# Extended Transition Function

## ✓ Extended transition function

An extended transition function $\hat{\delta}$ traces the path of an automaton and determines the final state when an initial state q and an input string x are passed through it.
The difference between a simple transition function and the extended transition function is that the former performs a transition of a single character/instance. In contrast, the latter performs the transitions on a complete string.

**A recursive algorithm is used to reach the final state, which is as follows:**

**Base condition:**

$\hat{\delta}$ (q,ϵ)→q

**Recursion rule:**

$\hat{\delta}$ (q,xa)→δ($\hat{\delta}$ (q,x),a)

Here, x∈Σ* and a∈Σ. Also, *x* is a string of characters belonging to the set of the input symbols and a*a* is a single character.

The input string is reduced from the right side, character by character, until the base condition—when all the strings are reduced and we are left with a null character (epsilon)—is reached. Then simple transitions are applied to the broken-down string.
A transition table is formed to show each transition in the DFA. If the output is a final state, the given string will be accepted by the DFA.

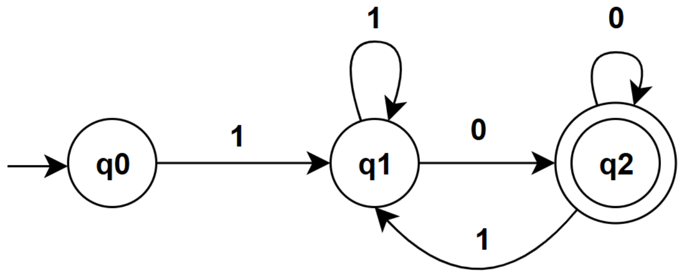
## Outlines:

Extended transition Function for DFA: example 1,2
Extended transition Function for NFA: example 1,2

# Extended Transition Function

## ❑ Extended Transition function for DFA:

**Example - 1: Construct a DFA to accept the string that start with 1 and end with 0, over Σ = {0,1} and check the string 1010 is accepted by DFA using extended transition function.**




**Transition function for DFA:**

| δ | 0 | 1 |
|---|---|---|
| →q0 | - | q1 |
| q1 | q2 | q1 |
| *q2 | q2 | q1 |

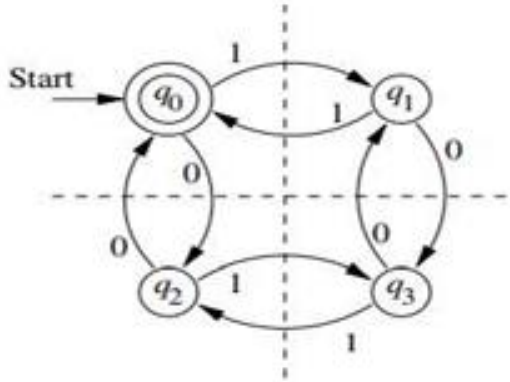
**Extended Transition function for DFA:**

$\hat{\delta}$ (q, ϵ) = q0
$\hat{\delta}$ (q0, 1) = δ ($\hat{\delta}$ (q0, ϵ), 1) = δ (q0, 1) = q1
$\hat{\delta}$ (q0, 10) = δ ($\hat{\delta}$ (q0, 1) , 0) = δ (q1, 0) = q2
$\hat{\delta}$ (q0, 101) = δ ($\hat{\delta}$ (q0, 10) , 1) = δ (q2, 1) = q1
$\hat{\delta}$ (q0, 1010) = δ ($\hat{\delta}$ (q0, 101) , 0) = δ (q1, 0) = q2

Since q2 is in Final state, 1010 is accepted by DFA

# Extended Transition Function

**Example - 2: check the string 110101 is accepted by DFA or not, using extended transition function.**




Transition function for DFA:

| δ | 0 | 1 |
|---|---|---|
| →*q0 | q2 | q1 |
| q1 | q3 | q0 |
| q2 | q0 | q3 |
| q3 | q1 | q2 |

