

TRƯỜNG ĐẠI HỌC SƯ PHẠM KỸ THUẬT VĨNH LONG  
KHOA CÔNG NGHỆ THÔNG TIN



## BÁO CÁO TIỂU LUẬN

**Môn: Lý Thuyết Ngôn Ngữ Hình Thức Và Tính Toán**

**Đề Tài:** Tìm hiểu về Automata hữu hạn

- a. Tìm ngôn ngữ được biểu diễn bởi FA
- b. Tìm NFA từ NFA-e. Xây dựng hệ thống

**Sinh viên thực hiện:**

1. Nguyễn Bá Anh Hào **MSSV:** 18004038
2. Nguyễn Phạm Anh Thư **MSSV:** 18004131
3. Trương Thành Khang **MSSV:** 18004054

**Lớp** ĐH CNTT 2018 Khóa : 43

**Giảng viên hướng dẫn :** Ths. Nguyễn Thị Hồng Yến

## **LỜI CẢM ƠN**

Chúng em xin chân thành cảm ơn cô Nguyễn Thị Hồng Yến đã giúp đỡ và hướng dẫn chúng em tận tình trong suốt thời gian viết bài tiểu luận, tạo cho chúng em những tiền đề, những kiến thức để tiếp cận vấn đề, phân tích giải quyết vấn đề. Nhờ đó mà chúng em hoàn thành bài luận của mình được tốt hơn. Chúng em cũng xin cảm ơn bạn bè, anh chị đã tận tình chỉ bảo, giúp đỡ chúng em trong quá trình hoàn thành bài tiểu luận, tạo cho chúng em hiểu thêm về những kiến thức thực tế.

Những kiến thức mà chúng em được học hỏi là hành trang ban đầu cho quá trình làm việc của chúng em sau này. Chúng em xin gửi tới mọi người lời chúc thành công trên con đường sự nghiệp của mình.

Vĩnh Long 3 tháng 7 năm 2020

**SINH VIÊN THỰC HIỆN :**

- Nguyễn Bá Anh Hào
- Nguyễn Phạm Anh Thư
- Trương Thành Khang

# MỤC LỤC

TÌM HIỂU VỀ ÔTÔMAT HỮU HẠN (FINITE AUTOMATA) .....	1
1. Automata hữu hạn đơn định .....	2
1.1 Giới thiệu .....	2
1.2 Hàm chuyển trạng thái mở rộng .....	3
1.3 Ngôn ngữ được chấp nhận bởi DFA .....	4
2. Ôtômat hữu hạn không đơn định .....	5
2.1 Giới thiệu .....	5
2.2 Hàm chuyển trạng thái mở rộng .....	6
3. Sự tương đương giữa DFA và NFA .....	6
4. NFA với $\epsilon$ -dịch chuyển (NFA $\epsilon$ ).....	7
5. Sự tương đương giữa NFA có và không có $\epsilon$ -dịch chuyển .....	10
6. Giải thuật xây dựng DFA từ NFA .....	12
XÂY DỰNG DFA ĐOÁN NHẬN NGÔN NGỮ .....	16

## TÌM HIỂU VỀ ÔTÔMAT HỮU HẠN (FINITE AUTOMATA)

Ôtômat hữu hạn FA là một mô hình tính toán của hệ thống với sự mô tả bởi các input và output. Tại mỗi thời điểm, hệ thống có thể được xác định ở một trong số hữu hạn các cấu hình nội bộ gọi là các trạng thái (states). Mỗi trạng thái của hệ thống thể hiện sự tóm tắt các thông tin liên quan đến những input đã chuyển qua và xác định các phép chuyển kế tiếp trên dãy input tiếp theo.

Trong khoa học máy tính, ta có thể tìm thấy nhiều ví dụ về hệ thống trạng thái hữu hạn, và lý thuyết về ôôtômat hữu hạn là một công cụ thiết kế hữu ích cho các hệ thống này. Chẳng hạn, một hệ chuyển mạch như bộ điều khiển (Control Unit) trong máy tính. Một chuyển mạch thì bao gồm một số hữu hạn các cổng (gate) input, mỗi cổng có 2 giá trị 0 hoặc 1. Các giá trị đầu vào này sẽ xác định 2 mức điện thế khác nhau ở cổng output. Mỗi trạng thái của một mạng chuyển mạch với  $n$  cổng bất kỳ sẽ là một trường hợp trong  $2^n$  phép gán của 0 và 1 đối với các cổng khác nhau. Các chuyển mạch thì được thiết kế theo cách này, vì thế chúng có thể được xem như hệ thống trạng thái hữu hạn. Các chương trình sử dụng thông thường, chẳng hạn trình soạn thảo văn bản hay bộ phân tích từ vựng trong trình biên dịch máy tính cũng được thiết kế như các hệ thống trạng thái hữu hạn. Ví dụ bộ phân tích từ vựng sẽ quét qua tất cả các dòng ký tự của chương trình máy tính để tìm nhóm các chuỗi ký tự tương ứng với một tên biến, hằng số, từ khóa, ... Trong quá trình xử lý này, bộ phân tích từ vựng cần phải nhớ một số hữu hạn thông tin như các ký tự bắt đầu hình thành những chuỗi từ khóa. Lý thuyết về ôôtômat hữu hạn thường được dùng đến nhiều cho việc thiết kế các công cụ xử lý chuỗi hiệu quả.

Máy tính cũng có thể được xem như một hệ thống trạng thái hữu hạn. Trạng thái hiện thời của bộ xử lý trung tâm, bộ nhớ trong và các thiết bị lưu trữ phụ ở mỗi thời điểm bất kỳ là một trong những số rất lớn và hữu hạn của số trạng thái. Bộ não con người cũng là một hệ thống trạng thái hữu hạn, vì số các tế bào thần kinh hay gọi là neurons là số có giới hạn, nhiều nhất có thể là  $2^{35}$ .

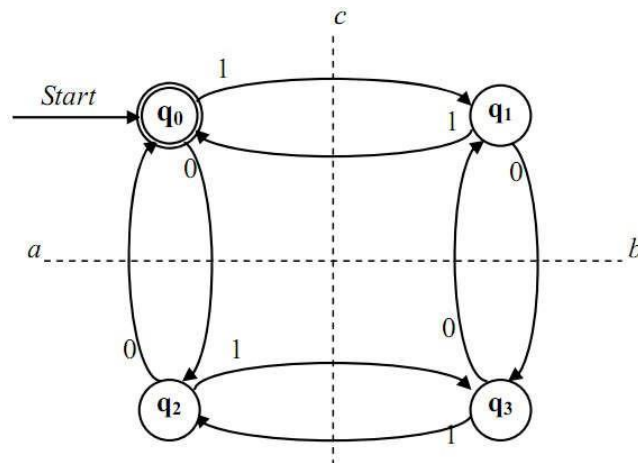
Lý do quan trọng nhất cho việc nghiên cứu các hệ thống trạng thái hữu hạn là tính tự nhiên của khái niệm và khả năng ứng dụng đa dạng trong nhiều lĩnh vực thực tế. Ôôtômat hữu hạn (FA) được chia thành 2 loại: đơn định (DFA) và không đơn định (NFA). Cả hai loại ôôtômat hữu hạn đều có khả năng nhận dạng chính xác tập chính quy. Ôôtômat hữu hạn đơn định có khả năng nhận dạng ngôn ngữ dễ dàng hơn ôôtômat hữu hạn không đơn định, nhưng thay vào đó thông thường kích thước của nó lại lớn hơn so với ôôtômat hữu

hạn không đơn định tương đương.

