

# 1 Finite state machine minimization

- *Algorithm:* Finite state machine minimization (algo. 1)
- *Input:* A finite state machine
- *Complexity:*  $\mathcal{O}(n^{2 \log n})$
- *Data structure compatibility:* Finite state machine (Deterministic finite automaton)
- *Common applications:* compilers, network protocols, theory of computation

## Finite state machine minimization

Given a finite state machine, minimize the states.

## Description

### 1. Definition of FSM

The formal definition of a finite state machine (deterministic finite automaton) is

$$M : (Q, \Sigma, \delta, q_0, F)$$

- (a)  $Q$ : finite set of states
- (b)  $\Sigma$ : finite set of input symbols
- (c)  $\delta : Q \times \Sigma \leftarrow Q$ : transition function
- (d)  $q_0 \in Q$ : initial state
- (e)  $F \subseteq Q$ : accept state

The automaton will accept a string  $w$  if it starts at start state  $q_0$ , and given each character in  $w$ , the transition rule will transit state to state according to  $\delta$ , and the final state shall halt at  $F$  states.

### 2. Input

The input of the algorithm shall be a finite state machine.

### 3. Complexity

---

**Algorithm 1:** FSM minimization

---

**Input :**

**Output:**

**1 return**

---

## References

- If available provide URLs, e.g. <http://mywebsite.org>
- Wikipedia is not acceptable if this is the unique reference
- Reference some books, or published articles
- Use reliable websites (no blog allowed) that are not likely to disappear any time soon