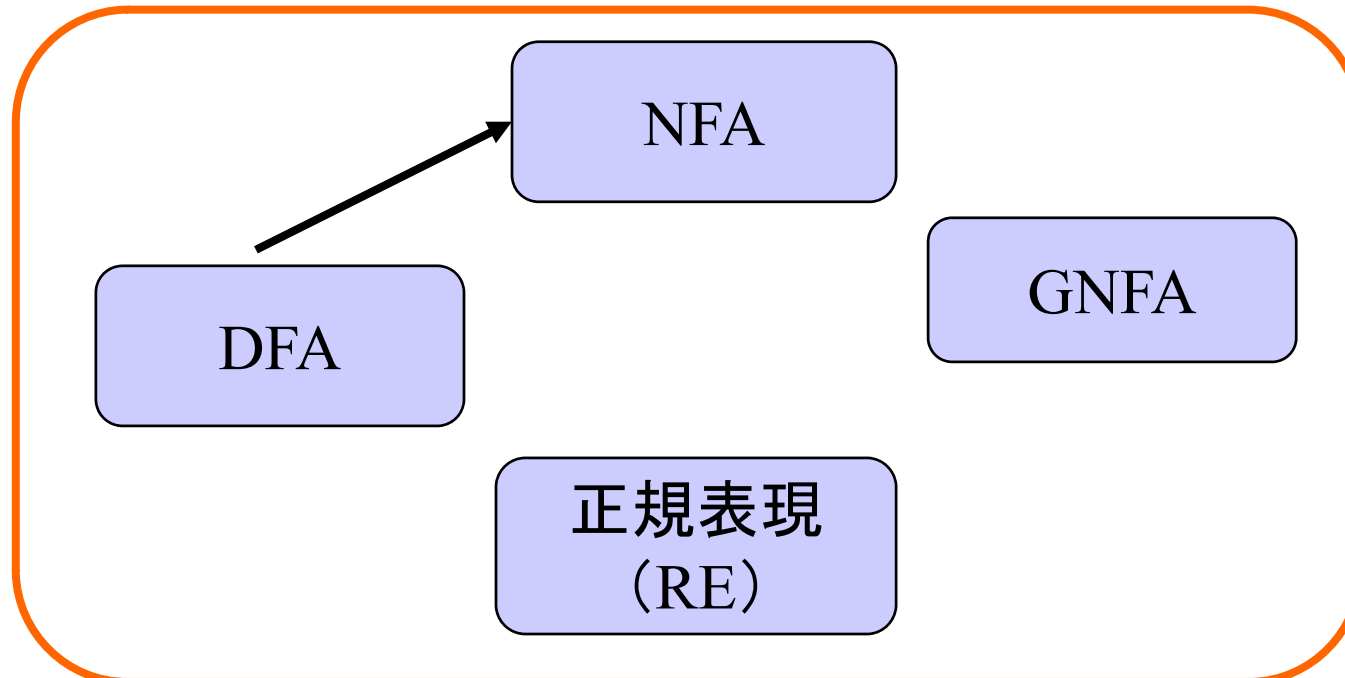


2. 正規言語とオートマトンの等価性

モデル間の関係

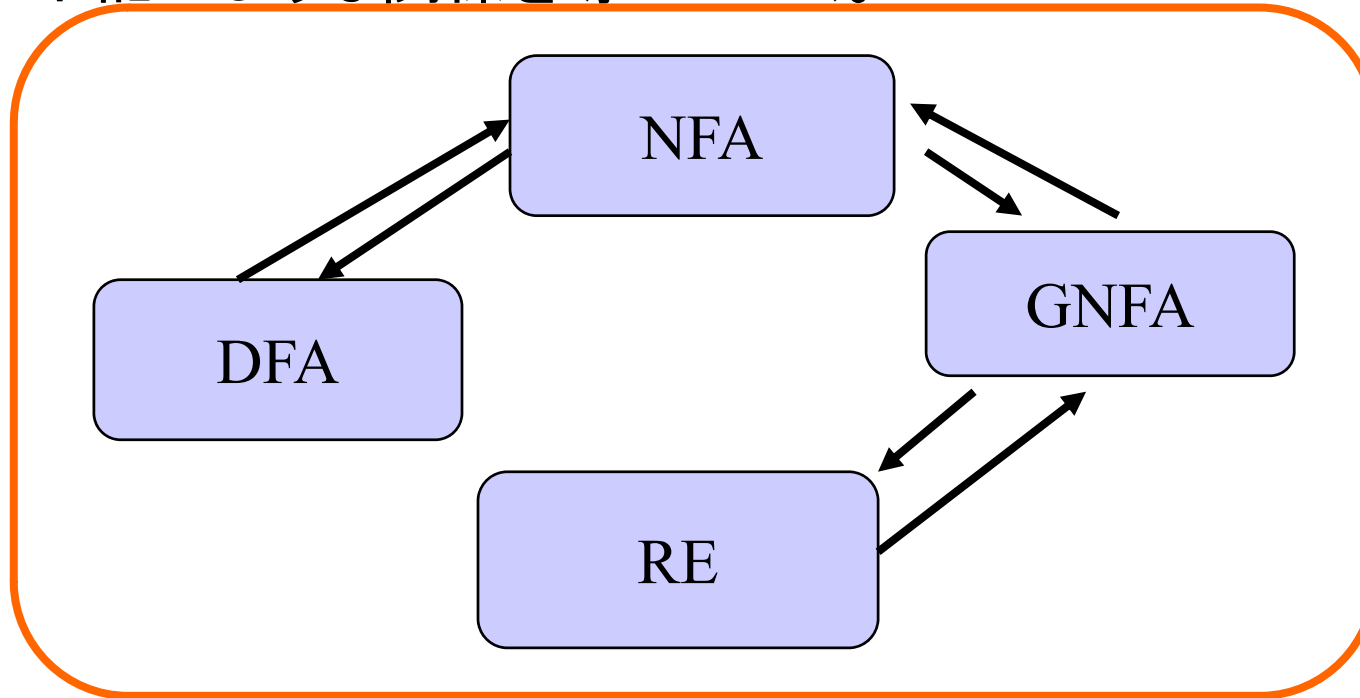
モデルの言語表現能力を評価する。
モデルAが与えられたとき、同じ言語を受理するモデルBが
作れるときモデルBの方が言語記述能力が高い(低くはない)。
これを、 $A \rightarrow B$ とアークを引いて表す。