Một ô tô máy hữu hạn đơn định (DFA) - gọi tắt là FA - gồm một tập hữu hạn các trạng thái và một tập các phép chuyển từ trạng thái này tới trạng thái khác trên các ký hiệu nhập (input symbols) được chọn từ một bộ chữ cái  $\Sigma$  nào đó. Mỗi ký hiệu nhập có đúng một phép chuyển khỏi mỗi trạng thái (có thể chuyển trở về chính nó). Một trạng thái, thường ký hiệu là  $q_0$ , gọi là trạng thái bắt đầu (trạng thái ô tô máy bắt đầu). Một số trạng thái được thiết kế như là các trạng thái kết thúc hay trạng thái chấp nhận.

Một đồ thị có hướng, gọi là sơ đồ chuyển (transition diagram) tương ứng với một DFA như sau: các đỉnh của đồ thị là các trạng thái của DFA; nếu có một đường chuyển từ trạng thái  $q$  đến trạng thái  $p$  trên input  $a$  thì có một cung nhãn  $a$  chuyển từ trạng thái  $q$  đến trạng thái  $p$  trong sơ đồ chuyển. DFA chấp nhận một chuỗi  $x$  nếu như tồn tại dãy các phép chuyển tương ứng trên mỗi ký hiệu của  $x$  dẫn từ trạng thái bắt đầu đến một trong những trạng thái kết thúc.

Chẳng hạn, sơ đồ chuyển của một DFA được mô tả trong hình 3.1. Trạng thái khởi đầu  $q_0$  được chỉ bằng mũi tên có nhãn "Start". Chỉ có duy nhất một trạng thái kết thúc, cũng là  $q_0$  trong trường hợp này, được chỉ ra bằng hai vòng tròn. Ô tô máy này chấp nhận tất cả các chuỗi số 0 và số 1 với số số 0 và số số 1 là số chẵn.



Hình 1: Sơ đồ chuyển của một DFA

Một điều cần lưu ý, DFA sử dụng mỗi trạng thái của nó để giữ chỉ một phần của chuỗi số 0 và 1 chứ không phải chứa một số thực sự, vì thế DFA cần dùng một số hữu hạn trạng thái.

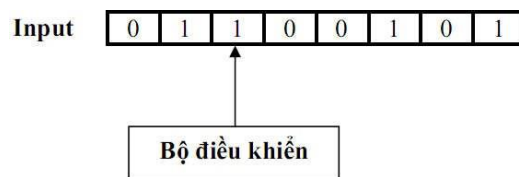
## 1. Automata hữu hạn đơn định

### 1.1 Giới thiệu

Một cách hình thức ta định nghĩa ô tô máy hữu hạn là bộ gồm năm thành phần  $(Q, \Sigma, \delta, q_0, F)$ , trong đó:

- $Q$  là tập hợp hữu hạn các trạng thái.
- $\Sigma$  là bộ chữ cái nhập hữu hạn.
- $\delta$  là hàm chuyển ánh xạ từ  $Q \times \Sigma \rightarrow Q$ , tức là  $\delta(q, a)$  là một trạng thái được cho bởi phép chuyển từ trạng thái  $q$  trên ký hiệu nhập  $a$ .
- $q_0 \in Q$  là trạng thái bắt đầu
- $F \subseteq Q$  là tập các trạng thái kết thúc.

Ta vẽ DFA như là bộ điều khiển hữu hạn, với mỗi trạng thái thuộc  $Q$ , DFA đọc một chuỗi các ký hiệu  $a$  từ  $\Sigma$  viết trên băng (như hình Hình 2)



Hình 2: Mô tả một DFA

Trong một lần chuyển, DFA đang ở trạng thái  $q$  đọc ký hiệu nhập  $a$  trên băng, chuyển sang trạng thái được xác định bởi hàm chuyển  $\delta(q, a)$ , rồi dịch đầu đọc sang phải một ký tự. Nếu  $\delta(q, a)$  chuyển đến một trong những trạng thái kết thúc thì DFA chấp nhận chuỗi được viết trên băng input phía trước đầu đọc, nhưng không bao gồm ký tự tại vị trí đầu đọc vừa dịch chuyển đến. Trong trường hợp đầu đọc đã dịch đến cuối chuỗi trên băng, thì DFA mới chấp nhận toàn bộ chuỗi trên băng.

### 1.2 Hàm chuyển trạng thái mở rộng

Để có thể mô tả một cách hình thức hoạt động của một DFA trên chuỗi, ta mở rộng hàm chuyển  $\delta$  để áp dụng đối với một trạng thái trên chuỗi hơn là một trạng thái trên từng ký hiệu. Ta định nghĩa hàm chuyển  $\delta$  như một ánh xạ từ  $Q \times \Sigma^* \rightarrow Q$  với ý nghĩa  $\delta(q, w)$  là trạng thái DFA chuyển đến từ trạng thái  $q$  trên chuỗi  $w$ . Một cách hình thức, ta định nghĩa:

1.  $\delta(q, \epsilon) = q$
2.  $\delta(q, wa) = \delta(\delta(q, w), a)$ , với mọi chuỗi  $w$  và ký hiệu nhập  $a$ .

Một số quy ước về ký hiệu:

- $Q$  là tập các trạng thái. Ký hiệu  $q$  và  $p$  (có hoặc không có chỉ số) là các trạng thái,  $q_0$  là trạng thái bắt đầu.
- $\Sigma$  là bộ chữ cái nhập. Ký hiệu  $a, b$  (có hoặc không có chỉ số) và các chữ số là các ký hiệu nhập.
- $\delta$  là hàm chuyển.

- $F$  là tập các trạng thái kết thúc.
- $w, x, y$  và  $z$  (có hoặc không có chỉ số) là các chuỗi ký hiệu nhập.

### 1.3 Ngôn ngữ được chấp nhận bởi DFA

Một chuỗi  $w$  được chấp nhận bởi ô tô máy hữu hạn  $M (Q, \Sigma, \delta, q_0, F)$  nếu  $\delta(q_0, w) = p$  với  $p \in F$ . Ngôn ngữ được chấp nhận bởi  $M$ , ký hiệu  $L(M)$  là tập hợp:

$$L(M) = \{ w \mid \delta(q_0, w) \in F \}$$

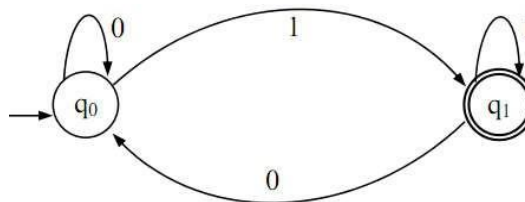
Ví dụ: Xây dựng DFA đoán nhận ngôn ngữ:

$L = \{w \mid w \text{ chỉ chứa ký hiệu } 0, 1 \text{ và luôn kết thúc bởi ký hiệu } 1\}$  DFA phải phân biệt 2 khả năng:

- 1) Khi đọc ký hiệu 1, thì ngay sau đó có thể kết thúc
- 2) Khi đọc ký hiệu 0, thì ngay sau đó không được kết thúc, mà phải đọc tiếp các ký hiệu khác cho đến khi gặp ký hiệu 1

Như thế, DFA sẽ có ít nhất 2 trạng thái tương ứng với hai khả năng trên. Ta xây dựng DFA như sau:

$M = (\{q_0, q_1\}, \{0, 1\}, d, q_0, \{q_1\})$ , trong đó hàm  $d$  như sau:



Hình 3: DFA

Hay chúng ta có thể biểu diễn hàm  $d$  bởi bảng dịch chuyển (Bảng 1)

Bảng 1

$\delta$	Inputs	
Trạng thái	0	1
$q_0$	$q_0$	$q_1$
$q_1$	$q_0$	$q_1$

Bây giờ, chúng ta sử dụng hàm dịch chuyển mở rộng để đoán nhận chuỗi  $w = 101101$  của ngôn ngữ  $L$ .

