Formal Method in Software Engineering (SE-313)

Course Teacher
Assistant Professor
Engr. Mustafa Latif

Finite-State Machines

Finite-State Machines

- Finite-state machines provide a simple computational model with many applications.
- Finite-state machines, also called finite-state automata (singular: automaton) or just finite automata.
- Application: Pattern matching(Regular expressions), model protocols, electronic circuits, Theory is used in model-checking as well.

Finite-State Machines

- Finite automata(FA) are finite collections of states with transition rules that take you from one state to another.
- Original application was sequential switching circuits.
- Today, several kinds of software can be modeled by FA.

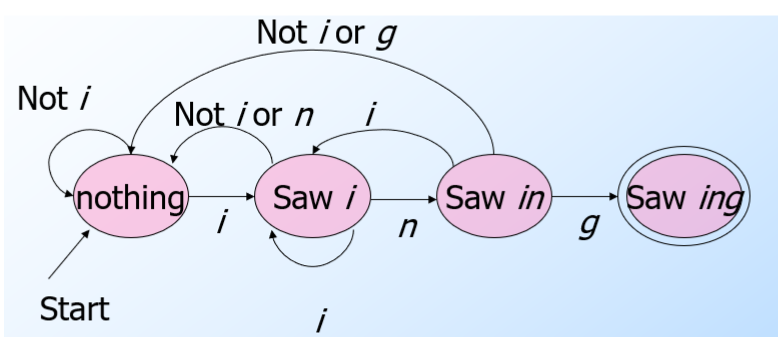
Finite-State Machines

Representing FA

- Simplest representation is often a graph.
  - Nodes = states.
  - Arcs indicate state transitions.
  - Labels on arcs tell what causes the transition.

Finite-State Machines

Example: Recognizing Strings Ending in "ing"



Finite-State Machines

Automata to Code

- In C/C++, make a piece of code for each state. This code:
- Reads the next input.
- Decides on the next state.
- Jumps to the beginning of the code for that state.

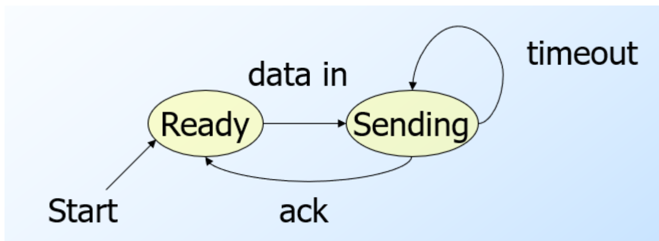
Finite-State Machines

Example: Automata to Code

```
2: /* i seen */
    c = getNextInput();
    if (c == ’n’) goto 3;
    else if (c == ’i’) goto 2;
    else goto 1;
3: /* ”in” seen */
    . . .
```

Finite-State Machines

Example: Protocol for Sending Data



Finite-State Machines

Definition

- A finite-state machine (FSM) is an abstract mathematical machine that consists of a finite number of states. It includes a start state q_0 in which the machine is in initially; a finite set of states Q; an input alphabet $\sum$; a state transition function δ and a set of final accepting states F (where $F \subseteq Q$).
- The state transition function δ takes the current state and an input symbol and returns the next state. That is, the transition function is of the form:

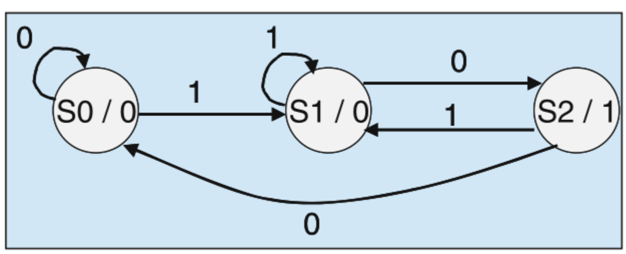
$$\delta : Q \times \Sigma \to Q$$

Finite-State Machines

- State diagrams are used to represent finite-state machines, and each state accepts a finite number of inputs.
- A finite-state machine may be deterministic or non-deterministic,
- A deterministic machine changes to exactly (or at most) one state for each input transition,
- A non-deterministic machine may have a choice of states to move to for a particular input symbol.
- Advance computing model of FA is Turing machine.

11

Finite-State Machines



Finite-state machine with output

- The behavior of the system at a point in time is determined from its current state and input, with behavior defined for the possible input to that state.
- The system starts in an initial state.

12

Finite-State Machines

- A finite-state machine (also known as finite-state automata) is a quintuple ($\sum$, Q, δ, q_0, F).
- The alphabet of the FSM is given by $\sum$;
- the set of states is given by Q;
- The transition function is defined by $\delta : Q \times \sum \rightarrow Q$;
- the initial state is given by q0 and
- the set of accepting states is given by F (where F is a subset of Q).
- A string is given by a sequence of alphabet symbols; that is, s $\in \sum^*$, and the transition function δ can be extended to $\delta^* : Q \times \sum^* \longrightarrow Q$.
- The set of strings (or language) accepted by an automaton M is denoted L(M).
- A language is termed regular if it is accepted by some finite-state machine.

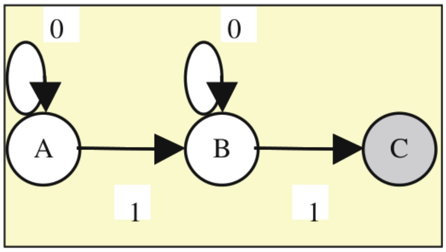
13

Finite-State Machines

- A string s $\in \sum^*$ is accepted by the finite-state machine if $\delta^*(q_0, s) = q_f$ where $q_f \in F$, and the set of all strings accepted by a finite-state machine is the language generated by the machine.
- A finite-state machine is termed deterministic if the transition function δ is a function, otherwise (where it is a relation) it is said to be non-deterministic.

14

Finite-State Machines



Deterministic FSM

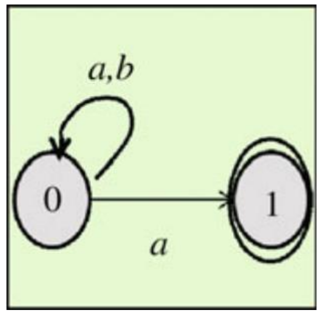
Table 13.1 State transition table

State	0	1
A	A	B
B	B	C
C	–	–

- **For the example above, the input alphabet is given by ∑ = {0, 1}; the set of states by Q= {A, B, C}; the start state by q_o=A; the accepting states by F={C} and the transition function is given by the state transition table below .**

Finite-State Machines

- **A non-deterministic automaton (NFA) or non-deterministic finite-state machine is a finite-state machine where from each state of the machine and any given input, the machine may go to several possible next states.**
- **NFA is defined formally as a 5-tuple (Q, ∑ , *δ*, q_o, F) as in the definition of a deterministic automaton, and the only difference is in the transition function *δ* .** $\delta : Q \times \Sigma \rightarrow \mathbb{P}Q$



Non-deterministic finite-state machine

17