Chapter 3
Finite State Automata

Plan

1. Deterministic Finite Automata (AEF)
2. Representations of an Automaton
3. Equivalent and Complete Automata
4. Non-Deterministic Finite Automata (Determinization)
5. Generalized Automata

Deterministic Finite State Automaton

Definition:

A simple and deterministic finite state automaton is a **5-tuple**,

denoted as $\mathcal{A}$ **< X, S, S_0, F, $\mathbb{I}$ >**, where:

- **X** : An alphabet,
- **S** : The set of states of the automaton,
- **S_0** : $S_0 \in S$ is the initial state,
- **F** : $F \subseteq S$ is the set of final states,
- **$\mathbb{I}$** : The set of instructions (transitions) $\mathbb{I} : S \times X \to S$ (single-letter reader).

Principle

- The elements of 𝕀 are triplets, denoted as **(Si, xi, Sj)**
  where **Si, Sj ∈ S** and **xi ∈ X**.
- i.e., if the automaton is in state **Si**, and under the reader's head there is the letter **xi**, then the automaton transitions to state **Sj**.

Representation of the Set of Instructions

- Matrix Representation :

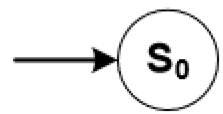
	x_1	x_2	………	x_i	……..	$x_n \in$ alphabet
S_0						
S_1						
…..						
S_i				S_j		
…….						
…….						
$S_k \in S$						

Representation of the Set of Instructions

- **Graphical representation:**

- **Initial state:**

  Represented by S_0 with an incoming arrow.

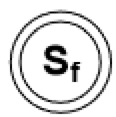


- **Simple state:**

  Represented by S_i.

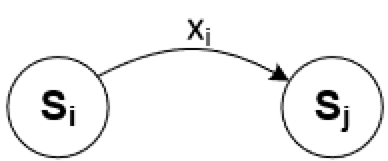
- **Final state:**

  Represented by S_f with a double circle.



- **A transition:**

  Represented as $S_i \xrightarrow{x_i} S_j$, meaning that from state S_i, the automaton transitions to state S_j upon reading input x_i.



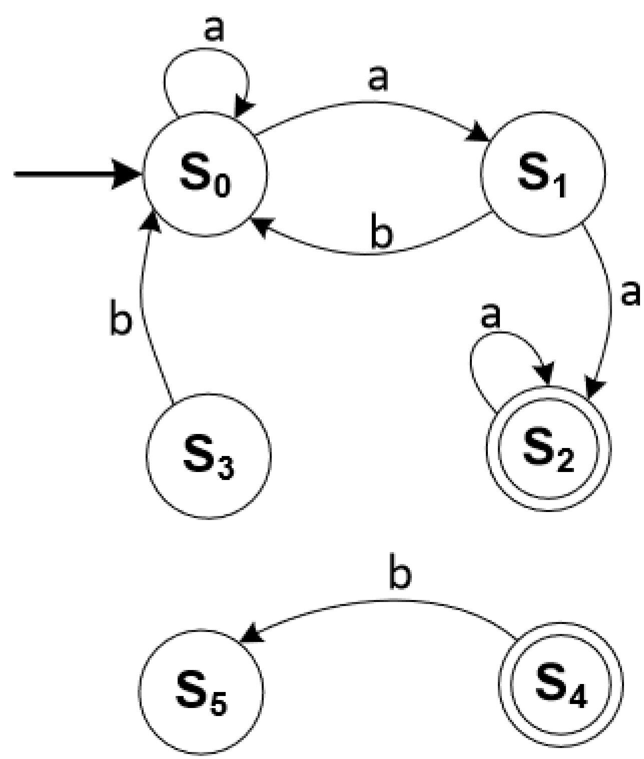
Example

- Let $\mathcal{A} < X, S, S0, \mathbb{F}, \mathbb{I} >$ be a final state automaton such that:

- $S = \{S_0, S_1, S_2, S_3, S_4, S_5\}$
- $\mathbb{F} = \{S_0, S_2, S_4\}$
- $\mathbb{I} = \{(S_0, a, S_0), (S_0, b, S_1), (S_1, b, S_0),$
  $(S_1, a, S_2), (S_2, a, S_2), (S_3, b, S_0), (S_4, b, S_5), \}$