$$d(q_0, \epsilon) = q_0$$

$$d(q_0, 1) = d(d(q_0, \epsilon), 1) = d(q_0, 1) = q_1$$

$$d(q_0, 10) = d(d(q_0, 1), 0) = d(q_1, 0) = q_0$$

$$d(q_0, 101) = d(d(q_0, 10), 0) = d(q_0, 1) = q_1$$

$$d(q_0, 1011) = d(d(q_0, 101), 1) = d(q_1, 1) = q_1$$

$$d(q_0, 10110) = d(d(q_0, 1011), 0) = d(q_1, 0) = q_0$$

$$d(q_0, 101101) = d(d(q_0, 10110), 1) = d(q_0, 1) = q_1$$

Như thế, chuỗi vào  $w = 101101$  được đoán nhận bởi DFA. Chúng ta có thể hiểu quá trình đoán nhận đơn giản như sau:  $q_0101101 \rightarrow q_101101 \rightarrow q_01101 \rightarrow q_1101 \rightarrow q_101 \rightarrow q_01 \rightarrow q_01 \rightarrow q_1 \in F$

**Nhận xét:** Đối với một chuỗi vào  $w$ , DFA chỉ cho một quá trình đoán nhận duy nhất.

Giải thuật mô phỏng hoạt động của một DFA

```
. Input: Chuỗi nhập x kết thúc bởi $
. Output: Câu trả lời "YES" nếu DFA chấp nhận chuỗi x và
"NO" nếu ngược lại.
. Giải thuật:
    q := q0 ;
    c := nextchar ; {c là ký hiệu nhập được đọc tiếp theo}
    While c < > $ do
        begin
            q := δ(q, c) ;
```

**Nhận xét:** Một cách tổng quát, ta thấy tập  $Q$  của DFA thể hiện các trạng thái lưu trữ của ô tô máy trong quá trình đoán nhận ngôn ngữ, và như vậy khả năng lưu trữ của ô tô máy là hữu hạn. Mặt khác, hàm chuyển  $\delta$  là hàm toàn phần và đơn trị, cho nên các bước chuyển của ô tô máy luôn luôn được xác định một cách duy nhất. Chính vì hai đặc điểm này mà DFA mô tả như trên được gọi là ô tô máy hữu hạn đơn định.

## 2. Ô tô máy hữu hạn không đơn định

Một ô tô máy hữu hạn không đơn định có khả năng ở trong nhiều trạng thái khác nhau ở tại một thời điểm. Sự khác nhau cơ bản giữa DFA và NFA là định nghĩa của hàm dịch chuyển. Đối với NFA, hàm dịch chuyển nhận một trạng thái và một ký hiệu nhập và trả về một tập hợp (có thể rỗng) các trạng thái (thay vì trả về đúng 1 trạng thái đối với DFA).

### 2.1 Giới thiệu

Một cách hình thức ta định nghĩa ô tô máy hữu hạn không đơn định NFA là một bộ gồm 5 thành phần  $(Q, \Sigma, \delta, q_0, F)$  trong đó  $Q, \Sigma, q_0$  và  $F$  có ý nghĩa như trong DFA, nhưng  $\delta$  là hàm chuyển ánh xạ từ  $Q \times \Sigma \rightarrow 2^Q$ .

Khái niệm  $\delta(q, a)$  là tập hợp tất cả các trạng thái  $p$  sao cho có phép chuyển trên nhãn  $a$



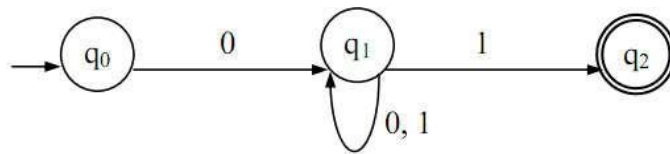
từ trạng thái  $q$  tới  $p$ .

## 2.2 Hàm chuyển trạng thái mở rộng

Để thuận tiện trong việc mô tả hoạt động ô tô máy trên chuỗi, ta mở rộng hàm chuyển  $\delta$  ánh xạ từ  $Q \times \Sigma^* \rightarrow 2^Q$  như sau:

- $\delta(q, \epsilon) = \{q\}$ 
  - $\delta(q, wa) = \{p \mid \text{có một trạng thái } r \text{ trong } \delta(q, w) \text{ mà } p \text{ thuộc } \delta(r, a)\}$
- $= \delta(\delta(q, w), a)$ 
  - $\delta(P, w) = \bigcup_{q \in P} \delta(q, w)$ , " $P \subset Q$ " Ngôn ngữ được chấp nhận bởi NFA
- Ngôn ngữ  $L(M)$ , với  $M$  là ô tô máy hữu hạn không đơn định NFA  $(Q, \Sigma, \delta, q_0, F)$  là tập hợp:  $L(M) = \{w \mid \delta(q_0, w) \text{ có chứa một trạng thái trong } F\}$

*Ví dụ:* Xây dựng NFA đoán nhận ngôn ngữ gồm các chuỗi trên bảng chữ cái  $\{0, 1\}$ , bắt đầu bởi ký hiệu 0 và kết thúc bởi ký hiệu 1. NFA được xây dựng như trong Hình 4.



Hình 4: NFA mô tả cho ví dụ

Giả sử chuỗi vào là  $w = 0101101$ , quá trình đoán nhận chuỗi  $w$  như sau:  
 $q_0 0101101 \rightarrow q_1 101101 \rightarrow q_1 01101 \rightarrow q_1 1101 \rightarrow q_1 101 \rightarrow q_1 01 \rightarrow q_1 1 \rightarrow q_2 \in F$ .

Tuy nhiên, chúng ta có thể dễ dàng nhận thấy rằng, quá trình đoán nhận trên chỉ là một trong những quá trình đoán nhận

*Nhận xét:* Đối với một chuỗi vào  $w$ , NFA có thể có nhiều quá trình đoán nhận khác nhau, trong khi DFA thì chỉ có một quá trình đoán nhận.

## 3. Sự tương đương giữa DFA và NFA

**Định lý:** Nếu  $L$  là tập được chấp nhận bởi một NFA thì tồn tại một DFA chấp nhận  $L$ .

Việc chứng minh định lý này được suy từ thuật toán đơn định hóa các ô tô máy.

*Ví dụ:* Cho NFA  $M(\{q_0, q_1\}, \{0, 1\}, \delta, q_0, \{q_1\})$  với hàm chuyển  $\delta$  như sau:

$$\delta(q_0, 0) = \{q_0, q_1\} \quad \delta(q_0, 1) = \{q_1\}$$

$$\delta(q_1, 0) = \emptyset \quad \delta(q_1, 1) = \{q_0, q_1\}$$

Ta xây dựng DFA tương đương  $M'(Q', \{0, 1\}, \delta', [q_0], F')$  chấp nhận ngôn ngữ  $L(M)$  như sau:

- $Q'$ : chứa tất cả các tập con của  $\{q_0, q_1\}$ , vậy  $Q' = \{\emptyset, [q_0], [q_1], [q_0, q_1]\}$
- Hàm chuyển  $\delta'$ :

Vì  $\delta(q_0, 0) = \{q_0, q_1\}$  nên  $\delta'([q_0], 0) = [q_0, q_1]$

Tương tự:  $\delta'([q_0], 1) = [q_1]$

$\delta'([q_1], 0) = \emptyset$

$\delta'([q_1], 1) = [q_0, q_1]$

Mặt khác:  $\delta'(\emptyset, 0) = \delta'(\emptyset, 1) = \emptyset$

Cuối cùng:  $\delta'([q_0, q_1], 0) = [q_0, q_1]$

$$(\text{vì } \delta(\{q_0, q_1\}, 0) = \delta(q_0, 0) \cup \delta(q_1, 0) = \{q_0, q_1\} \cup \emptyset = \{q_0, q_1\})$$

$\delta'([q_0, q_1], 1) = [q_0, q_1]$

$$(\text{vì } \delta(\{q_0, q_1\}, 1) = \delta(q_0, 1) \cup \delta(q_1, 1) = \{q_1\} \cup \{q_0, q_1\} = \{q_0, q_1\})$$

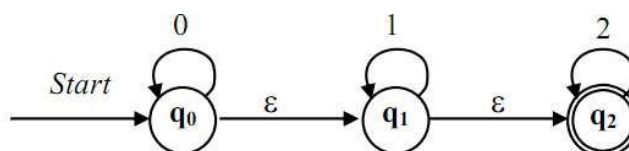
- Tập trạng thái kết thúc  $F' = \{[q_1], [q_0, q_1]\}$

Thực tế, có rất nhiều trạng thái của NFA không có hàm chuyển đến từ trạng thái bắt đầu  $[q_0]$ . Do đó, thông thường, cách tốt nhất là ta nên xây dựng DFA tương đương bắt đầu từ trạng thái  $[q_0]$  và thêm vào các trạng thái mới cho DFA chỉ khi có các hàm chuyển từ một trạng thái đã được thêm vào trước đó.