例えば、DFAの記述は、ほとんどその定義のまま、
NFAの記述になる。したがって、

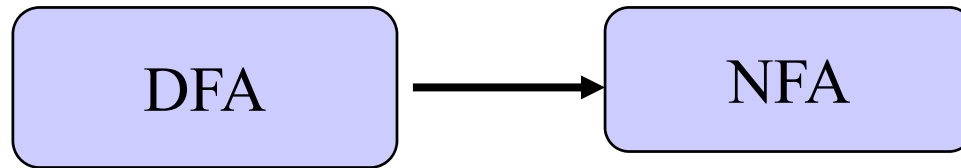


目標

これらのモデルがすべて同じ言語記述能力があることを示したい。そのため、
下記のような関係を導いていく。



DFAからNFAへ



直感的には、明らかだが、ここでは、形式的に示す。

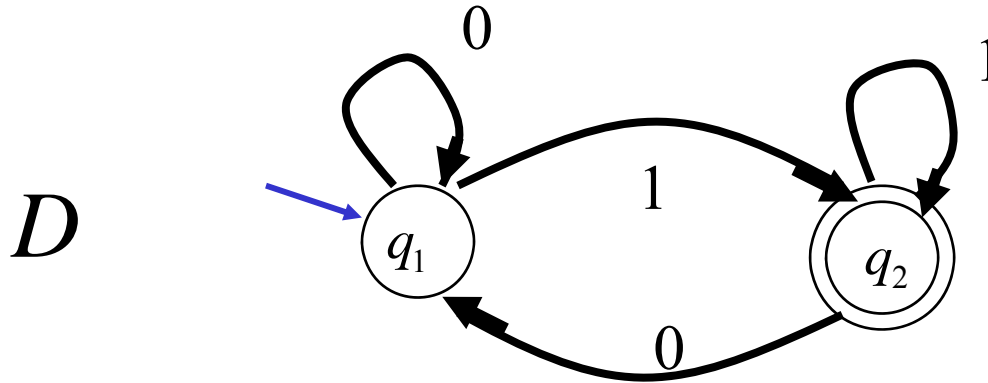
任意のDFAを $D = (Q, \Sigma, \delta, q_0, F)$ とする。
このとき、 D と同じ言語を受理するNFA
 $N = (Q', \Sigma, \delta', q_0', F')$ を構成する。

1. $Q' = Q, q_0' = q_0, F' = F$ とする。
2. $\delta(q_x) = q_y$ のとき、 $\delta'(q_x) = \{q_y\}$ とする。

例

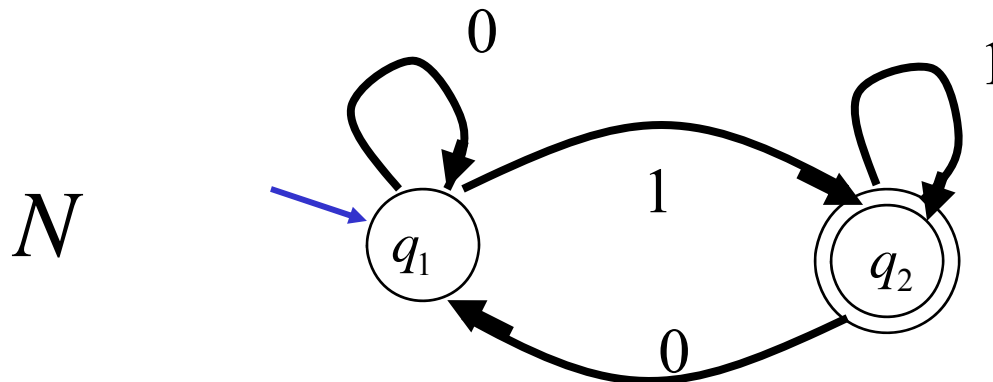
$$D = (\{q_1, q_2\}, \{0, 1\}, \delta, q_1, \{q_2\})$$

| δ | 0 | 1 |
|----------|-------|-------|
| q_1 | q_1 | q_2 |
| q_2 | q_1 | q_2 |

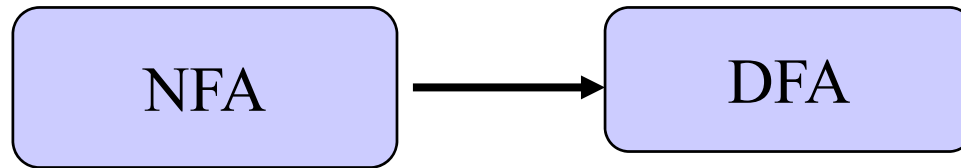


$$N = (\{q_1, q_2\}, \{0, 1\}, \delta', q_1, \{q_2\})$$

| δ' | 0 | 1 |
|-----------|-----------|-----------|
| q_1 | $\{q_1\}$ | $\{q_2\}$ |
| q_2 | $\{q_1\}$ | $\{q_2\}$ |



NFAからDFAへ



任意のNFAを $N = (Q, \Sigma, \delta, q_0, F)$ とする。
このとき、 N と同じ言語を受理するDFA
 $D = (Q', \Sigma, \delta', q_0', F')$ を構成する。

1. $Q' = \mathcal{P}(Q)$ とする。

(NFAの状態のべき集合で、DFAの状態を作る。)