Example

	a	b
S_0	S_0	S_1
S_1	S_2	S_0
S_2	S_2	-
S_3	-	S_0
S_4	-	S_5
S_5	-	-

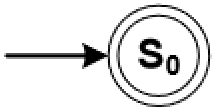


Definition

- **Given** $\mathcal{A} < X, S, S_0, F, \mathbb{I} >$ **is a finite state automaton and let** $\omega = \omega_1\omega_2 \ldots \omega_n$ be a word in X^n where $\omega_i \in X$.
  We say that $\omega_1\omega_2 \ldots \omega_n$ is **recognized** by the automaton $\mathcal{A}$
  if and only if there exists a sequence of states in S, namely $S_1, S_2, \ldots, S_n$,
  and there exists $S_f \in F$ such that:
  - $S_0 \xrightarrow{\omega_1} S_1 \xrightarrow{\omega_2} S_2 \ldots S_{n-1} \xrightarrow{\omega_n} S_n$ **(successful path)**
  - $S_0 \xrightarrow{\omega} S_f$

Accepted word

- A word is **accepted** by a finite automaton **if and only if** the word transitions the automaton from the **initial state** to the **final state**.
- If we have the automaton s_0, then **ε is accepted** by the automaton (this is the **empty path**).
- **Example:** In the **previous example**:



  - Is $\omega = bab$ in $L(\mathcal{A})$?
  - $S_0 \xrightarrow{b} S_1 \xrightarrow{a} S_2 \xrightarrow{b}$ **(blocked ⇒ not recognized by the automaton $\mathcal{A}$).**

Language Recognized by a Finite Automaton

$$L(\mathcal{A}) = \left\{ \omega \in X^*, \exists S_f \in F \text{ such that } S_0 \xrightarrow{\omega} S_f \right\}$$

Accessible Co-accessible states

- **Definition:** Let $\mathcal{A}$ < X, S, S_0, F, $\mathbb{I}$ > be a finite automaton.

  A state $\mathbf{S_i}$ is said to be **accessible** if and only if

$$\exists \omega \in X^* \text{ such that } S_0 \xrightarrow{\omega} S_i.$$

- **Definition:** Let $\mathcal{A}$ < X, S, S_0, F, $\mathbb{I}$ > be a finite automaton.

  A state $\mathbf{S_i}$ is said to be **co-accessible** if and only if

$$\exists \omega \in X^*, \exists S_f \in F \text{ such that } S_i \xrightarrow{\omega} S_f.$$

Proposition:

Let $\boldsymbol{\mathcal{A}} < \boldsymbol{X}, \boldsymbol{S}, \boldsymbol{S_0}, \mathbb{F}, \mathbb{I} >$ be a finite automaton, and let $\boldsymbol{\mathcal{A}'}$ be the automaton obtained from $\boldsymbol{\mathcal{A}}$ by removing **non-accessible** and **non-co-accessible** states, along with their incoming and outgoing transitions.

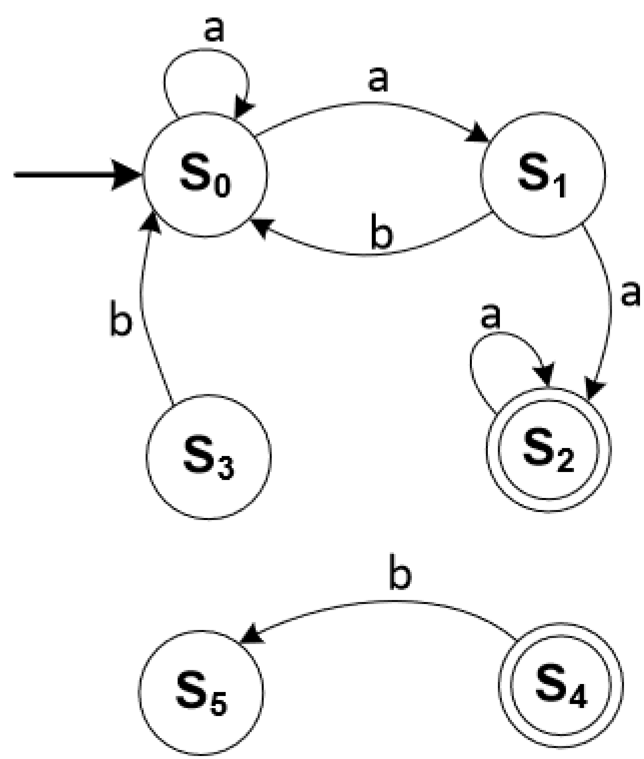
We have $\boldsymbol{L(\mathcal{A}) = L(\mathcal{A}')}$.

