Lecture 5. Regular Expressions (biểu thức chính quy)

Hà Chí Trung, BM KHMT, KCNTT, HVKTQS

hct2009@yahoo.com

01685-582-102

Biểu thức chính quy

- **Định nghĩa**: Biểu thức chính quy được định nghĩa một cách đệ quy như sau:
  1. ε là biểu thức chính quy. $L(\varepsilon)=\{\varepsilon\}$.
     $\varnothing$ là biểu thức chính quy. $L(\varnothing)=\varnothing$.
     nếu $a \in \Sigma$, a là biểu thức chính quy. $L(a)=\{a\}$.
  2. Nếu r, s là các biểu thức chính quy thì:
     - $((r))$ là biểu thức chính quy. $L((r))=L(r)$;
     - $r+s$ là biểu thức chính quy. $L(r+s)=L(r)\cup L(s)$;
     - $r.s$ là biểu thức chính quy. $L(r.s)=L(r).L(s)$;
     - r^* là biểu thức chính quy. $L(r^*)=L(r)^*$.

Biểu thức chính quy

- Tìm đọc về **RE**:
  - Jeffrey E. F. Friedl. *Mastering Regular Expressions*, 2nd Edition. O'Reilly & Associates, Inc. 2002.
  - http://www.regular-expressions.info/
- **VD:**
  - ❖ ^[A-Z0-9._%+-]+@[A-Z0-9.-]+\.[A-Z]{2,4}$
  - ❖ ^[A-Z0-9._%+-]+@[A-Z0-9.-]+\.(?:[A-Z]{2}|com|org|net|edu|gov|mil|biz|info|mobi|name|aero|asia|jobs|museum)$

Biểu thức chính quy

- **Ví dụ**:
  - **00** : L(00)={00}
  - **(0+1)*** : L((0+1)*) = {0,1}*
  - **(0+1)*011** : L((0+1)*011) = {0,1}*011
  - **(0+1)*00(0+1)***: L((0+1)*00(0+1)*) = {0,1}*00 {0,1}*
  - **(0+ ε)(1+10)*** : tất cả các chuỗi không có hai số 0 liên tiếp = {ε, 0, 01, 010, 1, 10, 01010, 0111, ... }
  - **0*1*2*** : {ε, 0, 1, 2, 01, 02, 12, 012, 0012, 0112, ... }
  - **00*11*22*** = **$0^+1^+2^+$**

Tính chất của biểu thức chính quy

Phép hợp:

- $r + \varnothing = \varnothing + r = r$
- $r + r = r$
- $r + s = s + r$
- $(r + s) + t = r + (s + t)$
  $= r + s + t$

Phép bao đóng:

- $\varepsilon^* = \varepsilon$
- $\varnothing^* = \varnothing$
- $r^*r^* = r^*$
- $(r^*)^* = r^*$
- $r^* = \varepsilon + r + r^2 + \ldots + r^k + \ldots$
- $r^* = \varepsilon + r^+$
- $(\varepsilon + r)^+ = (\varepsilon + r)^* = r^*$
- $r^*r = r\,r^* = r^+$

Tính chất của biểu thức chính quy

Phép nối kết:

- $r\varepsilon = \varepsilon r = r$
- $r\varnothing = \varnothing r = \varnothing$
- $(r + s)\ t = rt + st$
- $r\ (s + t) = rs + rt$

Tổng hợp:

- $(r^* + s^*)^* = (r^*s^*)^* = (r + s)^*$
- $(rs)^*r = r(sr)^*$
- $(r^*s)^*\ r^* = (r + s)^*$

Thứ tự ưu tiên của phép toán: * (bao đóng), . (phép nối kết), + (phép hợp).