2. $q_0' = \{q_0\}$ とする。

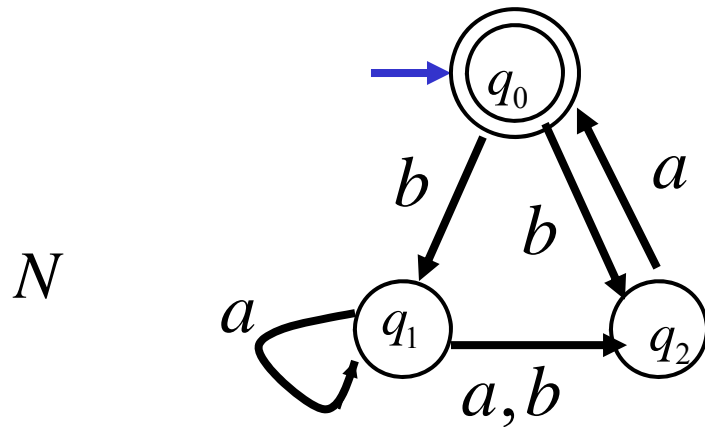
$$\begin{aligned} 3. F' &= \{\text{NFAの受理状態を含むような状態}\} \\ &= \{q \in Q' \mid q \cap F \neq \emptyset\} \end{aligned}$$

$$\begin{aligned} 4. \quad \delta'(q, s) &= \{q \text{のいずれかの状態から} \\ &\quad s \in \Sigma \text{で遷移する状態}\} \\ &= \bigcup_{r \in q} \{\delta(r, s)\} \end{aligned}$$

例 $\Sigma = \{a, b\}$ とする。

NFA $N = \{\{q_0, q_1, q_2\}, \Sigma, \delta, q_0, \{q_0\}\}$

| δ | a | b |
|----------|----------------|----------------|
| q_0 | ϕ | $\{q_1, q_2\}$ |
| q_1 | $\{q_1, q_2\}$ | $\{q_2\}$ |
| q_2 | $\{q_0\}$ | ϕ |



このNと同じ言語を受理するDFA
 $D = \{Q', \Sigma, \delta', q_0', \{q_0\}, F'\}$
を作成する。

$$\begin{aligned}
 1. \quad Q' &= \mathcal{P}(Q) \\
 &= \left\{ \phi, \{q_0\}, \{q_1\}, \{q_2\}, \{q_0, q_1\}, \{q_0, q_2\}, \right. \\
 &\quad \left. \{q_1, q_2\}, \{q_0, q_1, q_2\} \right\}
 \end{aligned}$$