State

Example

États accessibles	a	b
S_0	S_0	S_1
S_1	S_2	S_0
S_2	S_2	-

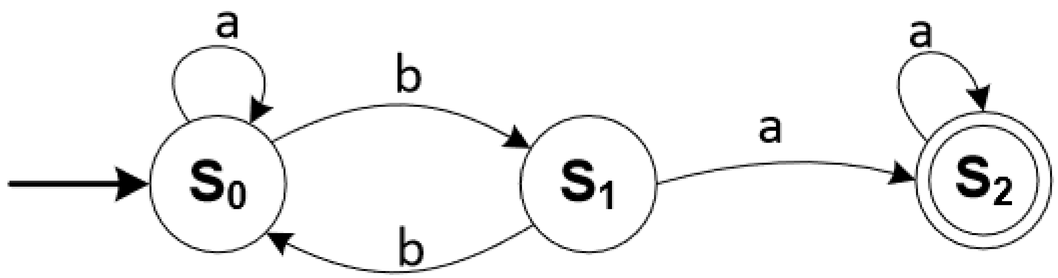
États coaccessibles	a	b
S_0	S_0	S_1, S_3
S_2	S_1, S_2	-
S_4	-	-
S_3	-	-
S_1	-	-



accessibles S = $\{S_0, S_1, S_2\}$

co-accessibles S = $\{S_0, S_1, S_2, S_3, S_4\}$

$$\mathcal{A}' \rightsquigarrow \{S_0, S_1, S_2\}$$



$$L(\mathcal{A}) = L(\mathcal{A}')$$

Complete Automaton

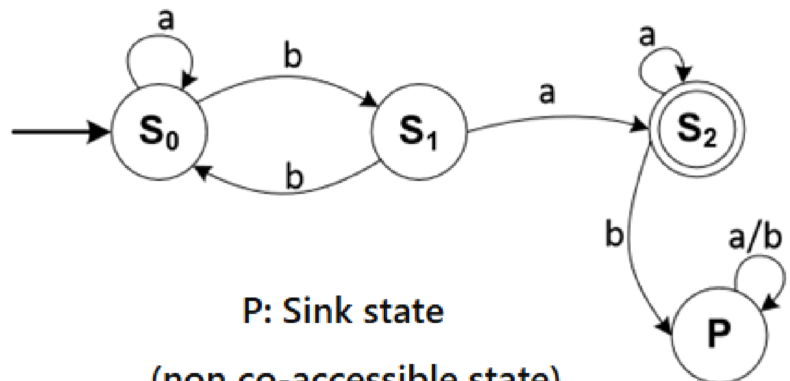
- **Definition:**
  A finite automaton is said to be **complete** if and only if

$$\forall\, S_i \in S, \forall\, x_i \in X\ \exists\, S_j \in S \text{ tel que } S_i \xrightarrow{x_i} S_j$$

baba $\in L(\mathcal{A})$?

$$S_0 \xrightarrow{b} S_1 \xrightarrow{a} S_2 \xrightarrow{b} P$$

baba $\notin L(\mathcal{A})$



Proposition: For every automaton $\boldsymbol{\mathcal{A} < X, S, S_0, \mathbb{F}, \mathbb{I} >}$, there exists an equivalent **complete automaton** $\boldsymbol{\mathcal{A}' < X, S', S_0', \mathbb{F}', \mathbb{I}' >}$ such that $\boldsymbol{L(\mathcal{A}) = L(\mathcal{A}')}$.

Deterministic Automaton

- **A** is said to be **deterministic** if and only if the transition function **δ** associates at most one state **p** to each pair **(q, a)**, where **a ∈ X** and **p, q ∈ Q**.