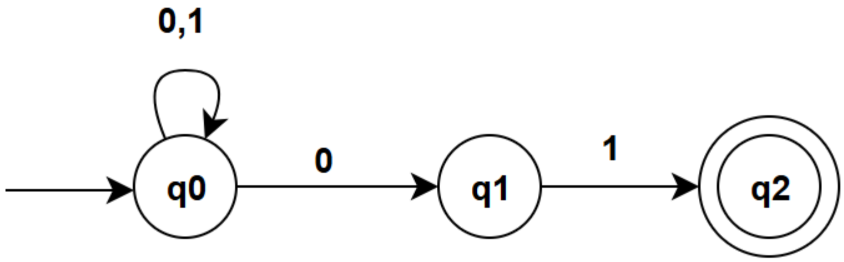
The check involves computing $\hat{\delta}(q_0, w)$ for each prefix $w$ of 110101, starting at $\epsilon$ and going in increasing size. The summary of this calculation is:

- $\hat{\delta}(q_0, \epsilon) = q_0$.
- $\hat{\delta}(q_0, 1) = \delta\big(\hat{\delta}(q_0, \epsilon), 1\big) = \delta(q_0, 1) = q_1$.
- $\hat{\delta}(q_0, 11) = \delta\big(\hat{\delta}(q_0, 1), 1\big) = \delta(q_1, 1) = q_0$.
- $\hat{\delta}(q_0, 110) = \delta\big(\hat{\delta}(q_0, 11), 0\big) = \delta(q_0, 0) = q_2$.
- $\hat{\delta}(q_0, 1101) = \delta\big(\hat{\delta}(q_0, 110), 1\big) = \delta(q_2, 1) = q_3$.
- $\hat{\delta}(q_0, 11010) = \delta\big(\hat{\delta}(q_0, 1101), 0\big) = \delta(q_3, 0) = q_1$.
- $\hat{\delta}(q_0, 110101) = \delta\big(\hat{\delta}(q_0, 11010), 1\big) = \delta(q_1, 1) = q_0$.

# Extended Transition Function

## ❑ Extended Transition function for NFA:

**Example-1: check the NFA is accepted or not for the input 00101.**




Transition function for NFA:

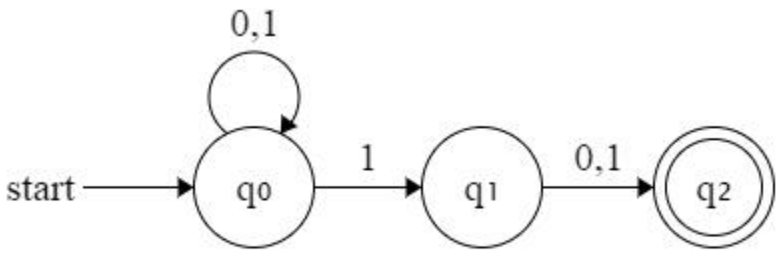
| δ | 0 | 1 |
|---|---|---|
| →q0 | {q0, q1} | q0 |
| q1 | - | q2 |
| *q2 | - | - |

1. $\hat{\delta}(q_0, \epsilon) = \{q_0\}$.
2. $\hat{\delta}(q_0, 0) = \delta(q_0, 0) = \{q_0, q_1\}$.
3. $\hat{\delta}(q_0, 00) = \delta(q_0, 0) \cup \delta(q_1, 0) = \{q_0, q_1\} \cup \emptyset = \{q_0, q_1\}$.
4. $\hat{\delta}(q_0, 001) = \delta(q_0, 1) \cup \delta(q_1, 1) = \{q_0\} \cup \{q_2\} = \{q_0, q_2\}$.
5. $\hat{\delta}(q_0, 0010) = \delta(q_0, 0) \cup \delta(q_2, 0) = \{q_0, q_1\} \cup \emptyset = \{q_0, q_1\}$.
6. $\hat{\delta}(q_0, 00101) = \delta(q_0, 1) \cup \delta(q_1, 1) = \{q_0\} \cup \{q_2\} = \{q_0, q_2\}$.