Tính chất của biểu thức chính quy

- **Ví dụ:** Biểu thức chính quy cho ngôn ngữ gồm các xâu nhị phân mà không có hai số 0 hay hai số 1 liên nhau.

$$(01)^*+(10)^*+0(10)^*+1(01)^*$$

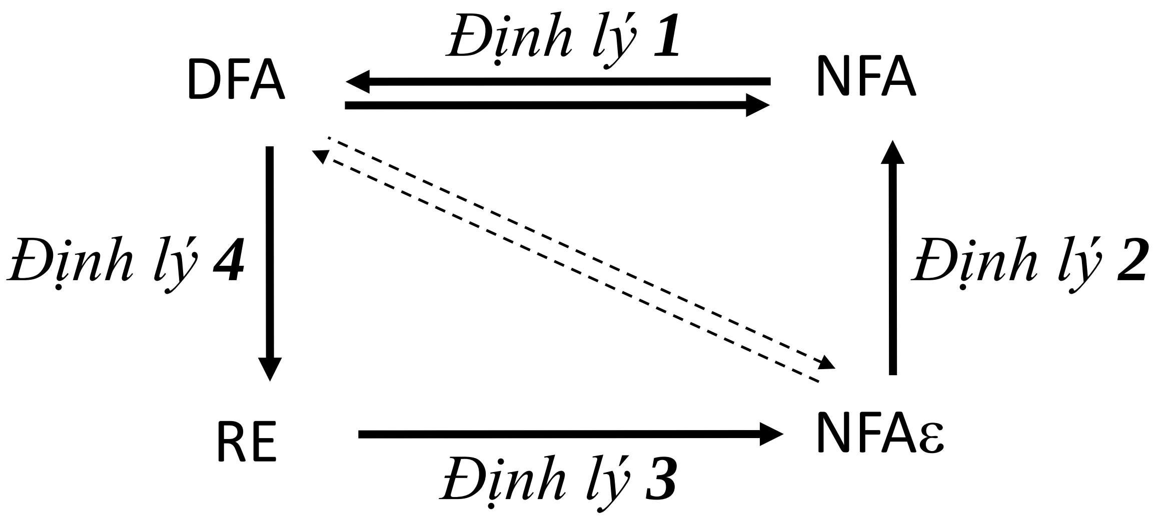
hoặc là:

$$(\varepsilon+1)(01)^*(\varepsilon+0)$$

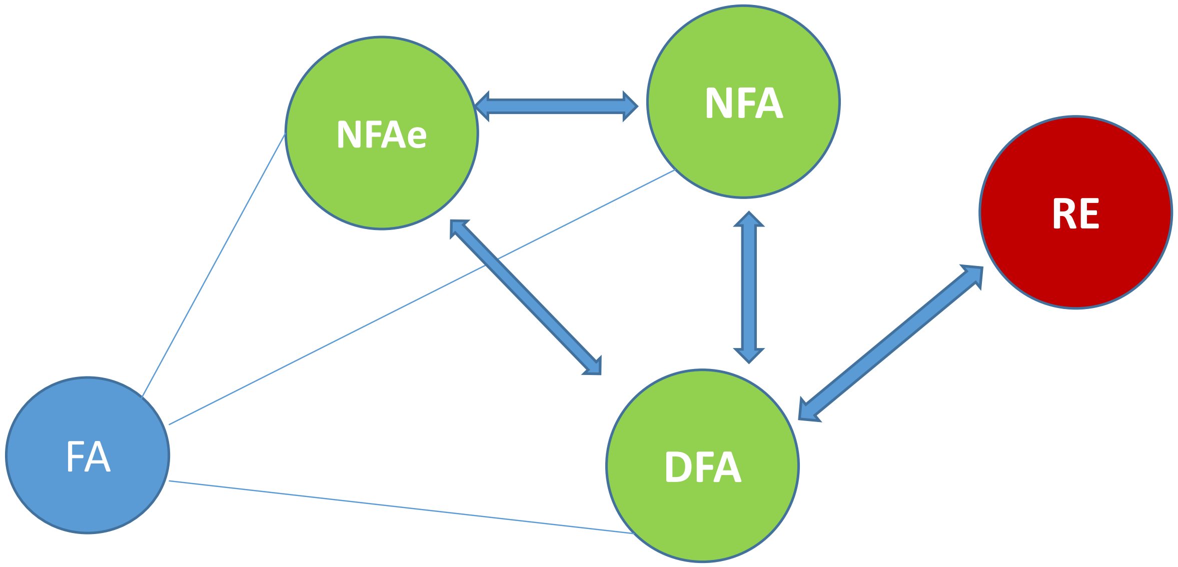
Thứ tự ưu tiên của phép toán: *, . , +

Do đó: $01^* + 1$ được hiểu như sau: $(0(1)^*) + 1$

Sự tương đương giữa FA và RE



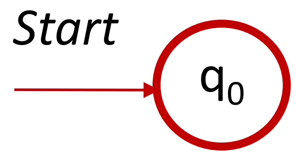
Sự tương đương giữa FA và RE



Sự tương đương giữa FA và RE

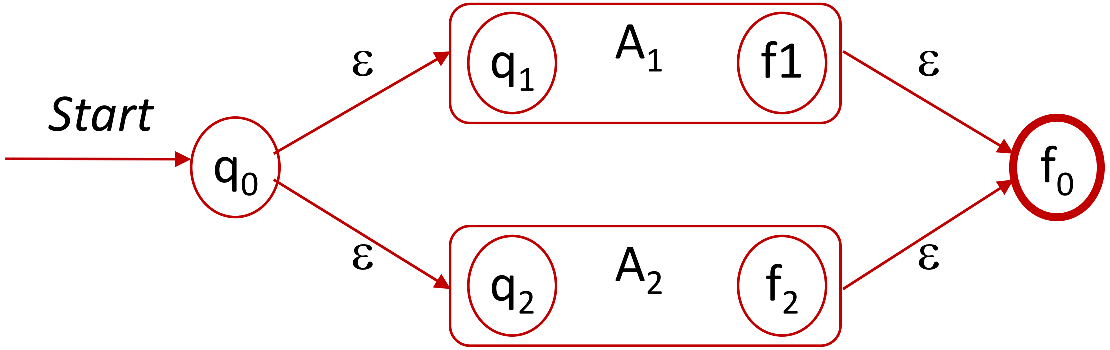
- **Định lý 1:** Nếu L là tập được chấp nhận bởi một NFA thì tồn tại một DFA chấp nhận L.
- **Định lý 2:** Nếu L được chấp nhận bởi một NFAε thì L cũng được chấp nhận bởi một NFA không có ε-dịch chuyển.
- **Định lý 3:** nếu r là RE thì tồn tại một NFAε chấp nhận L(r).
- **Định lý 4:** Nếu L được chấp nhận bởi một DFA, thì L được ký hiệu bởi một RE.