Deterministic Automaton

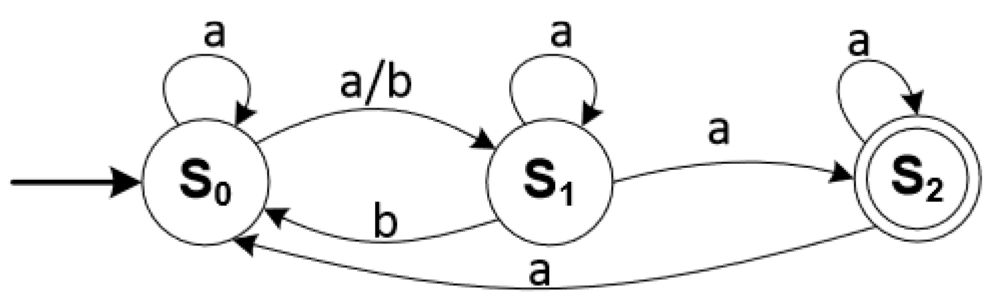
Ⅱ: S × X → S

1. $\forall\ S_i, S_j, S_k \in S, x_i \in X$

   If $S_i \xrightarrow{x_i} S_j$ and $S_i \xrightarrow{x_i} S_k$ then $S_j = S_k$.

2. The initial state is unique.

- **Example**



Non-deterministic Automaton

- $\omega = aaba \in L\,(\mathcal{A})$?

1) $S_0 \xrightarrow{a} S_1 \xrightarrow{a} S_1 \xrightarrow{b} S_0 \xrightarrow{a} S_0 \notin \mathbb{F}$

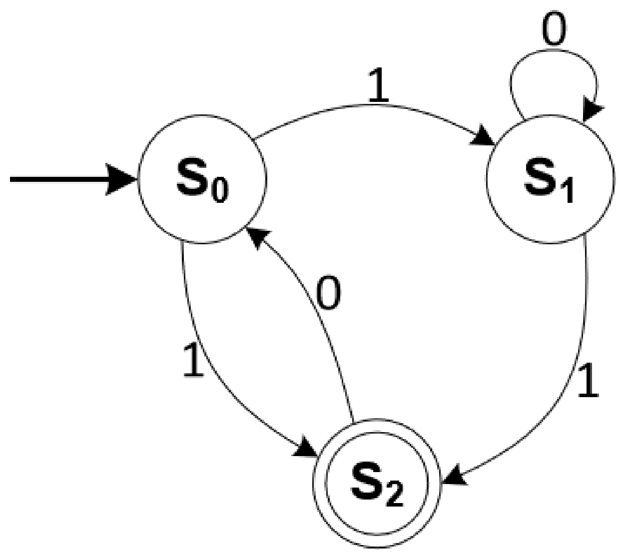
2) $S_0 \xrightarrow{a} S_1 \xrightarrow{a} S_1 \xrightarrow{b} S_0 \xrightarrow{a} S_1 \notin \mathbb{F}$

3) ....

4) $S_0 \xrightarrow{a} S_0 \xrightarrow{a} S_0 \xrightarrow{b} S_1 \xrightarrow{a} S_2 \in \mathbb{F}$

- **Proposition:** For every non-deterministic automaton $\boldsymbol{\mathcal{A} < X, S, S_0, \mathbb{F}, \mathbb{I} >}$, there exists an equivalent deterministic automaton $\boldsymbol{\mathcal{A}'}$ such that $\mathbf{L(\mathcal{A}) = L(\mathcal{A}')}$.

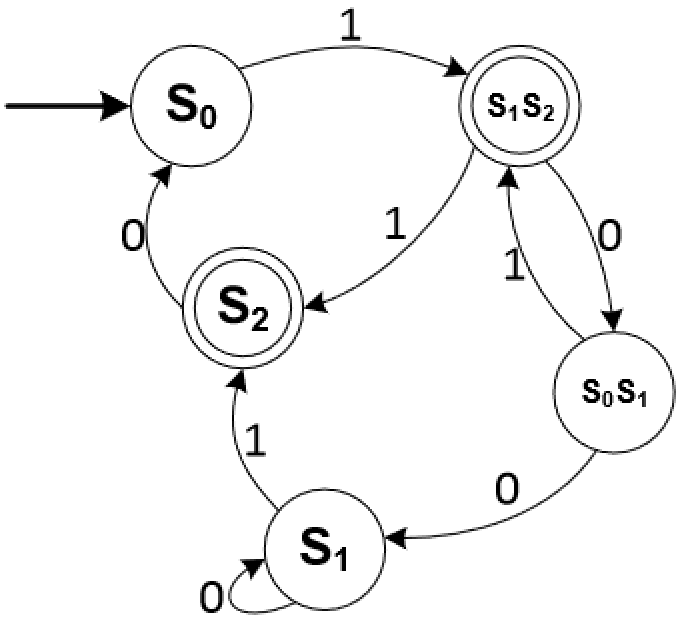
Example



- $\omega = 1011 \in L(\mathcal{A})$?
- It is enough to find a successful path that processes this word.

