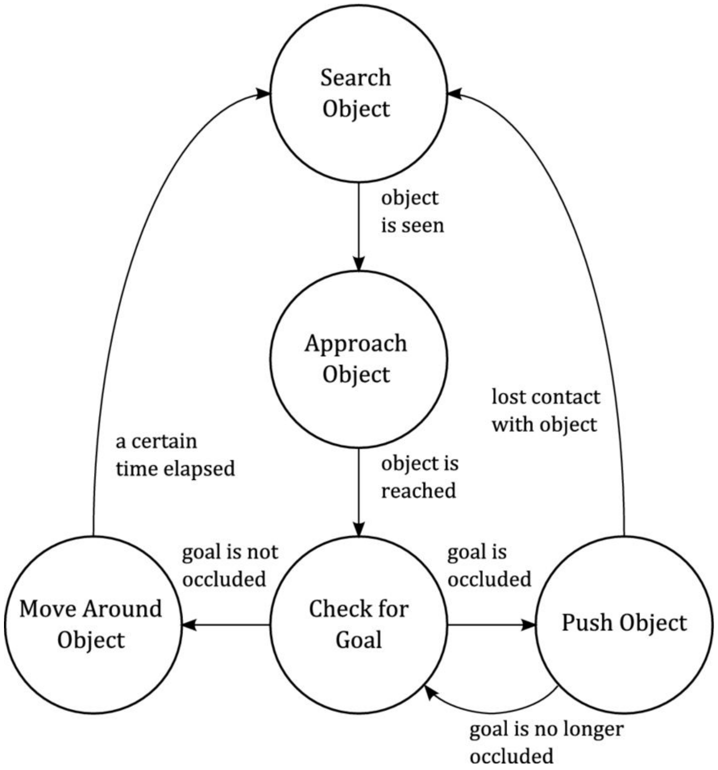
Non-deterministic useful to model:
– an **unspecified** behavior *(incomplete specification)*
– an **unknown** behavior (environment modeling)

δ λ *relations* instead of *functions*

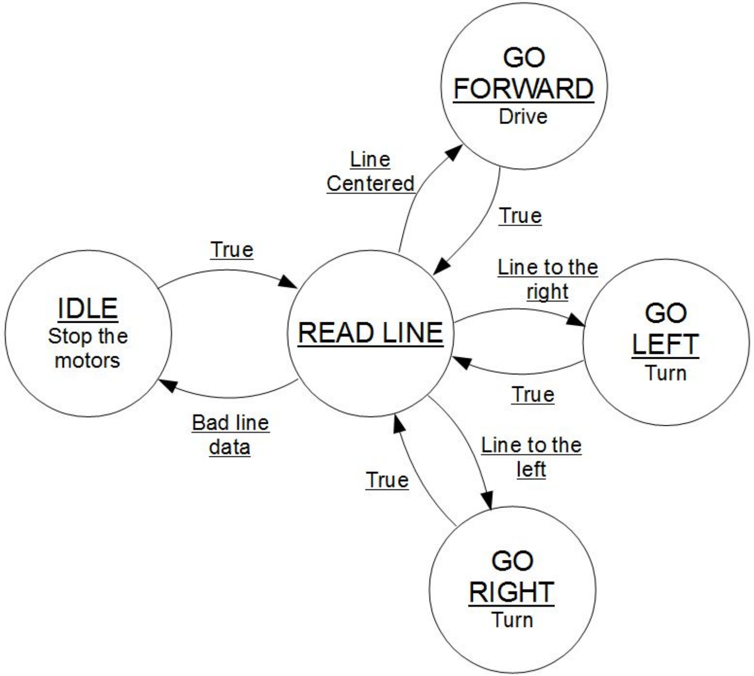
Deterministic: $\delta : 2^I \times S \times S$ e.g. ({ KEY_OFF }, WAIT) = OFF
Non-Deterministic: $\delta \subseteq 2^I \times S \rightarrow S$ e.g. δ({KEY_OFF, END_TIMER_5}, WAIT) = {{OFF}, {ALARM}}

Deterministic: λ: $2^I \times S \rightarrow 2^O$
Non-Deterministic: $\lambda \subseteq 2^I \times S \times 2^O$

State-Machines in Robotics



State-Machines in Robotics



Summary

- State machines
- FSM = (I, O, S, r, δ, λ)
- Deterministic vs Non-deterministic
- Examples