#### 4. NFA với $\epsilon$ -dịch chuyển (NFA $\epsilon$ )

Ta mở rộng mô hình NFA cho phép các phép chuyển trên nhãn rỗng  $\epsilon$ . Sơ đồ chuyển sau đây của một NFA chấp nhận chuỗi gồm một số bất kỳ (có thể là 0) chữ số 0 sau đó là một số bất kỳ chữ số 1 và sau nữa là một số bất kỳ chữ số 2. Thông thường, ta nói NFA chấp nhận một chuỗi  $w$  nếu có đường truyền nhãn  $w$  từ trạng thái bắt đầu đến một trạng thái kết thúc. Chẳng hạn, chuỗi 002 được chấp nhận bởi đường truyền  $q_0, q_0, q_0, q_1, q_2, q_2$  với các cung nhãn 0, 0,  $\epsilon, \epsilon, 2$ .

Ví dụ 1: Sơ đồ chuyển của một NFA với  $\epsilon$ -dịch chuyển:



Hình 5: NFA với  $\epsilon$ -dịch chuyển

**Định nghĩa:** Một cách hình thức ta định nghĩa NFA với  $\epsilon$ -dịch chuyển là bộ 5 thành phần  $(Q, \Sigma, \delta, q_0, F)$  với tất cả các thành phần có ý nghĩa như trên, nhưng hàm chuyển  $\delta$  là ánh xạ từ  $Q \times (\Sigma \cup \{\epsilon\}) \rightarrow 2^Q$ .

Khái niệm  $\delta(q, a)$  gồm tất cả các trạng thái  $p$  sao cho có phép chuyển nhãn  $a$  từ  $q$  tới  $p$ , trong đó  $a$  là một ký hiệu thuộc  $\Sigma$  hoặc là  $\epsilon$ .

**Hàm chuyển trạng thái mở rộng:** Ta mở rộng hàm chuyển  $\delta$  thành hàm chuyển  $\delta^*$  ánh xạ từ  $Q \times \Sigma^* \rightarrow 2^Q$ .  $\delta^*(q, w)$  chứa tất cả các trạng thái  $p$  sao cho có thể đi từ  $q$  tới  $p$  theo đường đi nhãn  $w$  (có thể chứa cạnh nhãn  $\epsilon$ ).

Ta sử dụng  $\epsilon$ -CLOSURE( $q$ ) để xác định tập tất cả các đỉnh  $p$  sao cho có đường đi từ  $q$  tới  $p$  với nhãn  $\epsilon$ .

Ví dụ 2: Trong Hình 5,  $\epsilon$ -CLOSURE( $q_0$ ) =  $\{q_0, q_1, q_2\}$ .

Vì đường đi chỉ có một đỉnh  $q_0$  (không có cung trên đường đi) là đường đi từ  $q_0$  tới  $q_0$  có tất cả các cạnh nhãn là  $\epsilon$ . Đường đi  $q_0, q_1$  chỉ ra rằng  $q_1$  thuộc  $\epsilon$ -CLOSURE( $q_0$ ). Và đường đi  $q_0, q_1, q_2$  chỉ ra rằng  $q_2$  thuộc  $\epsilon$ -CLOSURE( $q_0$ ).

Đặt  $\epsilon$ -CLOSURE( $P$ ) =  $\cup_{q \in P} \epsilon$ -CLOSURE( $q$ ), trong đó  $P$  là một tập các trạng thái

và  $q$  là một trạng thái. Ta định nghĩa hàm  $\delta^*$  như sau:

1.  $\delta^*(q, \epsilon) = \epsilon$ -CLOSURE( $q$ )
2.  $\delta^*(q, wa) = \epsilon$ -CLOSURE( $P$ )

Trong đó tập  $P = \{p \mid \text{có } r \text{ trong } \delta^*(q, w) \text{ sao cho } p \in \delta(r, a)\}$ , " $w \in \Sigma^*$  và  $a \in \Sigma$ " Hay  $\delta^*(q, wa) = \epsilon$ -CLOSURE( $\delta(\delta^*(q, w), a)$ )

Ta mở rộng  $\delta$  và  $\delta^*$  trên tập hợp các trạng thái  $R$  như sau:

3.  $\delta(R, a) = \cup_{q \in P} \delta(q, a)$
4.  $\delta^*(R, w) = \cup_{q \in P} \delta^*(q, w)$

*Nhận xét:*  $\delta^*(q, a)$  và  $\delta(q, a)$  không nhất thiết bằng nhau vì  $\delta^*(q, a)$  gồm tất cả các trạng thái có thể chuyển đến được từ  $q$  trên nhãn  $a$  gồm cả đường đi nhãn  $\epsilon$ , trong khi đó  $\delta(q, a)$  chỉ gồm các trạng thái có thể đến được từ  $q$  chỉ bằng các cung nhãn  $a$ . Tương tự  $\delta^*(q, \epsilon)$  có thể cũng không bằng  $\delta(q, \epsilon)$ . Vì vậy ta phải phân biệt ký hiệu  $\delta$  và  $\delta^*$  đối với NFA với  $\epsilon$ -dịch chuyển.

Ngôn ngữ được chấp nhận bởi NFA $\epsilon$ :

Ta định nghĩa  $L(M)$ , ngôn ngữ được chấp nhận bởi NFA $\epsilon$   $M = (Q, \Sigma, \delta, q_0, F)$  là tập hợp các chuỗi:

$$L(M) = \{w \mid \delta^*(q_0, w) \text{ có chứa ít nhất một trạng thái trong } F\}$$

Ví dụ 3: Xét sơ đồ chuyển của Hình 5.

Theo khái niệm hình thức, ta có NFA  $M (\{q_0, q_1, q_2\}, \{0, 1, 2\}, \delta, q_0, \{q_2\})$  với hàm

chuyển  $\delta$  như sau:

Bảng 2

$\delta$	Inputs			
Trạng thái	0	1	2	$\epsilon$
$q_0$	$\{q_0\}$	$\emptyset$	$\emptyset$	$\{q_1\}$
$q_1$	$\emptyset$	$\{q_1\}$	$\emptyset$	$\{q_1\}$
$q_2$	$\emptyset$	$\emptyset$	$\{q_2\}$	$\emptyset$

Xét chuỗi nhập  $w = 012$ . Ta cần tính  $\delta^*(q_0, 012)$

Ta có:  $\delta^*(q_0, \epsilon) = \epsilon\text{-CLOSURE}(q_0) = \{q_0, q_1, q_2\}$

vậy  $\delta^*(q_0, 0) = \epsilon\text{-CLOSURE}(\delta(\delta^*(q_0, \epsilon), 0))$

$= \epsilon\text{-CLOSURE}(\delta(\{q_0, q_1, q_2\}, 0))$

$= \epsilon\text{-CLOSURE}(\delta(q_0, 0) \cup \delta(q_1, 0) \cup \delta(q_2, 0))$

