State Machine Models

Finite State Machine (FSM) is a 5-tuple with the components:

- $States, Inputs, Outputs$
- *update* function
- *initialState*

- **Reaction**: An instantaneous computational step.
- **State**: A system's condition, encoding all necessary past information.
- **Transition Notation**: `guard / action`. The **guard** is a predicate on inputs/variables that must be true to fire. The **action** specifies the outputs produced.
  - **Guards**: `x` (present), ¬`x` (absent), `temp > 20`.
  - **Actions**: `/ y` (output y is present), `/ y:=0` (output y is 0), `/` (all outputs absent).
- **Pure Signal**: A signal that is only 'present' or 'absent'.
- **Default Transition**: (Often a dashed line). Taken if and only if no other non-default transition from the same state is enabled.
- **Determinism**: For any state and input, exactly one transition is enabled.
- **Receptiveness**: For any state and input, at least one transition is enabled (the machine never "gets stuck").

Example: Formal 5-Tuple Description

- $States = \{\text{red, yellow, green}\}$
- $Inputs : \{\text{tick}\} \to \{\text{pres, abs}\}$
- $Outputs : \{\text{go, stop}\} \to \{\text{pres, abs}\}$
- $initialState = \text{red}$

(pres=present, abs=absent)

The update function is:

$$update(s, i) = \begin{cases} (\text{green, go}) & \text{if } s = \text{red} \wedge i(\text{tick}) \\ (\text{yellow, stop}) & \text{if } s = \text{green} \wedge i(\text{tick}) \\ (\text{red, stop}) & \text{if } s = \text{yellow} \wedge i(\text{tick}) \\ (s, \text{absent}) & \text{otherwise} \end{cases}$$

Extended State Machine (ESM) augments an FSM with variables.

- **Transition Notation**:
  ```
  guard / output-action
        set-action
  ```
- **Execution Order**: The 'guard' and 'output-action' read the *current* values of variables. The 'set-action' (e.g., `c := c + 1`) updates the variables for the *next* state.

Behavior and Traces

- **Nondeterminism**: From a given state, one input can enable multiple transitions. Behavior is represented by a **computation tree**.
- **Execution Trace**: Sequence of states, inputs, and outputs.
  Notation: $s_0 \xrightarrow{x_0/y_0} s_1 \ldots$.

Temporal Logic (LTL)

Specifies properties over all possible execution traces of a state machine.

Operator	Sym.	Meaning
Globally	Gp	"p is always true from now on" (Invariant)
Finally	Fp	"p is eventually true"
Next	Xp	"p is true in the next state"
Until	pUq	"p holds until q holds" (q must happen)

- **Common Patterns**:
  - **Response**: $G(p \Rightarrow Fq)$
  - **Stabilization**: FGp
  - **Progress**: GFp
- **Operator Equivalences**:
  - $Gp \equiv \neg F \neg p$
  - $Fp \equiv \texttt{true}\ \mathrm{U}\ p$

Key Formulas from Exercises

- **Harmonic Motion**: $\ddot{y}(t) = -\omega_0^2 y(t)$
  - Describes frictionless oscillation.
  - $y(t)$: displacement (m), ω_0: natural frequency (rad/s).
- **DC Motor Torque**: $K_T i(t) - x(t) = I \frac{d}{dt}\omega(t)$
  - Relates torques to angular acceleration.
  - K_T: torque constant (Nm/A), $i(t)$: current (A), $x(t)$: load torque (Nm), I: inertia (kg·m^2), $\omega(t)$: angular velocity (rad/s).
- **DC Motor Voltage**: $v(t) = Ri(t) + K_b\omega(t)$
  - Electrical equation including back-EMF.
  - $v(t)$: voltage (V), R: resistance (Ω), K_b: back-EMF constant (V·s/rad).
- **Sensor Affine Model**: $f(x) = ax + b$
  - Linear model of sensor output.
  - $f(x)$: digital output (counts), x: physical value, a: sensitivity, b: bias.
- **Dynamic Range (dB)**: $20 \log_{10} \frac{H-L}{p}$
  - Sensor's range in decibels.
  - H, L: range limits, p: precision.
- **ADC/DAC Resolution**: $V_{\text{ref}}/2^n$
  - The smallest voltage step for an n-bit converter.
  - V_{ref}: reference voltage (V), n: number of bits.