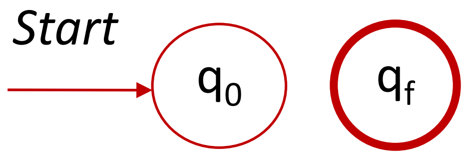
Giải thuật Thompson



$r = \varepsilon$

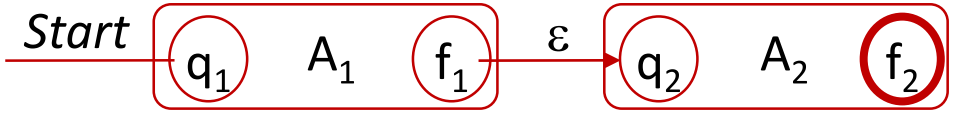
$\mathbf{r = r_1 + r_2}$

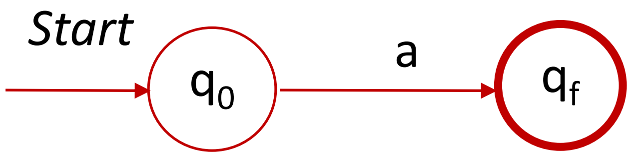




$r = \varnothing$

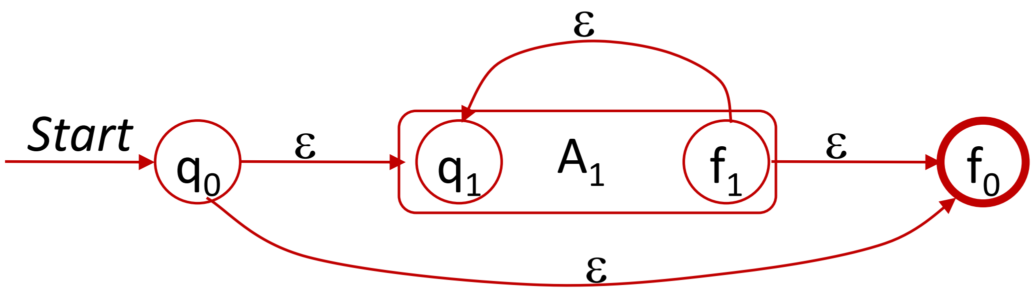
$\mathbf{r = r_1 r_2}$





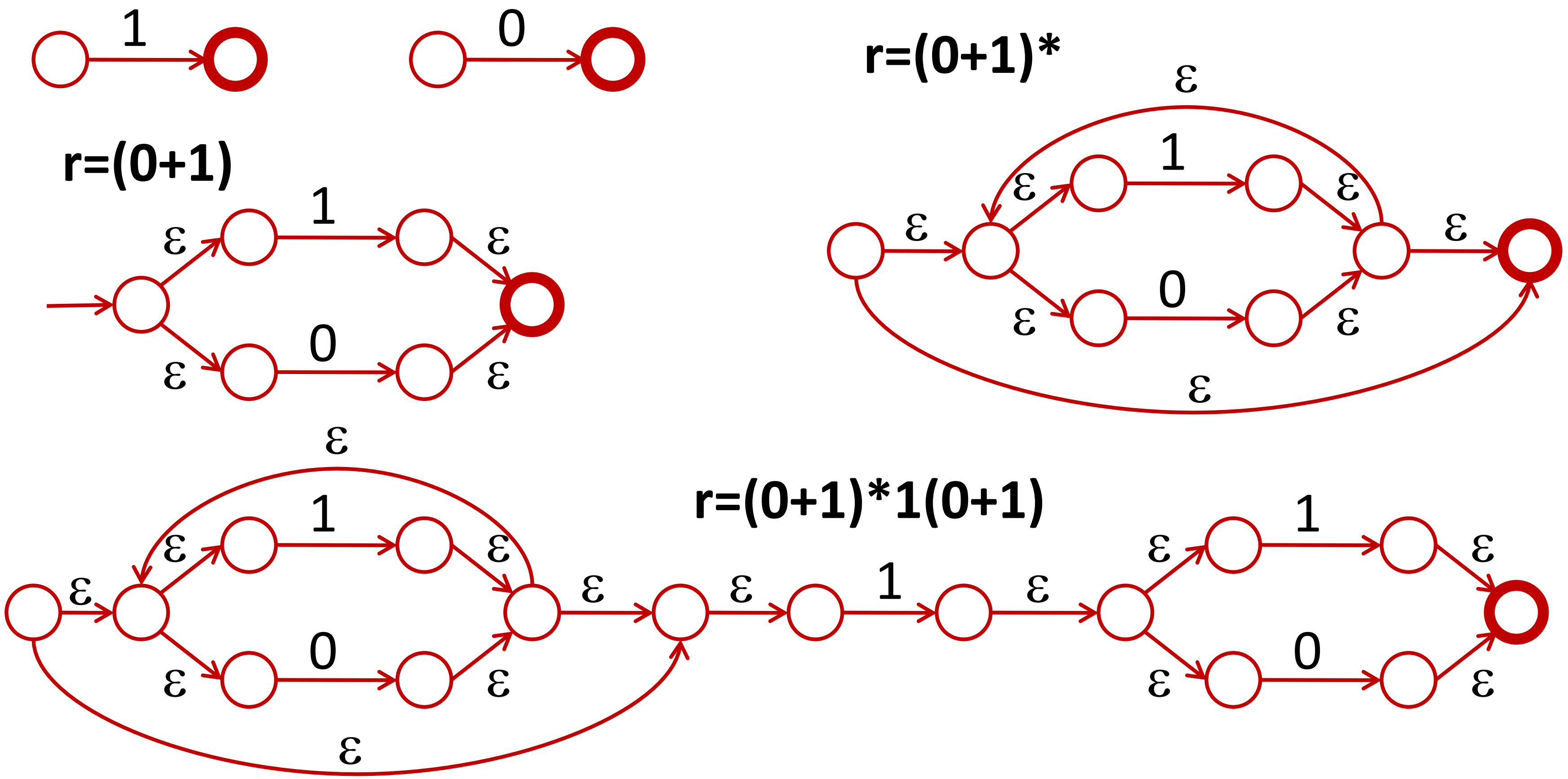
$r = a$

$\mathbf{r = r_1^*}$



Giải thuật Thompson

- **Ví dụ:** Tìm NFAε tương đương cho $(0+1)^*1(0+1)$



Xây dựng DFA trực tiếp từ RE

- **Biến đổi trực tiếp RE** về **DFA** không thông qua **NFA**:
- Trước hết, ta đánh dấu biểu thức cần biến đổi bằng ký hiệu đặc biệt nào đó, chẳng hạn #.

  r →(r)# augmented RE

- Sau đó chúng ta tạo ra cây cú pháp (**syntax tree**) cho biểu thức gia tố:
  - tất cả các ký hiệu kết thúc (gồm cả # và ε) trong biểu thức chính quy đã cho sẽ nằm ở các lá;
  - các nodes bên trong sẽ chứa các toán tử trong biểu thức;
  - Mỗi một ký hiệu kết thúc (gồm cả #) sẽ được đánh thứ tự;
  - Xác định hàm **followpos** cho từng node lá.

