

# Biểu thức chính quy

Trần Vĩnh Đức

Trường Đại Học Bách Khoa Hà Nội

Ngày 3 tháng 3 năm 2019

# Thuật ngữ

- ▶ Regular Expression (RE) : Biểu thức chính quy
- ▶ Finite Automaton (FA): Otomat hữu hạn
- ▶ Deterministic Finite Automaton (DFA) : Otomat hữu hạn đơn định
- ▶ Nondeterministic Finite Automaton (NFA): Otomat hữu hạn đa định

# Nội dung

Giới thiệu và ví dụ

Định nghĩa

Sự tương đương giữa RE và FA

# Nội dung

Giới thiệu và ví dụ

Định nghĩa

Sự tương đương giữa RE và FA

## Giới thiệu

- ▶ Trong số học, người ta dùng các phép toán  $+$  và  $\times$  để xây dựng các biểu thức, ví dụ

$$(5 + 3) \times 4.$$

Giá trị của biểu thức trên là số 32.

- ▶ Tương tự, ta có thể dùng các phép toán chính quy để mô tả ngôn ngữ, gọi là **Biểu thức chính quy**, ví dụ

$$(1 \cup 0)0^*.$$

Giá trị của biểu thức chính quy trên là ngôn ngữ

$$(\{0\} \cup \{1\}) \{0\}^*.$$

## Ví dụ

- ▶ Xét biểu thức chính quy

$$(0 \cup 1)^*.$$

Giá trị của biểu thức này là ngôn ngữ gồm **mọi xâu nhị phân**.

- ▶ Giá trị của biểu thức chính quy

$$\Sigma^*1$$

là mọi xâu kết thúc bởi 1 trên bảng chữ  $\Sigma$ .

- ▶ Ngôn ngữ

$$(0\Sigma^*) \cup (\Sigma^*1)$$

bao gồm mọi xâu **bắt đầu** bởi 0 hoặc **kết thúc** bởi 1 trên bảng chữ  $\Sigma$ .

# Ứng dụng của biểu thức chính quy

- ▶ Trong các ứng dụng liên quan tới xử lý văn bản.
- ▶ Tìm kiếm các chuỗi thỏa mãn một mẫu nào đó.
- ▶ Công cụ như AWK, GREP trong UNIX, và ngôn ngữ hiện đại như PERL cung cấp mẫu dùng biểu thức chính quy.

# Nội dung

Giới thiệu và ví dụ

Định nghĩa

Sự tương đương giữa RE và FA



# Biểu thức chính quy

## Định nghĩa

Ta nói  $R$  là **biểu thức chính quy** nếu  $R$  là

1.  $a$  với  $a \in \Sigma$ ,
2.  $\varepsilon$ ,
3.  $\emptyset$ ,
4.  $(R_1 \cup R_2)$  với  $R_1$  và  $R_2$  là các biểu thức chính quy,
5.  $(R_1 R_2)$  với  $R_1$  và  $R_2$  là các biểu thức chính quy,
6.  $(R_1^*)$  với  $R_1$  là biểu thức chính quy.

## Giá trị của biểu thức chính quy

<i>Biểu thức <math>R</math></i>	<i>biểu diễn ngôn ngữ <math>R</math></i>
$a$	$\{a\}$
$\varepsilon$	$\{\varepsilon\}$
$\emptyset$	$\emptyset$
$(R_1 \cup R_2)$	$R_1 \cup R_2$
$(R_1 R_2)$	$R_1 R_2$
$(R_1^*)$	$(R_1)^*$

Các dấu ngoặc có thể bỏ qua. Khi đó, biểu thức chính quy được đánh giá theo thứ tự ưu tiên:

*Phép sao* rồi đến *Phép ghép* rồi đến *Phép hợp*.

## Một vài ký hiệu

- ▶  $R^*$  tập mọi xâu tạo được bằng cách ghép 0 xâu hoặc nhiều xâu của  $R$
- ▶  $R^+ = RR^*$  tập mọi xâu tạo được bằng cách ghép 1 hoặc nhiều lần các xâu của  $R$ . Vậy

$$R^+ \cup \varepsilon = R^*$$

- ▶  $R^k = \underbrace{RR \dots R}_{k \text{ lần}}$

## Ví dụ

Giả sử  $\Sigma = \{0, 1\}$ .