Advanced State Machine Concepts

Composition of FSMs

- **Composite State Space**: $States_C = States_A \times States_B$.
- **Synchronous**: Components react simultaneously.
- **Asynchronous**: Components react independently (e.g., interleaving).
- **Cascade**: Output of A is input to B. In a synchronous step, A reacts, then B reacts immediately.
- **Shared Variables**: To avoid race conditions in synchronous composition, a sequential execution order within the step must be defined (e.g., "A reacts before B").

Hierarchical FSMs (Statecharts)

- **State Refinement**: A state can contain a nested FSM.
- **Preemptive Transition (•)**: A transition from a superstate whose guard is checked *before* its refinement. If taken, the refinement is skipped.
- **History Transition ('H') (default)**: Resumes the refinement in its last-visited sub-state.
- **Reset Transition**: Resets the refinement to its initial sub-state upon entry.

Hybrid Systems & Time

Hybrid Automaton

- **Definition**: Models systems with mixed discrete (modes) and continuous (ODEs) behavior.
- **Modes**: Discrete FSM states.
- **Continuous Dynamics**: Differential equations within a mode (e.g., $\dot{x}(t) = f(x)$).
- **Guards**: Conditions on continuous variables that trigger transitions (e.g., $s(t) \geq T$).
- **Set Actions**: Reset continuous variables on a transition (e.g., $s(t) := 0$).

Timed Automaton

- A hybrid automaton where continuous variables are clocks that run at a constant rate (e.g., $\dot{s}(t) = 1$). Used for real-time specifications.

Physical Modeling & Interface

Sensor & Actuator Formulas

- **Affine Model**: $f(x) = ax + b$
- **Dynamic Range (dB)**: $20 \log_{10} \frac{H-L}{p}$
- **ADC/DAC Resolution** (n bits):
  - Step Size (V/bit) = $V_{\text{ref}}/2^n$
- **Finding required bits 'n'** from Dynamic Range (DR):
  - Levels 2^n must cover the range ratio $\frac{H-L}{p}$.
  - From DR in dB: $\frac{H-L}{p} = 10^{\text{DR}_{\text{dB}}/20}$.
  - Solve for n: $2^n \geq \frac{H-L}{p}$.
- **Digital Output (ADC)** = $\lfloor V_{\text{in}}/\text{Step Size} \rfloor$.
- **Analog Output (DAC)** = `Decimal_Input` × `Step_Size`.

I/O Concepts

- **Polling**: CPU continuously checks I/O status in a loop.
- **Interrupts**: Hardware signals the CPU on an event. The CPU runs an **Interrupt Service Routine (ISR)**.

Memory & Security Notes

- **Key Security Goals**:
  - **Confidentiality**: Being secret from an attacker.
  - **Integrity**: Being unmodified by an attacker.
  - **Authenticity**: Knowing the identity of agents.
  - **Availability**: System provides sufficient quality of service.
- **Memory Hierarchy**: Registers > Cache > RAM > Disk.
- **Cache**: Small, fast memory. Direct-mapped caches are simple but prone to thrashing.
- **Software Memory**: **Stack** for local variables (LIFO). **Heap** for dynamic allocation ('malloc').
- **One-Time Pad**: A secure cipher, but the key must never be reused.
- **Buffer Overflow**: Writing past a buffer's boundary, corrupting adjacent memory.

Example: Automaton (Parallel Processors)

System with two processors, P1 (cap=1) and P2 (cap=2). State is (P_1, P_2). A list of transitions include: S1(0,0) → S4(1,0) on J1 arrival; S1(0,0) → S2(0,1) on J2 arrival; S2(0,1) → S1(0,0) on J2 leaves; etc.

Example: Automaton (3-Server System)

System of 3 servers S1, S2, S3 with total capacity of 2 customers. State is (s_1, s_2, s_3).

- Routing Policy: After S1, send to shorter queue (S2 or S3), with ties going to S2.
- Note: States like $(0, 2, 0)$ and $(0, 0, 2)$ are unreachable due to the total capacity constraint.

Common SI Prefixes

Prefix	Symbol	Factor
Giga	G	10^9
Mega	M	10^6
kilo	k	10^3
milli	m	10^{-3}
micro	μ	10^{-6}
nano	n	10^{-9}