Xây dựng DFA trực tiếp từ RE

- **followpos(i)** -- tập hợp các vị trí mà có thể đứng ở sau vị trí i trong biểu thức gia tố. **followpos** chỉ định nghĩa cho node lá, không cho các node trong.
- **Ví dụ:** (a|b) * a → (a|b) * a #

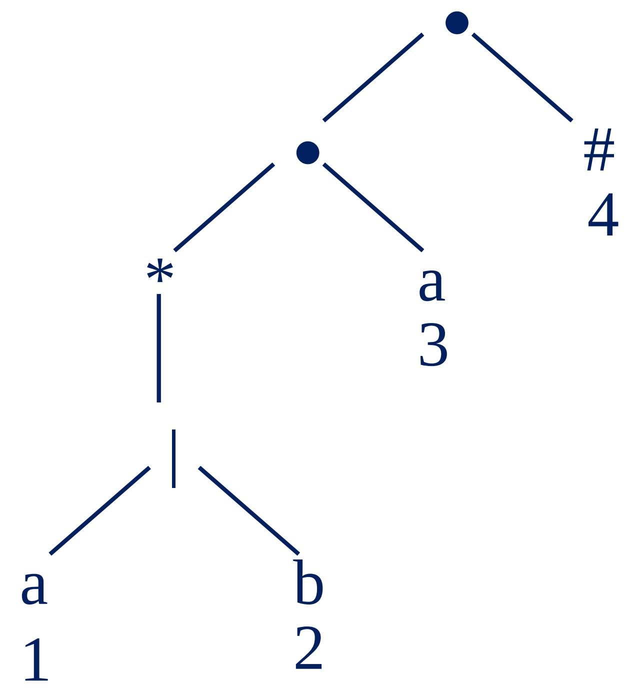
  1 2 3 4

  followpos(1) = {1,2,3}

  followpos(2) = {1,2,3}

  followpos(3) = {4}

  followpos(4) = {}



Xây dựng DFA trực tiếp từ RE

- Để tính được hàm **followpos**, chúng ta cần 2 hàm bổ trợ cho các node, bao gồm cả các nút trong của cây cú pháp.
  - ***firstpos(n)*** -- tập hợp các vị trí của những ký tự đầu tiên trong các xâu được tạo bởi cây con với đỉnh là n.
  - ***lastpos(n)*** -- tập hợp các vị trí của những ký hiệu cuối cùng trong các xâu được sinh bởi biểu thức con với đỉnh là n.
  - ***nullable(n)*** -- true nếu xâu rỗng nằm trong số những xâu sinh bởi biểu thức con với đỉnh là n, ngược là thì là false.
- Bảng tóm tắt cách tính các hàm firstpos, lastpos, nullable