- ▶  $0^*10^* = \{w \mid w \text{ chứa chỉ một } 1\}$
- ▶  $\Sigma^*1\Sigma^* = \{w \mid w \text{ chứa ít nhất một } 1\}$
- ▶  $\Sigma^*001\Sigma^* = \{w \mid w \text{ chứa xâu con } 001\}$
- ▶  $1^*(01^+)^* = \{w \mid \text{sau mỗi } 0 \text{ của } w \text{ phải có ít nhất một } 1\}$
- ▶  $(\Sigma\Sigma)^* = \{w \mid \text{độ dài } w \text{ là số chẵn}\}$
- ▶  $(\Sigma\Sigma\Sigma)^* = \{w \mid \text{độ dài } w \text{ chia hết cho } 3\}$
- ▶  $0\Sigma^*0 \cup 1\Sigma^*1 \cup 0 \cup 1 = \{w \mid w \text{ bắt đầu và kết thúc bởi cùng một ký tự}\}$

## Ví dụ (tiếp)

- ▶  $01 \cup 10 = \{01, 10\}$
- ▶  $(0 \cup \varepsilon)1^* = 01^* \cup 1^*$
- ▶  $(0 \cup \varepsilon)(1 \cup \varepsilon) = \{\varepsilon, 0, 1, 01\}$
- ▶  $1^*\emptyset = \emptyset$

Ghép mọi tập với tập rỗng đều bằng tập rỗng.

- ▶  $\emptyset^* = \{\varepsilon\}$

## Chú ý

- ▶  $R \cup \emptyset = R$ .
- ▶  $R \cup \varepsilon$  có thể không bằng  $R$ .
- ▶  $R \varepsilon = R$ .
- ▶  $R \emptyset$  có thể không bằng  $R$ .

## Ví dụ: Mô tả Hằng số của ngôn ngữ lập trình

- ▶ Các hằng số trong một số ngôn ngữ lập trình có thể mô tả bởi biểu thức chính quy sau:

$$(+ \cup - \cup \varepsilon)(D^+ \cup D^+.D^* \cup D^*.D^+)$$

với  $D = \{0, 1, \dots, 9\}$ .

- ▶ Ví dụ:            72            3.14159            +7.            −.01

# Nội dung

Giới thiệu và ví dụ

Định nghĩa

Sự tương đương giữa RE và FA



# Sự tương đương giữa RE và FA

## Định lý

Một ngôn ngữ là chính quy ***nếu và chỉ nếu*** có một biểu thức chính quy mô tả nó.

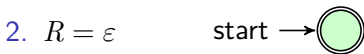
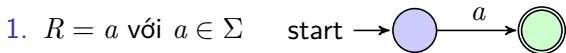
Chứng minh định lý dựa vào hai cách xây dựng sau:

1. RE  $\Rightarrow$  NFA
2. DFA  $\Rightarrow$  RE.

## Bổ đề

*Nếu một ngôn ngữ được mô tả bởi một biểu thức chính quy, vậy nó là ngôn ngữ chính quy.*

Ta sẽ xây dựng NFA  $N$  từ biểu thức chính quy  $R$ .

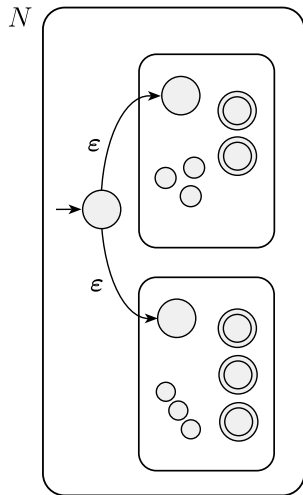
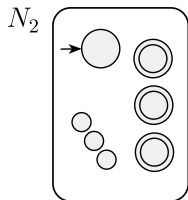
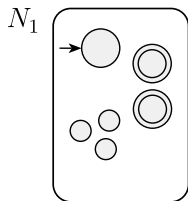