$= \epsilon\text{-CLOSURE}(\{q_0\} \cup \emptyset \cup \emptyset)$

$= \epsilon\text{-CLOSURE}(\{q_0\}) = \{q_0, q_1, q_2\}$

và  $\delta^*(q_0, 01) = \epsilon\text{-CLOSURE}(\delta(\delta^*(q_0, 0), 1))$

$= \epsilon\text{-CLOSURE}(\delta(\{q_0, q_1, q_2\}, 1))$

$= \epsilon\text{-CLOSURE}(\delta(q_0, 1) \cup \delta(q_1, 1) \cup \delta(q_2, 1))$

$= \epsilon\text{-CLOSURE}(\emptyset \cup \{q_1\} \cup \emptyset)$

$= \epsilon\text{-CLOSURE}(\{q_1\}) = \{q_1, q_2\}$

$\rightarrow \delta^*(q_0, 012) = \epsilon\text{-CLOSURE}(\delta(\delta^*(q_0, 01), 2))$

$= \epsilon\text{-CLOSURE}(\delta(\{q_1, q_2\}, 2))$

$= \epsilon\text{-CLOSURE}(\delta(q_1, 2) \cup \delta(q_2, 2))$

$= \epsilon\text{-CLOSURE}(\emptyset \cup \{q_2\})$

$= \epsilon\text{-CLOSURE}(\{q_2\}) = \{q_2\}$

Do  $\delta^*(q_0, 012)$  có chứa trạng thái  $q_2 \in F$  nên chuỗi  $w \in L(M)$ . Giải thuật mô phỏng hoạt động của một NFA $\epsilon$ :

```

. Input: Chuỗi nhập x được kết thúc bởi $.
. Output: Câu trả lời "YES" nếu NFA chấp nhận chuỗi x và
"NO" nếu ngược lại.
. Giải thuật:
    q :=  $\epsilon$ -CLOSURE( $q_0$ );
    c := nextchar ; {c là ký hiệu nhập được đọc tiếp theo}
    While c <> $ do
        begin
            q :=  $\epsilon$ -CLOSURE( $\delta(q, c)$ );
            c := nextchar ;
        end
    If q in F then write ("YES") else write ("NO");

```

### 5. Sự tương đương giữa NFA có và không có $\epsilon$ -dịch chuyển

Tương tự như NFA, khả năng có thể thực hiện phép chuyển trên nhãn  $\epsilon$  của NFA $\epsilon$  cũng không làm cho NFA $\epsilon$  chấp nhận được các tập hợp không chính quy. Ta có thể dẫn chứng điều này bằng cách mô phỏng hoạt động của một NFA $\epsilon$  bởi một NFA không có  $\epsilon$ -dịch chuyển.

**Định lý:** Nếu L được chấp nhận bởi một NFA có  $\epsilon$ -dịch chuyển thì L cũng được chấp nhận bởi một NFA không có  $\epsilon$ -dịch chuyển.

**Chứng minh:** Đặt  $M(Q, \Sigma, \delta, q_0, F)$  là NFA với  $\epsilon$ -dịch chuyển.

Ta xây dựng NFA  $M'(Q, \Sigma, \delta', q_0, F')$  tương đương không có  $\epsilon$ -dịch chuyển, trong đó:

- $F' = \begin{cases} F \cup \{q_0\} & \text{nếu } \epsilon - \text{CLOSURE}(q_0) \text{ chứa một trạng thái thuộc } F \\ F & \text{trong các trường hợp còn lại} \end{cases}$
- $\delta'(q, a)$  là  $\delta(q, a)$  với  $q \in Q$  và  $a \in \Sigma$ . Chú ý rằng  $M'$  không có  $\epsilon$ -dịch chuyển nên ta có thể dùng  $\delta'$  thay cho  $\delta^*$ , nhưng phải phân biệt  $\delta$  và  $\delta^*$ .

Ta chứng minh bằng quy nạp trên  $|x|$  rằng  $\delta'(q_0, x) = \delta^*(q_0, x)$ . Tuy nhiên, điều đó có thể không đúng với  $x = \epsilon$  vì  $\delta'(q_0, \epsilon) = \{q_0\}$  trong khi  $\delta^*(q_0, \epsilon) = \epsilon\text{-CLOSURE}(q_0)$ .

Do đó, cơ sở quy nạp bắt đầu với độ dài chuỗi là 1.

Với  $|x| = 1$  thì x là một ký hiệu a và  $\delta'(q, a) = \delta^*(q, a)$  theo định nghĩa  $\delta'$ . Xét  $|x| > 1$ :

đặt  $x = wa$  với  $a$  là một ký hiệu trong  $\Sigma$ .

Ta có  $\delta'(q, wa) = \delta'(\delta'(q_0, w), a)$

Theo giả thiết quy nạp thì  $\delta'(q_0, w) = \delta^*(q_0, w)$ .

Đặt  $\delta^*(q_0, w) = P$ , ta cần chỉ ra rằng  $\delta(P, a) = \delta^*(q_0, wa)$ .

Ta có  $\delta'(P, a) = \bigcup_{q \in P} \delta'(q, a) = \bigcup_{q \in P} \delta^*(q, a)$ .

Hơn nữa vì  $P = \delta^*(q_0, w)$  nên  $\bigcup_{q \in P} \delta^*(q, a) = \delta^*(q_0, wa)$  ( quy tắc 2 trong định nghĩa  $\delta^*$ ).

Vậy  $\delta'(q_0, wa) = \delta^*(q_0, wa)$

Để đầy đủ chứng minh ta còn phải chỉ ra rằng  $\delta'(q_0, x)$  chứa một trạng thái trong  $F'$  nếu và chỉ nếu  $\delta^*(q_0, x)$  chứa một trạng thái trong  $F$ .

Nếu  $x = \varepsilon$  thì điều đó hiển nhiên đúng (theo định nghĩa của  $F'$ )

Nếu  $x \neq \varepsilon$  thì ta đặt  $x = wa$  với  $a \in \Sigma$ .

Nếu  $\delta^*(q_0, x)$  chứa một trạng thái trong  $F$  thì chắc chắn  $\delta'(q_0, x)$  chứa cùng trạng thái trong  $F'$ . Ngược lại, nếu  $\delta'(q_0, x)$  chứa một trạng thái trong  $F'$  khác hơn  $q_0$  thì  $\delta(q_0, x)$  phải chứa một trạng thái trong  $F$  (vì tập  $F$  và  $F'$  chỉ chênh lệch nhau trạng thái  $q_0$ ). Nếu  $\delta'(q_0, x)$  có chứa trạng thái  $q_0$  và  $q_0$  cũng là một trạng thái thuộc tập trạng thái kết thúc  $F$  thì vì  $\delta^*(q_0, x) = \varepsilon\text{-CLOSURE}(\delta(\delta^*(q_0, w), a))$ , nên trạng thái chung trong  $\varepsilon\text{-CLOSURE}(q_0)$  và trong  $F$  phải ở trong  $\delta^*(q_0, x)$ .

Ví dụ: Chuyển NFA với  $\varepsilon$ -dịch chuyển ở Hình 5 sang dạng NFA không có chứa  $\varepsilon$ - dịch chuyển.