Line (1) is the basis rule. We obtain line (2) by applying $\delta$ to the lone state, $q_0$, that is in the previous set, and get $\{q_0, q_1\}$ as a result. Line (3) is obtained by taking the union over the two states in the previous set of what we get when we apply $\delta$ to them with input 0. That is, $\delta(q_0, 0) = \{q_0, q_1\}$, while $\delta(q_1, 0) = \emptyset$. For line (4), we take the union of $\delta(q_0, 1) = \{q_0\}$ and $\delta(q_1, 1) = \{q_2\}$. Lines (5) and (6) are similar to lines (3) and (4). □

# Extended Transition Function

**Example-2: Check w1 = 001 and w2 = 01010 is accepted by the NFA using extended transition function or not.**




Transition Table:

| $\delta$ | 0 | 1 |
|---|---|---|
| $\rightarrow q_0$ | $\{q_0\}$ | $\{q_0, q_1\}$ |
| $q_1$ | $\{q_2\}$ | $\{q_2\}$ |
| $* \quad q_2$ | $\emptyset$ | $\emptyset$ |

Extended transition Function Input Processing:

For input, w = 011

$\hat{\delta}(q_0, \epsilon) = \{q_0\}$

$\hat{\delta}(q_0, 0) = \{q_0\}$

$\hat{\delta}(q_0, 01) = \delta(\hat{\delta}(q_0, 0), 1)$
$\ldots\ldots\ldots\ldots = \delta(q_0, 1)$
$\ldots\ldots\ldots\ldots = \{q_0, q_1\}$

$\hat{\delta}(q_0, 011) = \delta(\hat{\delta}(q_0, 01), 1)$
$\ldots\ldots\ldots\ldots = \delta(\{q_0, q_1\}, 1)$
$\ldots\ldots\ldots\ldots = \delta(q_0, 1) \cup \delta(q_1, 1)$
$\ldots\ldots\ldots\ldots = \{q_0, q_1\} \cup \{q_2\}$
$\ldots\ldots\ldots\ldots = \{q_0, q_1, q_2\}$

For input, w = 01010. Since sub-string 01 is already calculated I will reuse it.

$\hat{\delta}(q_0, 010) = \delta(\hat{\delta}(q_0, 01), 0)$
$\ldots\ldots\ldots\ldots = \delta(\{q_0, q_1\}, 0)$
$\ldots\ldots\ldots\ldots = \delta(q_0, 0) \cup \delta(q_1, 0)$
$\ldots\ldots\ldots\ldots = \{q_0\} \cup \{q_2\}$
$\ldots\ldots\ldots\ldots = \{q_0, q_2\}$

$\hat{\delta}(q_0, 0101) = \delta(\hat{\delta}(q_0, 010), 1)$
$\ldots\ldots\ldots\ldots\ldots = \delta(\{q_0, q_2\}, 1)$
$\ldots\ldots\ldots\ldots\ldots = \delta(q_0, 1) \cup \delta(q_2, 1)$
$\ldots\ldots\ldots\ldots\ldots = \{q_0, q_1\} \cup \emptyset$
$\ldots\ldots\ldots\ldots\ldots = \{q_0, q_1\}$

$\hat{\delta}(q_0, 01010) = \delta(\hat{\delta}(q_0, 0101), 0)$
$\ldots\ldots\ldots\ldots\ldots\ldots = \delta(\{q_0, q_1\}, 0)$
$\ldots\ldots\ldots\ldots\ldots\ldots = \delta(q_0, 0) \cup \delta(q_1, 0)$
$\ldots\ldots\ldots\ldots\ldots\ldots = \{q_0\} \cup \{q_2\}$
$\ldots\ldots\ldots\ldots\ldots\ldots = \{q_0, q_2\}$