01204211 Discrete Mathematics
Lecture 8b: Finite automa[1]

Jittat Fakcharoenphol

August 24, 2023

[1]Based on lecture notes of *Models of Computation* course by Jeff Erikson.

Example: syntax highlighting

HTML tokenizer

Game programming

State-transition graphs

More examples over $\Sigma = \{0, 1\}$

All strings, except 010.

Strings containing the subsequence 010.

Formal definitions

A **finite-state machine** or a **deterministic finite-state automaton** (DFA) has five components:

Formal definitions

A **finite-state machine** or a **deterministic finite-state automaton** (DFA) has five components:

- the input alphabet Σ,

Formal definitions

A **finite-state machine** or a **deterministic finite-state automaton** (DFA) has five components:

- the input alphabet Σ,
- a finite set of states Q,

Formal definitions

A **finite-state machine** or a **deterministic finite-state automaton** (DFA) has five components:

- the input alphabet Σ,
- a finite set of states Q,
- a transition function δ

Formal definitions

A **finite-state machine** or a **deterministic finite-state automaton** (DFA) has five components:

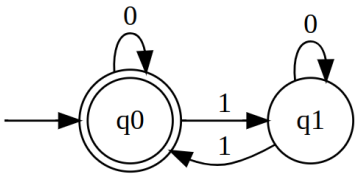
- the input alphabet Σ,
- a finite set of states Q,
- a transition function $\delta : Q \times \Sigma \longrightarrow Q$

Formal definitions

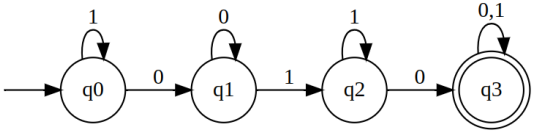
A **finite-state machine** or a **deterministic finite-state automaton** (DFA) has five components:

- the input alphabet Σ,
- a finite set of states Q,
- a transition function $\delta : Q \times \Sigma \longrightarrow Q$
- a start state $s \in Q$, and
- a subset $A \subseteq Q$ of accepting states.

Example 1



Example 2



Moves

One step move: from state q with input symbol a, the machine changes its state to

Moves

One step move: from state q with input symbol a, the machine changes its state to $\delta(q, a)$.
Extension: from state q with input string q, the machine changes its state to $\delta^*(q, w)$ defined as

Moves

One step move: from state q with input symbol a, the machine changes its state to $\delta(q, a)$.
Extension: from state q with input string q, the machine changes its state to $\delta^*(q, w)$ defined as

$$\delta^*(q, w) = \begin{cases} q & \text{if } w = \varepsilon, \\ \delta^*(\delta(q, a), x) & \text{if } w = ax. \end{cases}$$

The signature of δ^* is $Q \times \Sigma^* \longrightarrow Q$.

Acceptance

For a finite-state machine with starting state s and accepting states A, it accepts string w iff

Acceptance

For a finite-state machine with starting state s and accepting states A, it accepts string w iff

$$\delta^*(s, w) \in A.$$

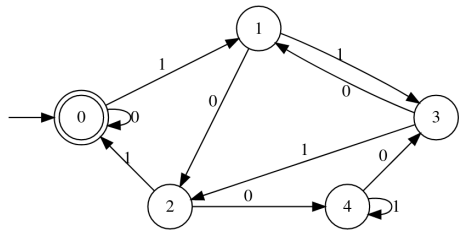
Multiple of 5

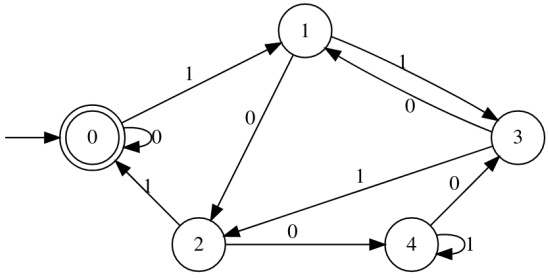
Multiple of 5

```
def multiple_of_5(w):
    r = 0
    for i in w:
        r = (2*r + w) % 5
    return r == 0
```

Multiple of 5

```
def multiple_of_5(w):
    r = 0
    for i in w:
        r = (2*r + w) % 5
    return r == 0
```




0
1
2
3
4
1
0
1
0
1
0
1
0
0
1

Digital design: Implementation

Digital design: Moore and Mealy machines

In the digital design class, you will encounter finite-state machines as well. The version we consider in this class is refered to as a **Moore machine**.
In practices, there is another variant of FSM called **Mealy machines**, whose outputs depend on input symbols as well.

Digital design: Moore and Mealy machines

In the digital design class, you will encounter finite-state machines as well. The version we consider in this class is refered to as a **Moore machine**.
In practices, there is another variant of FSM called **Mealy machines**, whose outputs depend on input symbols as well.
Formally, they differ in output function.

- Moore machine: $G : Q \longrightarrow [0, 1]$
- Mealy machine: $G : Q \times \Sigma \longrightarrow [0, 1]$

Example: even number of 1's

Example: strings containing 00 as a substring

Combining DFAs

What if we want to build a DFA that accepts strings with an even number of 1's and containing 00 as a substring?