Ta xây dựng NFA tương đương  $M'(Q, \Sigma, \delta', q_0, F')$  chấp nhận  $L(M)$  như sau:

$Q = \{q_0, q_1, q_2\}$

$\Sigma = \{0, 1, 2\}$

- Trạng thái bắt đầu:  $q_0$
- $F' = \{q_0, q_2\}$  do  $\varepsilon\text{-CLOSURE}(q_0) = \{q_0, q_1, q_2\}$  có chứa  $q_2 \in F$
- Hàm chuyển  $\delta'$  của  $M'$  được xác định theo công thức:

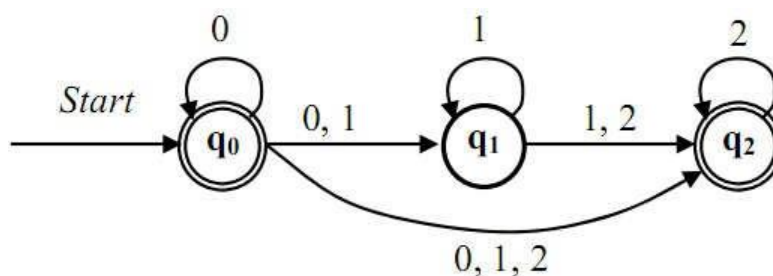
$\delta'(q, a) = \delta^*(q, a) = \varepsilon\text{-CLOSURE}(\delta(\delta^*(q_0, \varepsilon), a))$

Kết quả được chỉ ra trong bảng hàm chuyển sau:

Bảng 3

$\delta$	Inputs		
Trạng thái	0	1	2
$q_0$	$\{q_0, q_1, q_2\}$	$\{q_1, q_2\}$	$\{q_2\}$
$q_1$	$\emptyset$	$\{q_1, q_2\}$	$\{q_2\}$
$q_2$	$\emptyset$	$\emptyset$	$\{q_2\}$

Sơ đồ chuyển trạng thái:



Hình 6: NFA tương đương cho ví dụ

## 6. Giải thuật xây dựng DFA từ NFA

Qua khảo sát các dạng mở rộng từ mô hình ô tô máy hữu hạn ban đầu, ta thấy DFA thực chất là một trường hợp đặc biệt của NFA, nhưng:

- Nó không có sự truyền rỗng (truyền trên nhãn  $\epsilon$ )
- Với mỗi trạng thái  $q$  và ký hiệu nhập  $a$ , chỉ có duy nhất một đường truyền đến một trạng thái khác.

Giả sử mỗi trạng thái của DFA là một tập trạng thái của NFA, DFA dùng trạng thái của mình để lưu giữ tất cả các trạng thái của NFA đạt được sau khi NFA đọc một ký tự nhập. Như vậy sau khi đọc các ký tự nhập  $a_1, a_2, \dots, a_n$ , DFA ở trạng thái là tập con của các trạng thái thuộc NFA, đạt được khi NFA đi từ trạng thái bắt đầu theo một con đường nào đó có tên  $a_1 a_2 \dots a_n$ . Số trạng thái của DFA lúc đó phải bằng số phần tử trong tập lũy thừa của số trạng thái NFA. Song, trên thực tế trường hợp xấu nhất này ít khi xảy ra. Các trạng thái thực sự được dùng trong sơ đồ chuyển cho một DFA sẽ được xác định theo các phép chuyển trạng thái trên nhãn là mọi ký hiệu từ trạng thái bắt đầu của DFA, và sau đó lần lượt được bổ sung thêm vào tập trạng thái nếu như nó chưa có trong đó. Giải thuật chi tiết được trình bày như sau:

- *Input*: Một ô tô máy hữu hạn không đơn định NFA.
- *Output*: Một ô tô máy hữu hạn đơn định DFA nhận dạng cùng ngôn ngữ như NFA.
- *Phương pháp*: Xây dựng bảng hàm chuyển cho DFA mô phỏng đồng thời tất cả các chuyển dịch của NFA trên chuỗi nhập cho trước.

Ta dùng các tác vụ sau để lưu giữ các tập trạng thái của NFA: ( $q$ : là một trạng thái của NFA,  $T$ : là tập trạng thái của NFA)

- a)  $\epsilon$ -closure( $q$ ): là tập trạng thái của NFA đạt được từ trạng thái  $q$  trên sự truyền rỗng.

- b)  $\epsilon$ -closure(T): là tập trạng thái của NFA đạt được từ tất cả các trạng thái q thuộc tập T trên sự truyền rỗng.
- c)  $\delta(T, a)$ : là tập trạng thái của NFA đạt được từ tất cả các trạng thái q thuộc tập T trên sự truyền bằng ký hiệu a.
- *Phân tích*: Trước khi đọc vào một ký tự nhập, DFA có thể ở một trạng thái bất kỳ trong các trạng thái thuộc  $\epsilon$ -closure( $q_0$ ) với  $q_0$  là trạng thái bắt đầu của NFA. Gọi trạng thái này là T. Giả sử các trạng thái của T là các trạng thái đạt được từ  $q_0$  trên các ký hiệu nhập và giả sử a là ký hiệu nhập kế tiếp. Khi đọc a, NFA có thể chuyển đến một trạng thái bất kỳ trong tập trạng thái  $\delta(T, a)$ . Khi chúng ta cho phép sự truyền rỗng, NFA có thể ở bất kỳ trạng thái nào trong  $\epsilon$ -closure( $\delta(T, a)$ ) sau khi đã đọc a.

- *Giải thuật*

```

Trạng thái bắt đầu  $\epsilon$ -closure( $q_0$ ) chỉ là một trạng thái trong
các trạng thái của DFA và trạng thái này chưa được đánh dấu;

While Có một trạng thái T của DFA chưa được đánh dấu do
Begin

    Đánh dấu T; { xét trạng thái T}

For Với mỗi ký hiệu nhập a do
    begin

         $U := \epsilon$ -closure( $\delta(T, a)$ )

        If U không có trong tập trạng thái của DFA then
            begin

                Thêm U vào tập các trạng thái của DFA và
                    trạng thái này chưa được đánh dấu;

                 $\delta[T, a] := U$ ; { $\delta[T, a]$  là phần tử của bảng
                    chuyển DFA}
            end
    end

```

Ta xây dựng các trạng thái và bảng hàm chuyển cho DFA theo cách như sau:

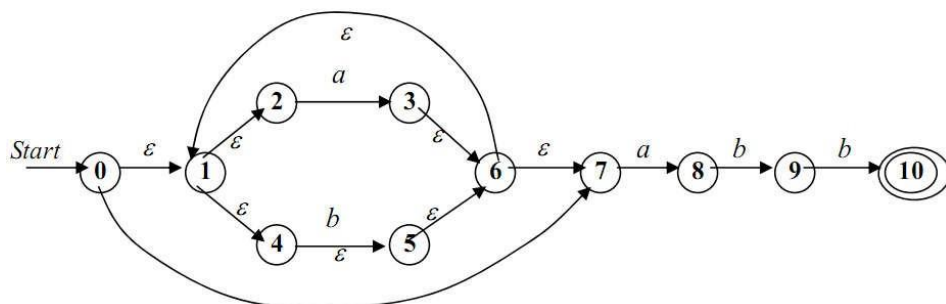
- Mỗi trạng thái của DFA tương trưng bởi một tập trạng thái của NFA mà NFA có thể chuyển đến sau khi đọc một chuỗi ký hiệu nhập gồm: tất cả sự truyền rỗng có thể xảy ra trước hoặc sau các ký hiệu được đọc.
- Trạng thái bắt đầu của DFA là  $\epsilon$ -closure( $q_0$ )
- Các trạng thái và hàm chuyển sẽ được thêm vào D bằng giải thuật trên.
- Một trạng thái của DFA là trạng thái kết thúc nếu nó là tập các trạng thái



của NFA chứa ít nhất một trạng thái kết thúc của NFA.

Việc tính toán  $\epsilon$ -closure(T) có thể xem như quá trình tìm kiếm một đồ thị của các nút từ các nút cho trước và đồ thị bao gồm toàn những cạnh có nhãn  $\epsilon$  của NFA. Giải thuật đơn giản để tìm  $\epsilon$ -closure(T) là dùng Stack để lưu giữ các trạng thái mà cạnh của chúng chưa được kiểm tra cho sự truyền rỗng.