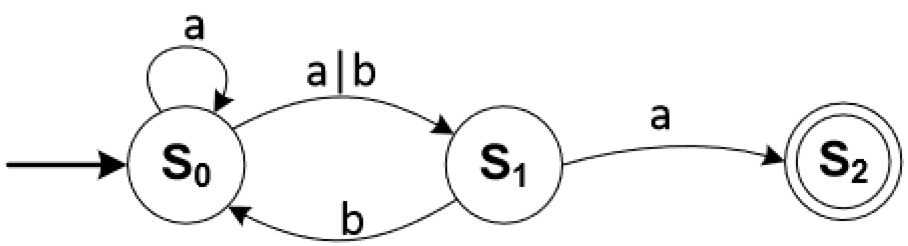
Deterministic Automaton

	0	1
→ S_0	-	S_1S_2
# S_1S_2	S_0S_1	S_2
S_0S_1	S_1	S_1S_2
# S_2	S_0	-
S_1	S_1	S_2

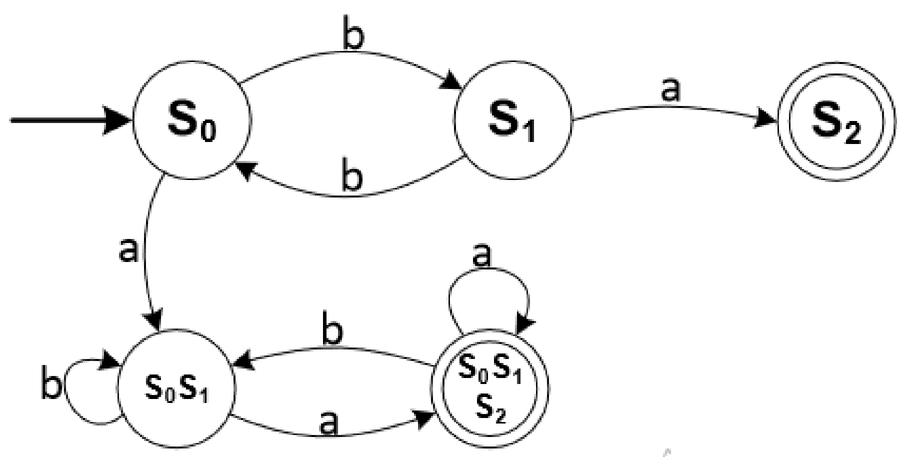


Example

- Make the automaton $\mathcal{A}$ deterministic and then complete it.