4.  $R = R_1 \cup R_2$

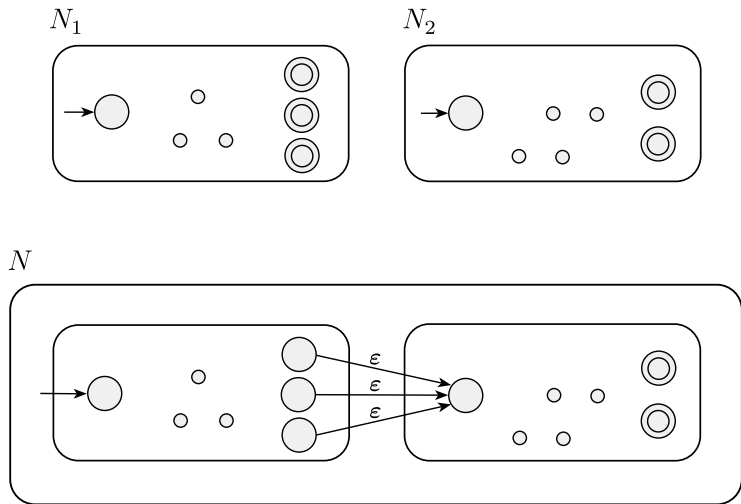
5.  $R = R_1 R_2$

6.  $R = R_1^*$

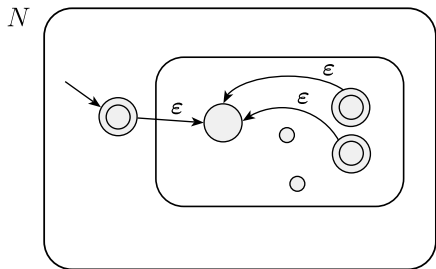
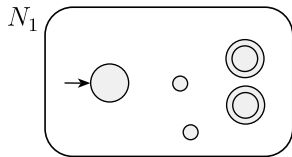
Xây dựng  $R = R_1 \cup R_2$



Xây dựng  $R = R_1 R_2$

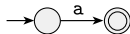


Xây dựng  $R = R_1^*$

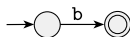


$(ab \cup a)^*$

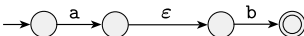
a



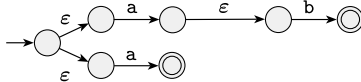
b



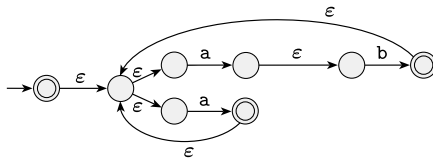
ab



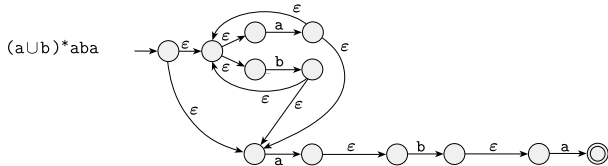
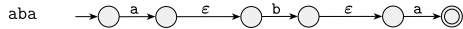
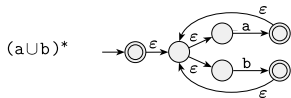
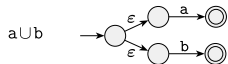
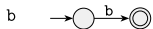
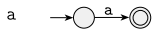
$ab \cup a$



$(ab \cup a)^*$



$(a \cup b)^* aba$



# Xây dựng RE từ DFA

## Bổ đề

*Nếu một ngôn ngữ là chính quy, vậy nó được mô tả bởi một biểu thức chính quy.*

Xây dựng RE từ DFA gồm hai giai đoạn:

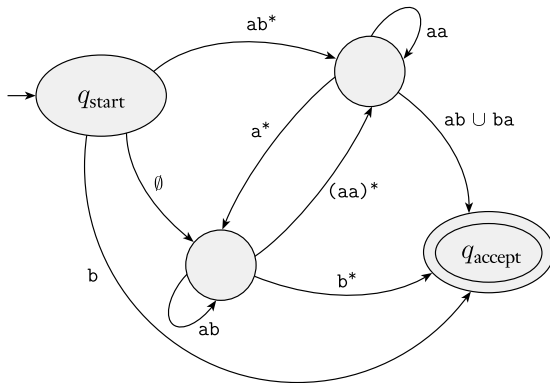
1. Chuyển từ DFA thành otomat đa định được tổng quát hóa (GNFA);
2. Chuyển từ GNFA thành RE.



# Otomat đa định tổng quát hóa (GNFA)

- ▶ GNFA là một NFA *với các chuyển được gán nhãn bởi một biểu thức chính quy* thay vì chỉ  $\epsilon$  hoặc  $a$ .
- ▶ Ta yêu cầu GNFA thỏa mãn ba điều sau:
  1. Từ trạng thái bắt đầu *có* mũi tên đi tới mọi trạng thái khác nhưng *không có* mũi tên nào *đến* nó;
  2. *Chỉ có một trạng thái kết thúc*; và *có mũi tên từ mọi trạng thái khác đi đến nó*, nhưng *không có mũi tên nào từ nó đi ra*. Hơn nữa, nó khác trạng thái bắt đầu.
  3. Trừ trạng thái bắt đầu và kết thúc, mỗi trạng thái có *một mũi tên tới mỗi trạng thái khác*, và cũng có một mũi tên *tới chính nó*.