ϕ

$\{q_0\}$

$\{q_1\}$

$\{q_2\}$

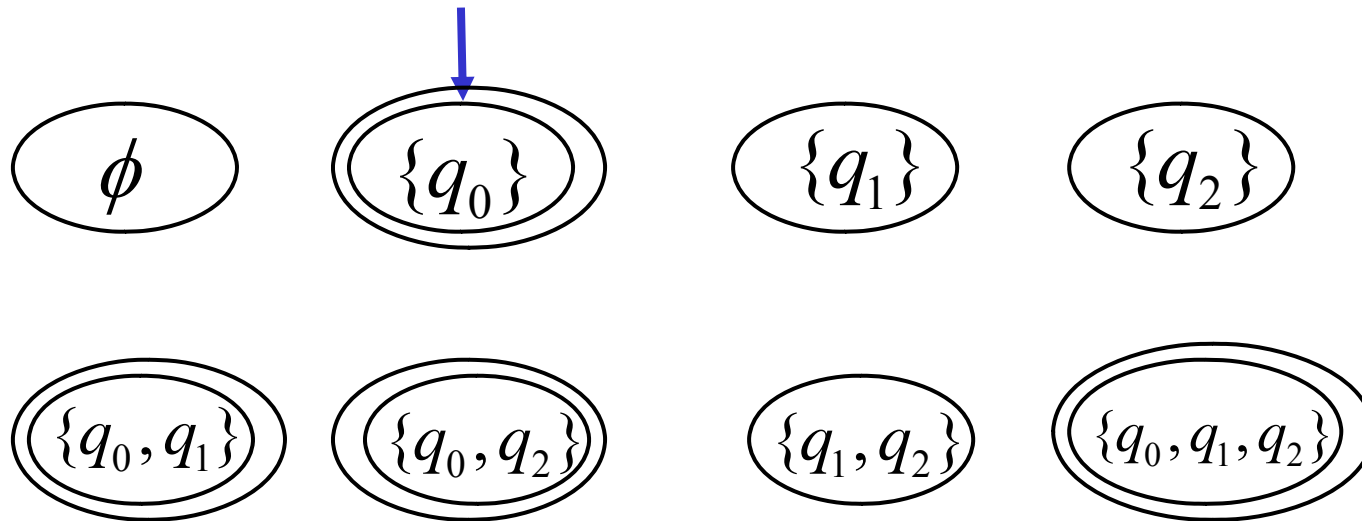
$\{q_0, q_1\}$

$\{q_0, q_2\}$

$\{q_1, q_2\}$