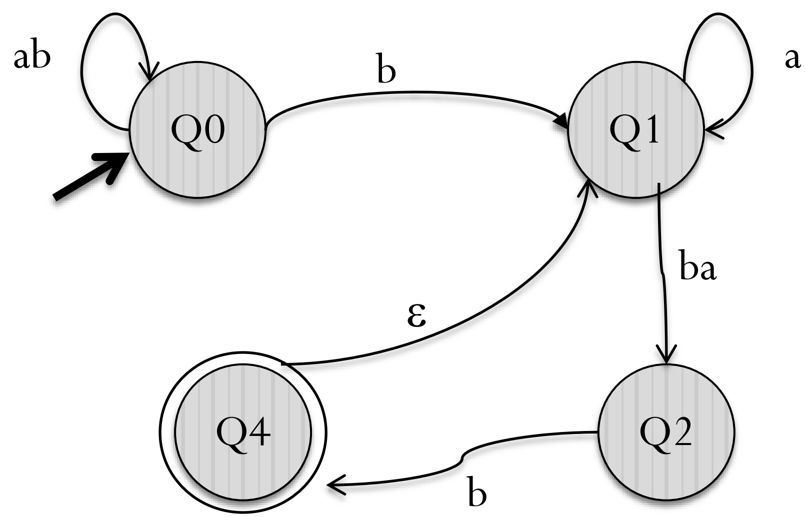


	a	b
→ S_0	S_0S_1	S_1
S_0S_1	$S_0S_1S_2$	S_0S_1
S_1	S_2	S_0
# $S_0S_1S_2$	$S_0S_1S_2$	S_0S_1
# S_2	-	-



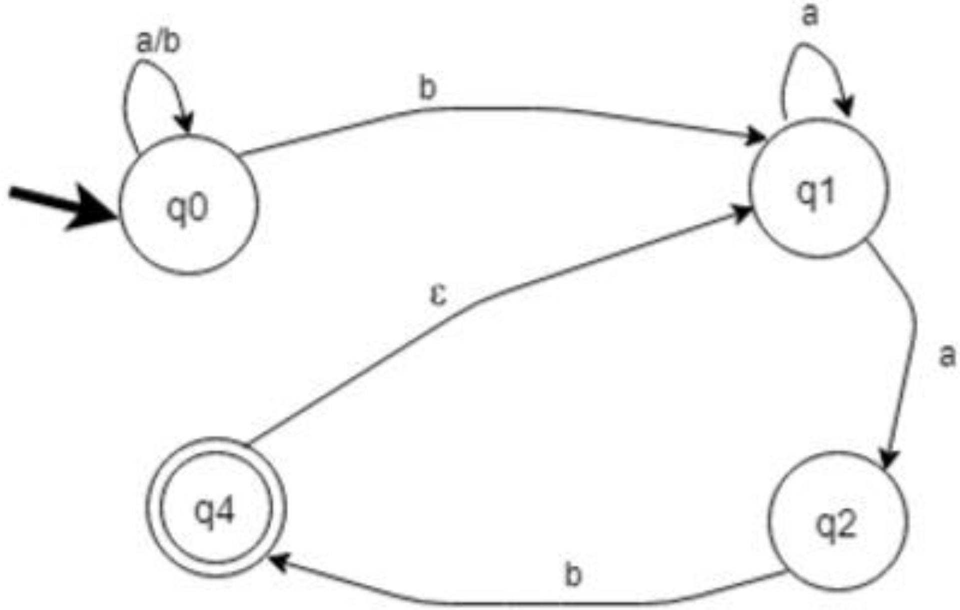
Generalized Automata

- In a generalized finite state automaton , transitions (labels) can be generated by words. Transitions caused by the word ε are called spontaneous transitions (ε-transitions), which represent a state change without reading any input..



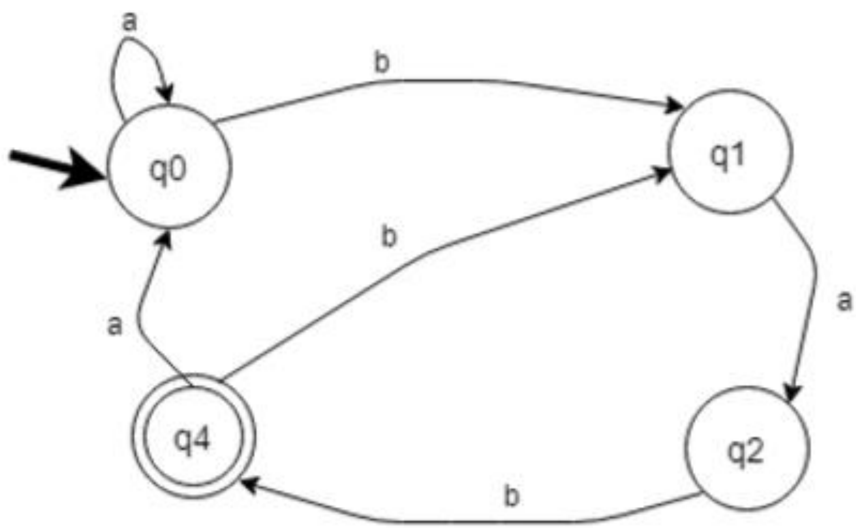
Partially Generalized Automaton

In a partially generalized finite state automaton (AEF), transitions (labels) are generated by a single symbol from the alphabet or by ε.



Simple Automata

- In a simple AEF, transitions (labels) are always generated by one and only one symbol of the alphabet.



Generalized Finite State Automata

- **Definition:**

  A **generalized automaton** is a **5-tuple** $\mathcal{AG} < X*, S, S_0, \mathbb{F}, \mathbb{I} >$, where $\mathbb{I} \subseteq S \times X* \times S$.