Xây dựng DFA trực tiếp từ RE

n	*nullable(n)*	*firstpos(n)*	*lastpos(n)*
leaf labeled ε	true	∅	∅
leaf labeled with position i	false	{i}	{i}
$\mid$ c_1 c_2	nullable(c_1) or nullable(c_2)	firstpos(c_1) ∪ firstpos(c_2)	lastpos(c_1) ∪ lastpos(c_2)
• c_1 c_2	nullable(c_1) and nullable(c_2)	if (nullable(c_1)) firstpos(c_1) ∪ firstpos(c_2) else firstpos(c_1)	if (nullable(c_2)) lastpos(c_1) ∪ lastpos(c_2) else lastpos(c_2)
* c_1	true	firstpos(c_1)	lastpos(c_1)

Xây dựng DFA trực tiếp từ RE

- **followpos(n)** được tính theo các quy tắc sau:
  1. Nếu n là node kết nối với con bên trái là c_1 và con phải c_2, và i là một vị trí trong **lastpos(c_1)**, khi đó tất cả các vị trí trong **firstpos(c_2)** cũng thuộc về **followpos(i)**.
  2. Nếu **n** là node bao đóng sao (*****), và **i** là một vị trí trong **lastpos(n),** khi đó tất cả các vị trí trong **firstpos(n)** cũng thuộc về **followpos(i)**.
- Nếu **firstpos** và **lastpos** được tính cho từng node, followpos cho mỗi vị trí có thể tính bởi một lần duyệt theo chiều sâu của syntax tree.

Xây dựng DFA trực tiếp từ RE

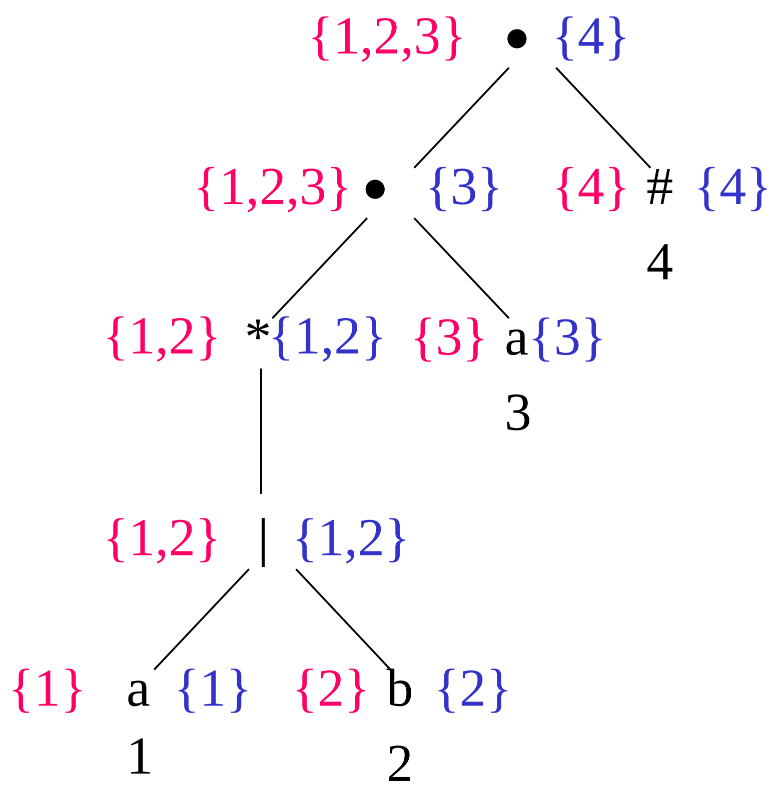
- Xác định firstpos, lastpos

  red – firstpos

  blue – lastpos

- Sau đó tính followpos:

  followpos(1) = {1,2,3}
  followpos(2) = {1,2,3}
  followpos(3) = {4}
  followpos(4) = {}



- Sau khi tính **followpos**, chúng ta có thể biến đổi trực tiếp RE về DFA theo thuật toán sau.