Ví dụ: Tạo DFA từ NFA  $\epsilon$  sau. Các bước xây dựng tập trạng thái cho DFA:



Hình 7: Ví dụ chuyển NFA có  $\epsilon$ -dịch chuyển

Các bước xây dựng tập trạng thái cho DFA:

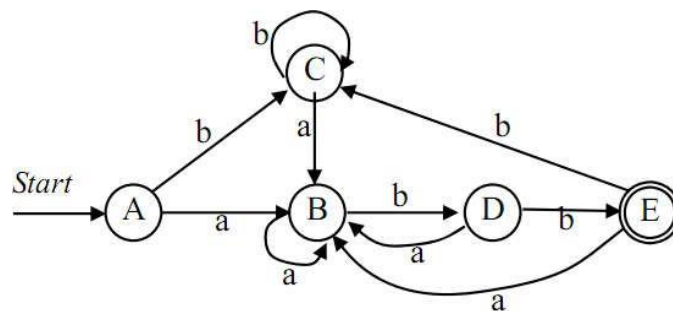
- 1) Trạng thái bắt đầu của DFA:  $\epsilon$ -closure(0) = {0, 1, 2, 4, 7} = A\*
- 2)  $\epsilon$ -closure( $\delta(A, a)$ ) =  $\epsilon$ -closure({3, 8}) = {1, 2, 3, 4, 6, 7, 8} = B\*
- 3)  $\epsilon$ -closure( $\delta(A, b)$ ) =  $\epsilon$ -closure({5}) = {1, 2, 4, 5, 6, 7} = C\*
- 4)  $\epsilon$ -closure( $\delta(B, a)$ ) =  $\epsilon$ -closure({3, 8}) = B
- 5)  $\epsilon$ -closure( $\delta(B, b)$ ) =  $\epsilon$ -closure({5, 9}) = {1, 2, 4, 5, 6, 7, 9} = D\*
- 6)  $\epsilon$ -closure( $\delta(C, a)$ ) =  $\epsilon$ -closure({3, 8}) = B
- 7)  $\epsilon$ -closure( $\delta(C, b)$ ) =  $\epsilon$ -closure({5}) = C
- 8)  $\epsilon$ -closure( $\delta(D, a)$ ) =  $\epsilon$ -closure({3, 8}) = B
- 9)  $\epsilon$ -closure( $\delta(D, b)$ ) =  $\epsilon$ -closure({5, 10}) = {1, 2, 4, 5, 6, 7, 10} = E\*
- 10)  $\epsilon$ -closure( $\delta(E, a)$ ) =  $\epsilon$ -closure({3, 8}) = B
- 11)  $\epsilon$ -closure( $\delta(E, b)$ ) =  $\epsilon$ -closure({5}) = C

Từ các tập trạng thái này, ta xác định được A là trạng thái bắt đầu, E là trạng thái kết thúc (vì trong E có chứa trạng thái 10 là trạng thái kết thúc của NFA) và bảng hàm chuyển của DFA như sau:

Bảng 4

Trạng thái	Ký hiệu nhập	
	a	b
A	B	C
B	B	D
C	B	C
D	B	E
E	B	C

Từ bảng hàm chuyển như trên, ta xây dựng sơ đồ chuyển trạng thái cho DFA tương đương nhận dạng cùng ngôn ngữ có dạng như sau:



Hình 8: DFA tương đương cho ví dụ

*Nhận xét:* Mặc dù có sự khác nhau trong định nghĩa, ta thấy dạng không đơn định NFA được định nghĩa tổng quát hơn dạng đơn định DFA, nhưng rõ ràng khả năng nhận dạng cùng lớp ngôn ngữ của chúng là tương đương nhau. Trong thực tế, các máy tính số hoàn toàn là đơn định, trạng thái của chúng tại mỗi thời điểm là xác định được duy nhất từ một chuỗi nhập bất kỳ và trạng thái bắt đầu.

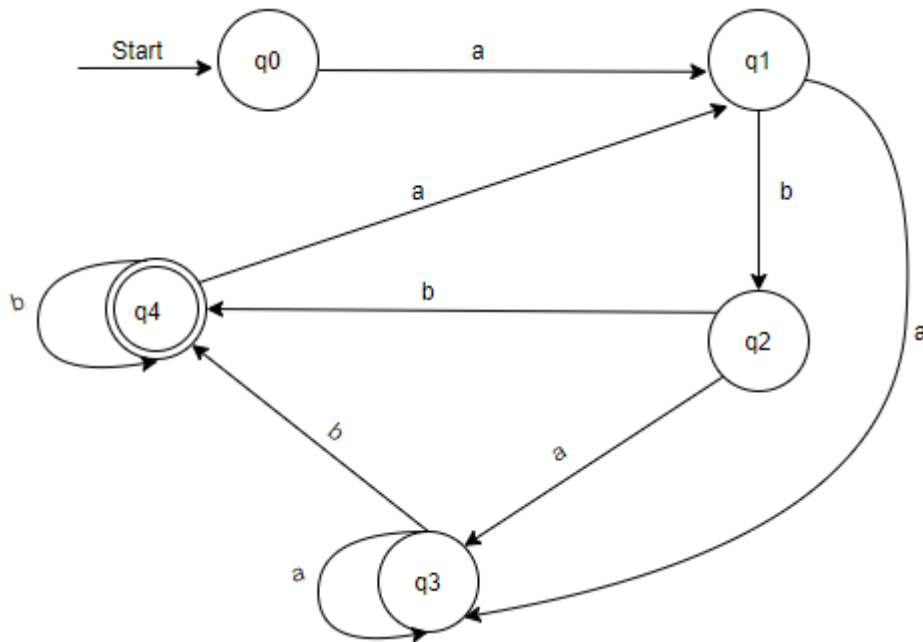
# XÂY DỰNG DFA ĐOÁN NHẬN NGÔN NGỮ

Ví dụ 1: DFA đoán nhận ngôn ngữ

Xây dựng DFA  $M(\{a, b\}, \{q_0, q_1, q_2, q_3, q_4\}, \delta, q_0, \{q_4\})$

Bảng trạng thái:

$\delta$	a	b
$\rightarrow q_0$	$q_1$	$\emptyset$
$q_1$	$q_3$	$q_2$
$q_2$	$q_3$	$q_4$
$q_3$	$q_3$	$q_4$
$(q_4)$	$q_1$	$q_4$



1. Xét chuỗi  $x = abaabbb$

Từ trạng thái bắt đầu  $\delta(q_0, \epsilon) = q_0$

$\delta(q_0, a) = \delta(\delta(q_0, \epsilon), a) = \delta(q_0, a) = q_1$

$\delta(q_0, ab) = \delta(\delta(q_0, a), b) = \delta(q_1, a) = q_2$

$\delta(q_0, aba) = \delta(\delta(q_0, ab), a) = \delta(q_2, a) = q_3$

$\delta(q_0, abaa) = \delta(\delta(q_0, aba), a) = \delta(q_3, a) = q_3$

$\delta(q_0, abaab) = \delta(\delta(q_0, abaa), b) = \delta(q_3, b) = q_4$

$\delta(q_0, abaabb) = \delta(\delta(q_0, abaab), b) = \delta(q_4, b) = q_4$

$$\delta(q_0, abaabbb) = \delta(\delta(q_0, abaabb), b) = \delta(q_4, b) = q_4$$

$q_4$  là trạng thái kết thúc ( $q_4 \in F$ )  $\rightarrow$  Chuỗi  $x = abaabbb$  được đoán nhận