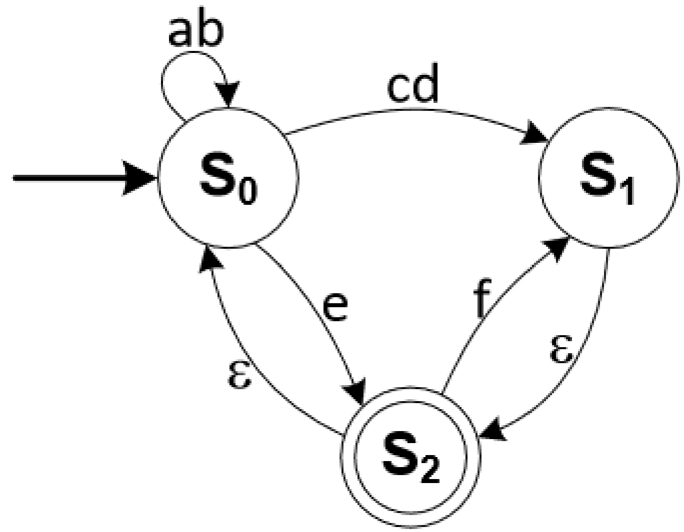
- The transitions in a generalized automaton are of three types:

– Transitions caused by letters of **X**.

– Transitions caused by words of length **>1**.

– Transitions caused by the empty word (**spontaneous transition**).

Example



- $\omega = \text{abcd} \in L(\mathcal{A})$?
- $S_0 \xrightarrow{ab} S_0 \xrightarrow{cd} S_1 \xrightarrow{\epsilon} S_2 \in \mathbb{F} \Rightarrow S_0 \xrightarrow{abcd} S_2 \Rightarrow \text{abcd} \in L(\mathcal{A})$

Generalized Finite State Automata

- **Theorem:**

For each **generalized finite automaton (AEF)** $\mathcal{AG} < X, S, S_0, \mathbb{F}, \mathbb{I} >$, there exists a **simple and deterministic finite automaton (AEF)** $\mathcal{AS} < X, SS, S_0S, \mathbb{F}S, \mathbb{I}S >$ that recognizes the same language.

Construction of the partially generalized automaton

- **Elimination of words containing at least two letters:**
  For each transition with a word **ω**, such that |**ω**| = **n**, with **n** ≥ **2**, create **n-1** additional states and add transitions that connect these states with the letters of **ω**. At the end of this step, we obtain a **partially generalized finite automaton (AEF).**

Construction of the partially generalized automaton

Construction of the Partially Generalized Automaton:

- $\mathcal{A}PG < X \cup \{\epsilon\}, S', S0', \mathbb{F}, \mathbb{I}' >$
- At initialization: $\mathbb{I}' = S' = S$

For each instruction of $\mathbb{I}$ (S_i, x, S_j) where $x \in X^*$

- If $|x| = 0$ or $|x| = 1$, then $(S_i, x, S_j) \in \mathbb{I}'$
- If $|x| > 1$, then $(x = x_1 x_2 \ldots x_n, x_i \in X)$

  Add to $\mathbb{I}'$:

  (S_i, x_1, S_{I1})

  (S_{I1}, x_2, S_{I2})

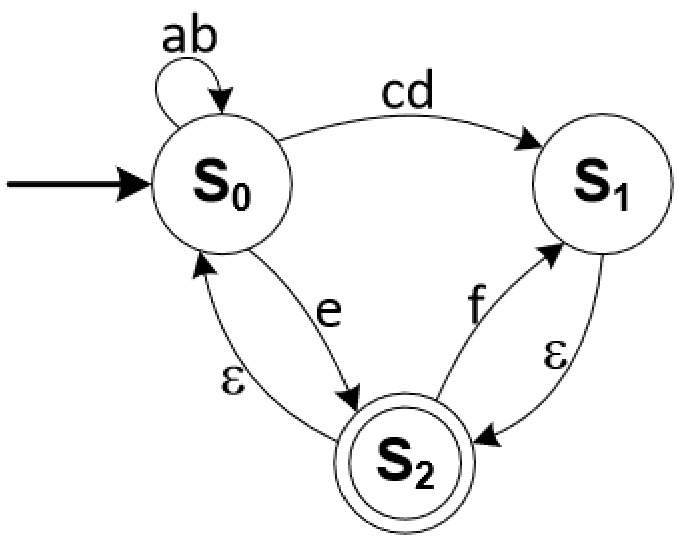
  ......