$\{q_0, q_1, q_2\}$

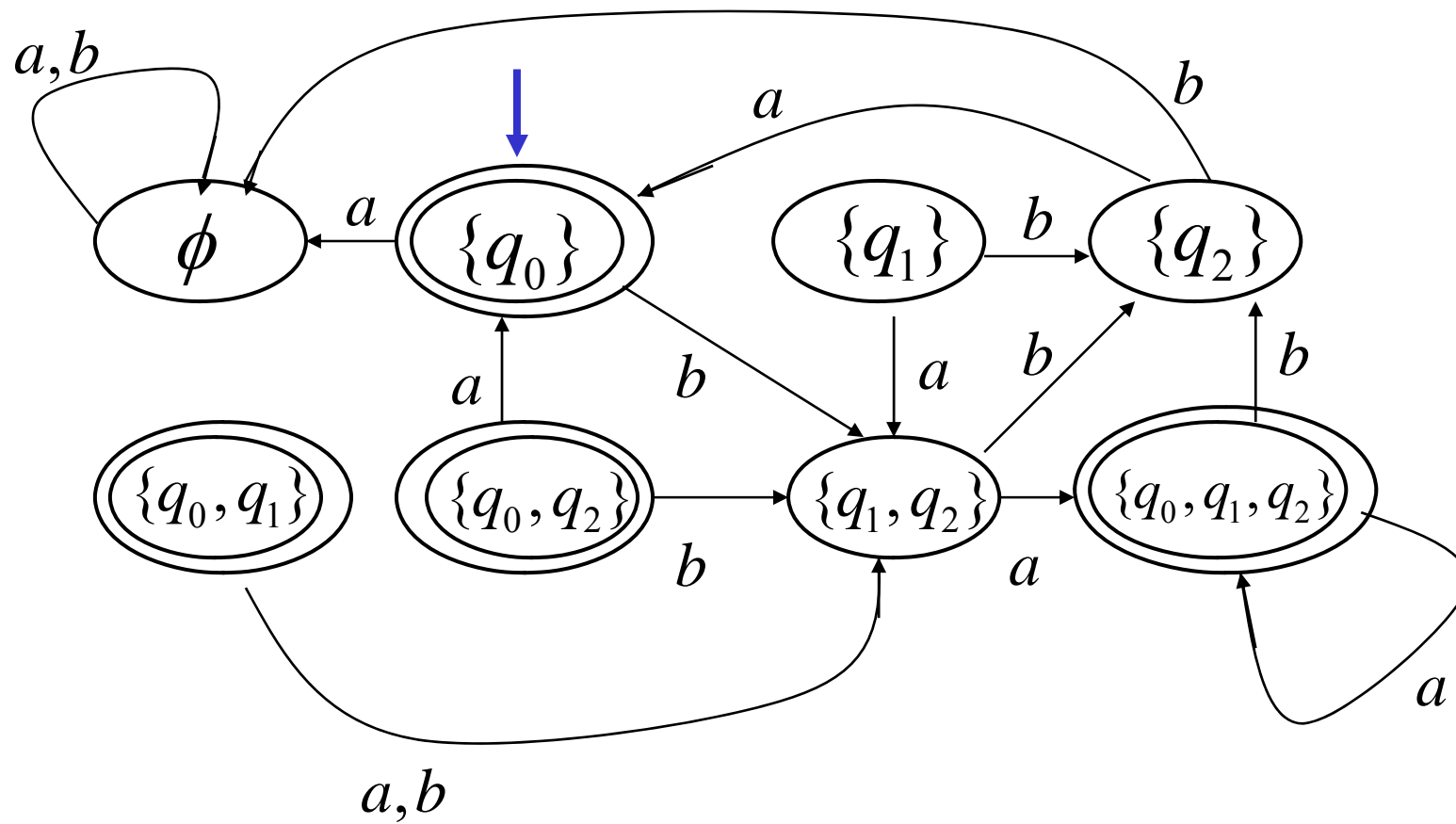
2. $q_0' = \{q_0\}$
3. $F' = \{q \in Q' \mid q \cap F \neq \emptyset\}$