## 2. Xét chuỗi $y = aababbabab$

Từ trạng thái bắt đầu  $\delta(q_0, \varepsilon) = q_0$

$$\delta(q_0, a) = \delta(\delta(q_0, \varepsilon), a) = \delta(q_0, a) = q_1$$

$$\delta(q_0, aa) = \delta(\delta(q_0, a), a) = \delta(q_1, a) = q_3$$

$$\delta(q_0, aab) = \delta(\delta(q_0, aa), b) = \delta(q_3, b) = q_4$$

$$\delta(q_0, aaba) = \delta(\delta(q_0, aab), a) = \delta(q_4, a) = q_1$$

$$\delta(q_0, aabab) = \delta(\delta(q_0, aaba), b) = \delta(q_1, b) = q_2$$

$$\delta(q_0, aababb) = \delta(\delta(q_0, aabab), b) = \delta(q_2, b) = q_4$$

$$\delta(q_0, aababba) = \delta(\delta(q_0, aababb), a) = \delta(q_4, a) = q_1$$

$$\delta(q_0, aababbab) = \delta(\delta(q_0, aababba), b) = \delta(q_1, b) = q_2$$

$$\delta(q_0, aababbaba) = \delta(\delta(q_0, aababbab), a) = \delta(q_2, a) = q_3$$

$$\delta(q_0, aababbabab) = \delta(\delta(q_0, aababbaba), b) = \delta(q_3, b) = q_4$$

$q_4$  là trạng thái kết thúc ( $q_4 \in F$ )  $\rightarrow$  Chuỗi  $y = aababbabab$  được đoán nhận

## 3. Xét chuỗi $z = abbabaab$

Từ trạng thái bắt đầu  $\delta(q_0, \varepsilon) = q_0$

$$\delta(q_0, a) = \delta(\delta(q_0, \varepsilon), a) = \delta(q_0, a) = q_1$$

$$\delta(q_0, ab) = \delta(\delta(q_0, a), b) = \delta(q_1, b) = q_2$$

$$\delta(q_0, abb) = \delta(\delta(q_0, ab), b) = \delta(q_2, b) = q_4$$

$$\delta(q_0, abba) = \delta(\delta(q_0, abb), a) = \delta(q_4, a) = q_1$$

$$\delta(q_0, abbab) = \delta(\delta(q_0, abba), b) = \delta(q_1, b) = q_2$$

$$\delta(q_0, abbaba) = \delta(\delta(q_0, abbab), a) = \delta(q_2, a) = q_3$$

$$\delta(q_0, abbabaa) = \delta(\delta(q_0, abbaba), a) = \delta(q_3, a) = q_3$$

$$\delta(q_0, abbabaab) = \delta(\delta(q_0, abbabaa), b) = \delta(q_3, b) = q_4$$

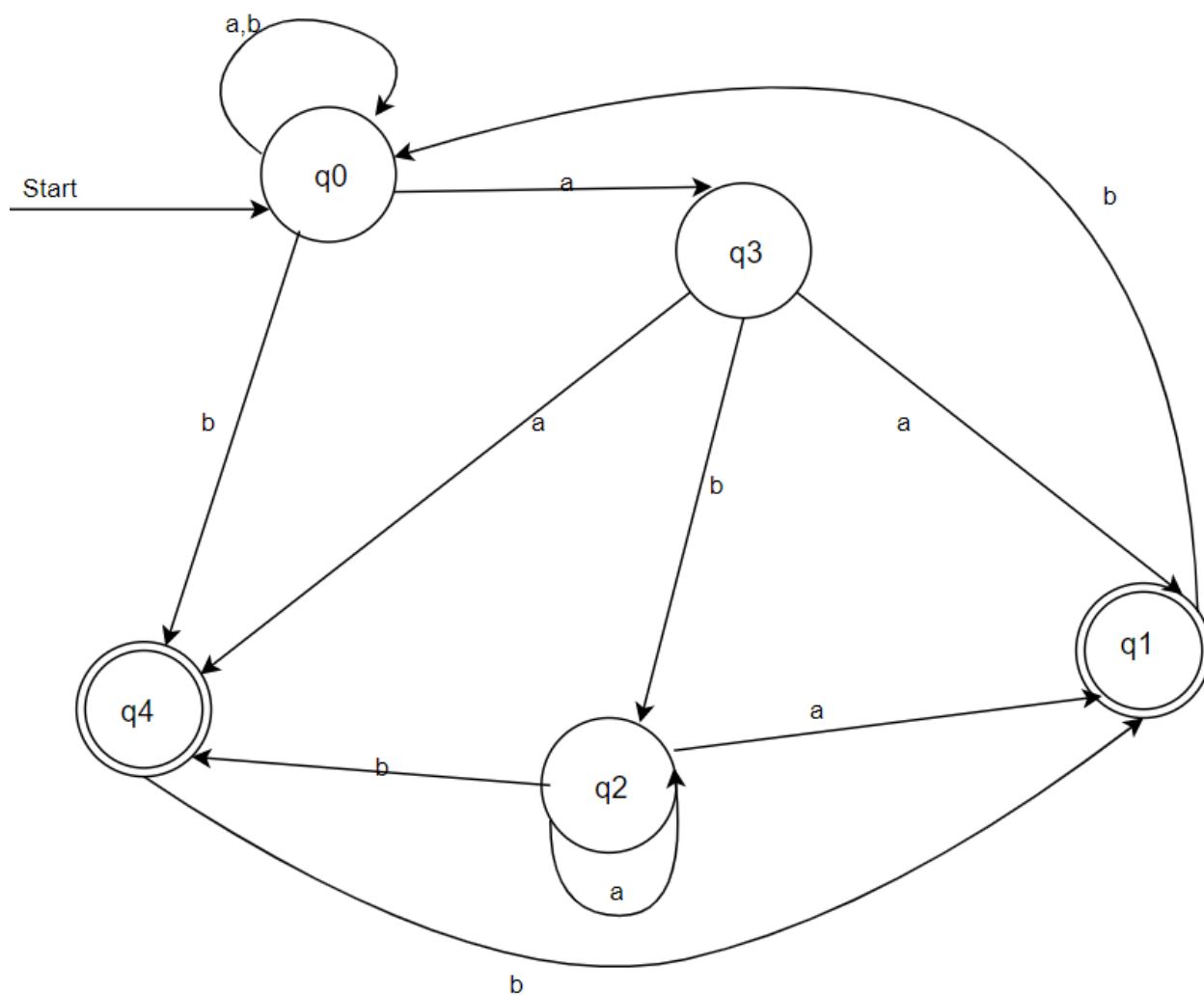
$q_4$  là trạng thái kết thúc ( $q_4 \in F$ )  $\rightarrow$  Chuỗi  $y = aababbabab$  được đoán nhận

Ví dụ 2: NFA đón nhận ngôn ngữ

Xây dựng NFA  $N (\{a, b\}, \{q_0, q_1, q_2, q_3, q_4\}, \delta, q_0, \{q_1, q_4\})$

Bảng trạng thái:

$\delta$	a	b
$\rightarrow q_0$	$\{q_0, q_3\}$	$\{q_1\}$
$\{q_1\}$	$\emptyset$	$\{q_0\}$
$q_2$	$\{q_1, q_2\}$	$\{q_4\}$
$q_3$	$\{q_1, q_4\}$	$\{q_2\}$
$(q_4)$	$\emptyset$	$\{q_1\}$



1. Xét chuỗi  $x = abaabbb$

Quá trình đoán nhận chuỗi:

$$q_0abaabbb \rightarrow q_3baabbb \rightarrow q_2aabbb \rightarrow q_2abbb \rightarrow q_1bbb \rightarrow q_0bb \rightarrow q_0b \rightarrow q_4 \in F$$

2. Xét chuỗi  $y = aababbabb$

Quá trình đoán nhận chuỗi:

$$q_0aababbabb \rightarrow q_0ababbabb \rightarrow q_3babbabb \rightarrow q_2abbabb \rightarrow q_1bbabb \rightarrow q_0babb \\ \rightarrow q_0abab \rightarrow q_0bab \rightarrow q_4b \rightarrow q_1 \in F$$

3. Xét chuỗi  $z = abbbabaab$

Quá trình đoán nhận chuỗi:

$$q_0abbbabaab \rightarrow q_0bbbabaab \rightarrow q_4bbabaab \rightarrow q_1babaab \rightarrow q_0abaab \rightarrow q_3baab \\ \rightarrow q_2aab \rightarrow q_2ab \rightarrow q_2b \rightarrow q_4 \in F$$