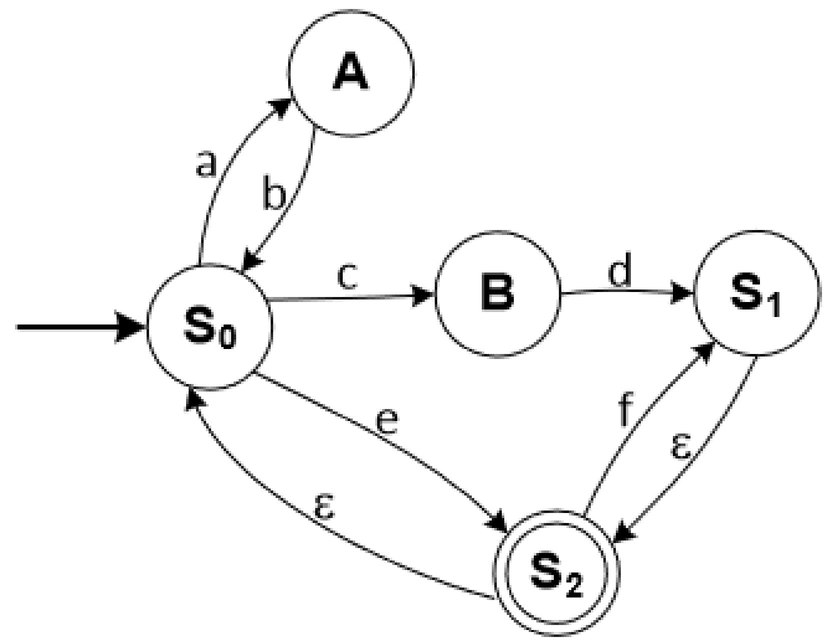
  $(S_{I_{n-1}}, x_n, S_j)$

  We add $(n-1)$ intermediate states $S_{I1}, S_{I2}, \ldots, S_{I_{n-1}}$ to $S' \Rightarrow$

  $S' = S \cup \{S_{I1}, S_{I2}, \ldots, S_{I_{n-1}}\}$

Construction of the partially generalized automaton

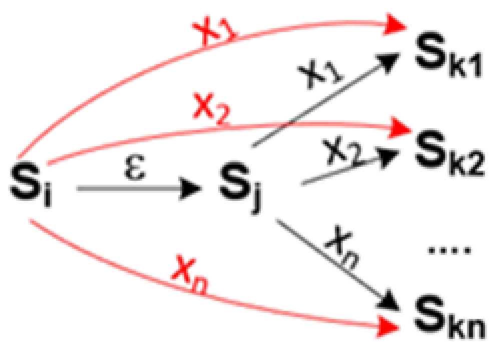




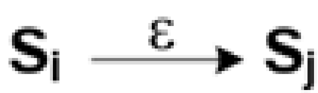
Construction of the simple automaton

- **Elimination of ε-transitions:**
  The removal of these transitions gives us a simple DFA. To achieve this, we must first eliminate the transitions by ε

Rule 1

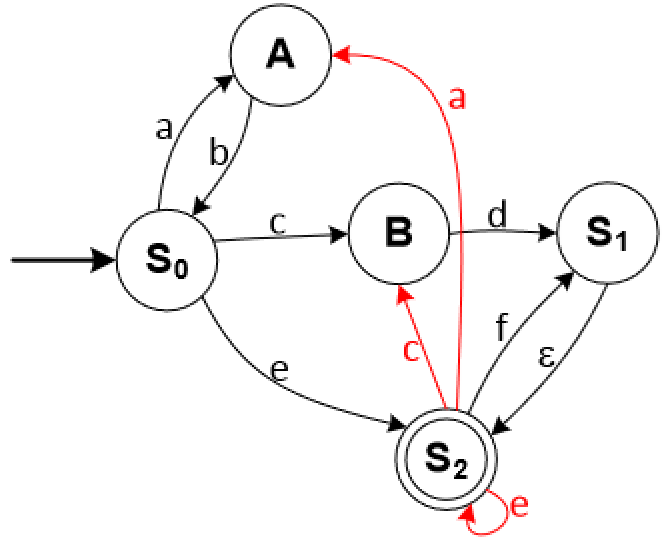


Rule 2



If $S_j \in \mathfrak{F}$ then S_i becomes a final state

Elimination of the first spontaneous transition



Elimination of the second spontaneous transition