## Ví dụ: Một GNFA đặc biệt



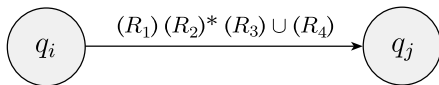
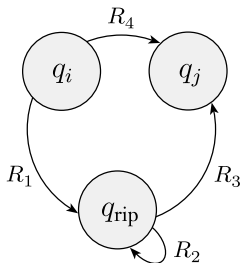
## Chuyển từ DFA thành GNFA đặc biệt

1. Thêm trạng thái mới với một  $\varepsilon$ -chuyển tới trạng thái bắt đầu cũ;
2. Thêm một trạng thái kết thúc mới và các  $\varepsilon$ -chuyển từ mỗi trạng thái kết thúc cũ đến nó;
3. Nếu có nhiều mũi tên (cùng hướng) giữa hai trạng thái, ta thay thế bởi một mũi tên với nhãn là hợp của các nhãn trước;
4. Ta có thể thêm mũi tên với nhãn  $\emptyset$  giữa các cặp trạng thái không có mũi tên.

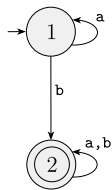
## Chuyển từ GNFA thành RE

Giả sử GNFA có  $k$  trạng thái ( $k \geq 2$ ).

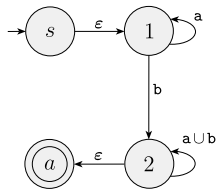
- ▶ Nếu  $k = 2$  vậy GNFA chỉ có một mũi tên từ trạng thái bắt đầu đến trạng thái kết thúc. Nhãn của mũi tên này chính là RE cần tìm.
- ▶ Nếu  $k > 2$ , ta chọn một trạng thái  $q_{rip}$  khác trạng thái bắt đầu và trạng thái kết thúc để loại bỏ, và với mỗi cặp trạng thái  $q_i, q_j$  ta sửa lại nhãn của nó theo quy tắc sau:



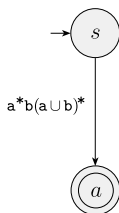
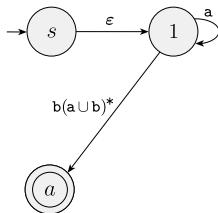
# Ví dụ 1



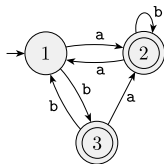
(a)



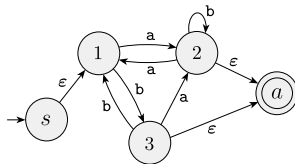
(b)



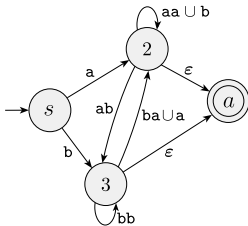
## Ví dụ 2



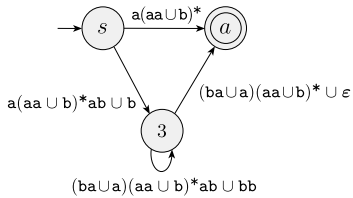
(a)



(b)

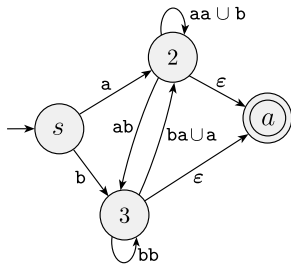


(c)

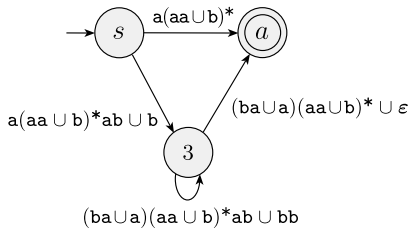


(d)

## Ví dụ 2 (tiếp)



(c)



(d)



$$(a(aa \cup b)^*ab \cup b)((ba \cup a)(aa \cup b)^*ab \cup bb)^*((ba \cup a)(aa \cup b)^* \cup \epsilon) \cup a(aa \cup b)^*$$