Xây dựng DFA trực tiếp từ RE

1. Tạo cây cú pháp từ (r) #
2. Tính cho từng node của cây: **followpos, firstpos, lastpos, nullable**
3. Đặt **firstpos(root)** làm trạng thái mới của DFA và chưa đánh dấu.
4. *while* (còn trạng thái chưa đánh dấu q của DFA) *do*
   - Đánh dấu **q;**
   - *foreach* (ký hiệu **a**) *do*
     - let $s_1,\ldots,s_n$ là các vị trí trong **q** & ký hiệu tại vị trí đó là **a**
     - **q'** ← followpos(s_1) ∪ ... ∪ followpos(s_n)
     - **δ(S,a) → q'**
     - if (**q'** khác rỗng và chưa có trong DFA)
       thêm **q'** vào tập trạng thái của DFA và chưa đánh dấu.

- *Trạng thái bắt đầu DFA là firstpos(root), trạng thái kết thúc là các nhãn có chứa #*

Xây dựng DFA trực tiếp từ RE

followpos(1)={1,2,3}   followpos(2)={1,2,3}   followpos(3)={4}   followpos(4)={}

S_1=firstpos(root)={1,2,3}

$\Downarrow$ mark q_1

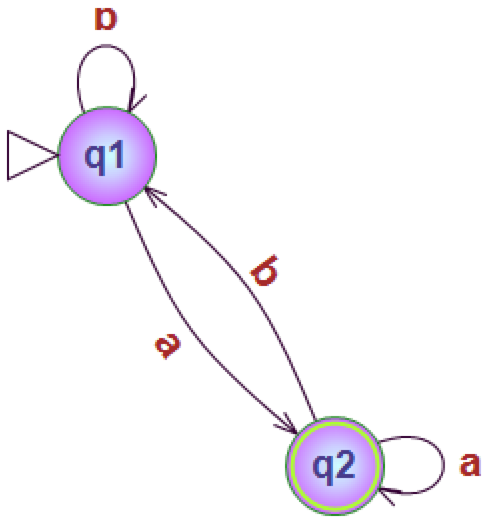
a: followpos(1) $\cup$ followpos(3)={1,2,3,4}=q_2 move(q_1,a)=q_2

b: followpos(2)={1,2,3}=S_1 move(q_1,b)=q_1

$\Downarrow$ mark q_2

a: followpos(1) $\cup$ followpos(3)={1,2,3,4}=q_2 move(q_2,a)=q_2

b: followpos(2)={1,2,3}=S_1 move(q_2,b)=q_1



Trạng thái bắt đầu: q_1 Trạng thái kết thúc: {q_2}

Biến đổi từ FA sang RE

- **Định lý 4:** Nếu L được chấp nhận bởi một DFA, thì L được ký hiệu bởi một RE.
- **Chứng minh:**
  - **L** được chấp nhận bởi **DFA** **A**($\{q_1, q_2, \dots, q_n\}, \Sigma, \delta, q_1, F$)
  - Đặt $\mathbf{R^k_{ij}} = \{x \mid \delta(q_i, x) = q_j$ và nếu $\delta(q_i, y) = q_l$ $(y \subset x)$ thì $l \le k\}$ (có nghĩa là $\mathbf{R^k_{ij}}$ - tập hợp tất cả các chuỗi làm cho automata đi từ trạng thái i đến trạng thái j mà không đi ngang qua trạng thái nào lớn hơn k)
  - Định nghĩa đệ quy của R^k_{ij}:

$$\mathbf{R^k_{ij}} = R^{k-1}_{ik}(R^{k-1}_{kk})^* R^{k-1}_{kj} \cup R^{k-1}_{ij}$$

$$\mathbf{R^0_{ij}} = \begin{cases} \{a \mid \delta(q_i, a) = q_j\}, \text{ nếu } i \ne j \\ \{a \mid \delta(q_i, a) = q_j\} \cup \{\varepsilon\}, \text{ nếu } i = j \end{cases}$$

