Embedded and Cyber-Physical Systems

- Modeling discrete dynamics-
Actor Models of Discrete Systems

• Example: count the number of cars that enter and leave a parking garage

• Pure signal:  \( up : \mathbb{R} \rightarrow \{ \text{absent}, \text{present} \} \)

• Discrete actor:

\[
\text{Counter}: (\mathbb{R} \rightarrow \{ \text{absent}, \text{present} \})^P \rightarrow (\mathbb{R} \rightarrow \{ \text{absent} \} \cup \mathbb{N})
\]

\[
P = \{ \text{up}, \text{down} \}\\
\]
Reaction

For any $t \in \mathbb{R}$ where $up(t) \neq \text{absent}$ or $down(t) \neq \text{absent}$ the Counter *reacts*. It produces an output value in $\mathbb{N}$ and changes its internal state.

**Counter**: $(\mathbb{R} \to \{\text{absent, present}\})^P \to (\mathbb{R} \to \{\text{absent}\} \cup \mathbb{N})$

$P = \{\text{up, down}\}$
Input and Output Valuations at a Reaction

For $t \in \mathbb{R}$ a port $p$ has a valuation, which is an assignment of a value in $V_p$ (the type of port $p$). A valuation of the input ports $P = \{\text{up, down}\}$ assigns to each port a value in $\{\text{absent, present}\}$.

A reaction gives a valuation to the output port $count$ in the set $\{\text{absent}\} \cup \mathbb{N}$. 
Finite State Machines

- Input events
- Guards
- Output events
Garage Counter Example

- Counting cars in a parking garage:

  **inputs:** $up, down : pure$
  
  **output:** $count : \{0, \cdots, M\}$

  $$
  \begin{align*}
  &up \land \neg down / 1 & up \land \neg down / 2 & up \land \neg down / 3 & up \land \neg down / M \\
  &down \land \neg up / 0 & down \land \neg up / 1 & down \land \neg up / 2 & down \land \neg up / M - 1
  \end{align*}
  $$

- The notation here is a bit awkward, because the parameter $M$ may be large, and we are stuck using a somewhat informal ... notation.
Extended State Machines

- Extended state machines augment the FSM model with variables that may be read or written

What is the size of the state space?
Traffic light example (continuous)

Continuous variable: $x(t) : \mathbb{R}$

Inputs: pedestrian: pure

Outputs: sigR, sigG, sigY: pure

- **Green**
  - $x(t) \geq 60 / \text{sigG}$
  - $\dot{x}(t) = 1$
  - $x(t) := 0$

- **Red**
  - $x(t) \geq 5 / \text{sigR}$
  - $\dot{x}(t) = 1$
  - $x(t) := 0$

- **Yellow**
  - $x(t) \geq 60 / \text{sigY}$
  - $\dot{x}(t) = 1$
  - $x(t) := 0$

- **Pending**
  - $\text{pedestrian} \land x(t) < 60 /$
  - $\dot{x}(t) = 1$
  - $x(t) := 0$
Traffic light example (discrete)

**variable:** count: \( \{0, \cdots, 60\} \)

**inputs:** pedestrian : pure

**outputs:** sigR, sigG, sigY : pure

Default transition with implicit guard / action (true / none)
When does a reaction occur?

• When a reaction occurs is not specified in the state machine itself. It is up to the environment.

$$\text{variable: } count \in \{0, \ldots, 60\}$$

$$\text{count} := \text{count} + 1$$

• This traffic light controller design assumes one reaction per second. This is a \textit{time-triggered model}. 
When does a reaction occur?

- Suppose all inputs are discrete and a reaction occurs when any input is present. Then the above transition will be taken whenever the current state is $s_1$ and $x$ is present.

- This is an **event-triggered model**.
(Non)Determinism

• A state machine is **deterministic** if, for each state, at most one outgoing transition is enabled by each input.

• Example of non-deterministic state machine:
Actor Model of an FSM

This model enables **composition** of state machines.
Composition of State Machines

• Concurrent composition:
  – Side-by-side
  – Cascade
  – Feedback

• Hierarchical composition
A key question: When do these machines react?
Two possibilities:
• Together (synchronous composition)
• Independently (asynchronous composition)
Synchronous Composition

\[ S_C = S_A \times S_B \]

Note that these two states are not reachable.

Synchronous composition
Asynchronous Composition

$$S_C = S_A \times S_B$$

Asynchronous composition using interleaving semantics

outputs: $a$, $b$ (pure)

Note that now all states are reachable.
Syntax vs. Semantics

Synchronous or Asynchronous composition?

If asynchronous, does it allow simultaneous transitions in A & B?
Cascade Composition

\[(\text{States}, \text{Inputs}, \text{Outputs}, \text{update}, \text{initialState})\]

- \[(\text{States}_A, \text{Inputs}_A, \text{Outputs}_A, \text{update}_A, \text{initialState}_A)\]
- \[(\text{States}_B, \text{Inputs}_B, \text{Outputs}_B, \text{update}_B, \text{initialState}_B)\]

Output port(s) of A connected to input port(s) of B
Example: Traffic Light

variable: count: \{0, \cdots, 60\}
inputs: pedestrian : pure
outputs: sigR, sigG, sigY : pure
Example: Pedestrian Light

variable: \( pcount: \{0, \cdots, 55\} \)
input: \( \text{sigR}: \text{pure} \)
outputs: \( \text{pedG}, \text{pedR}: \text{pure} \)

\[
\begin{align*}
pcount := 0 \\
pcount \geq 55 / \text{pedR} \\
\text{sigR} / \text{pedG} \\
pcount := pcount + 1 \\
\text{pedR} : \text{pure}
\end{align*}
\]

This light stays green for 55 seconds, then goes red. Upon receiving a sigR input, it repeats the cycle.
What is the size of the state space of the composite machine?
Synchronous composition

unsafe states
Synchronous composition with unreachable states removed
Feedback Composition

\[(\text{States}, \text{Inputs}, \text{Outputs}, \text{update}, \text{initialState})\]

\[(\text{States}_A, \text{Inputs}_A, \text{Outputs}_A, \text{update}_A, \text{initialState}_A)\]

\[\text{Outputs}_A \subseteq \text{Inputs}_A\]
Observation: Any Composition is a Feedback Composition

If every actor is a function, then the semantics of the overall system is the least $s \in S^N$ such that $F(s) = s$.

The behavior of the system is a “fixed point.”
Discrete feedback models

- **Well-formed**: a unique fixed point exists

- **Ill-formed**: no fixed point

- **Multiple fixed points**