Biến đổi từ FA sang RE

- bổ đề: với mọi $\mathbf{R}^k_{ij}$ đều tồn tại một biểu thức chính quy ký hiệu cho $\mathbf{R}^k_{ij}$.
  - ❖ k = 0: R^0_{ij} là tập hữu hạn các chuỗi 1 ký hiệu hoặc ε
  - ❖ Giả sử ta có bổ đề trên đúng với k-1, tức là tồn tại RE R^{k-1}_{lm} sao cho $L(R^{k-1}_{lm}) = r^{k-1}_{lm}$
  - ❖ Vậy đối với r^k_{ij} ta có thể chọn RE:

$$r^k_{ij} = (r^{k-1}_{ik})(r^{k-1}_{kk})^*(r^{k-1}_{kj}) + r^{k-1}_{ij}$$

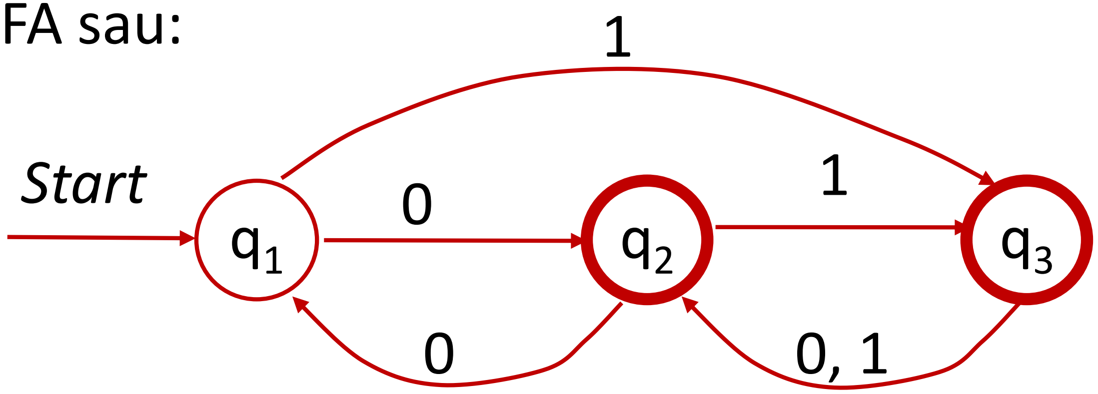
  → bổ đề đã được chứng minh
- *nhận xét:* $L(A) = \bigcup_{q_j \in F} R^n{}_{1j}$. Vậy L có thể được ký hiệu bằng RE:

$$r = r^n_{1j1} + r^n_{1j2} + \ldots + r^n_{1jp}$$

  với $F = \{q_{j1}, q_{j2}, \ldots, q_{jp}\}$

Biến đổi từ FA sang RE

- **Ví dụ** viết RE cho DFA sau:



- Ta cần viết biểu thức:

$$r = r^3_{12} + r^3_{13}$$

- Ta có:

$$r^3_{12} = r^2_{13}(r^2_{33})^*r^2_{32} + r^2_{12}$$

$$r^3_{13} = r^2_{13}(r^2_{33})^*r^2_{33} + r^2_{13}$$

Biến đổi từ FA sang RE

	k = 0	k = 1	k = 2
r^k_{11}	ε	ε	(00)*
r^k_{12}	0	0	0(00)*
r^k_{13}	1	1	0*1
r^k_{21}	0	0	0(00)*
r^k_{22}	ε	ε + 00	(00)*
r^k_{23}	1	1 + 01	0*1
r^k_{31}	∅	∅	(0 + 1)(00)*0
r^k_{32}	0 + 1	0 + 1	(0 + 1)(00)*
r^k_{33}	ε	ε	ε + (0 + 1)0*1

r = 0*1((0 + 1)0*1)* (ε + (0 + 1)(00)*) + 0(00)*