$$4. \quad \delta'(q, s) = \bigcup_{r \in q} \{\delta(r, s)\}$$

| δ | a | b |
|----------|----------------|----------------|
| q_0 | ϕ | $\{q_1, q_2\}$ |
| q_1 | $\{q_1, q_2\}$ | $\{q_2\}$ |
| q_2 | $\{q_0\}$ | ϕ |

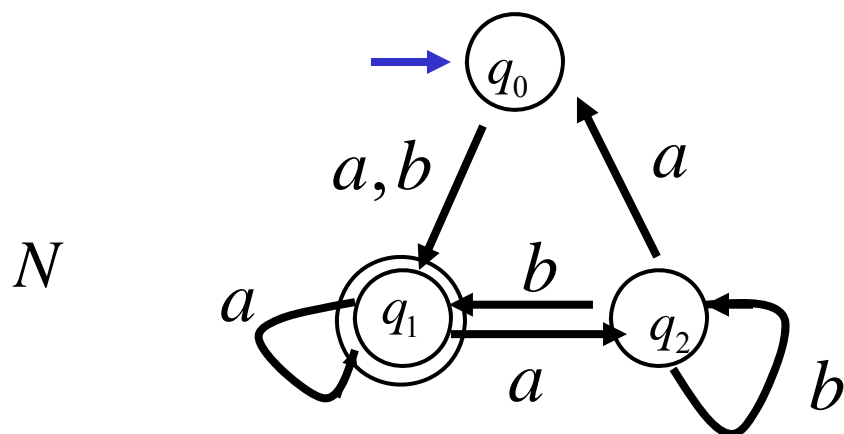
| δ' | a | b |
|---------------------|---------------------|----------------|
| ϕ | ϕ | ϕ |
| $\{q_0\}$ | ϕ | $\{q_1, q_2\}$ |
| $\{q_1\}$ | $\{q_1, q_2\}$ | $\{q_2\}$ |
| $\{q_2\}$ | $\{q_0\}$ | ϕ |
| $\{q_0, q_1\}$ | $\{q_1, q_2\}$ | $\{q_1, q_2\}$ |
| $\{q_0, q_2\}$ | $\{q_0\}$ | $\{q_1, q_2\}$ |
| $\{q_1, q_2\}$ | $\{q_0, q_1, q_2\}$ | $\{q_2\}$ |
| $\{q_0, q_1, q_2\}$ | $\{q_0, q_1, q_2\}$ | $\{q_1, q_2\}$ |



練習

$\Sigma = \{a, b\}$ とする。

NFA $N = \{\{q_0, q_1, q_2\}, \Sigma, \delta, q_0, \{q_1\}\}$



| δ | a | b |
|----------|----------------|----------------|
| q_0 | $\{q_1\}$ | $\{q_1\}$ |
| q_1 | $\{q_1, q_2\}$ | ϕ |
| q_2 | $\{q_0\}$ | $\{q_1, q_2\}$ |

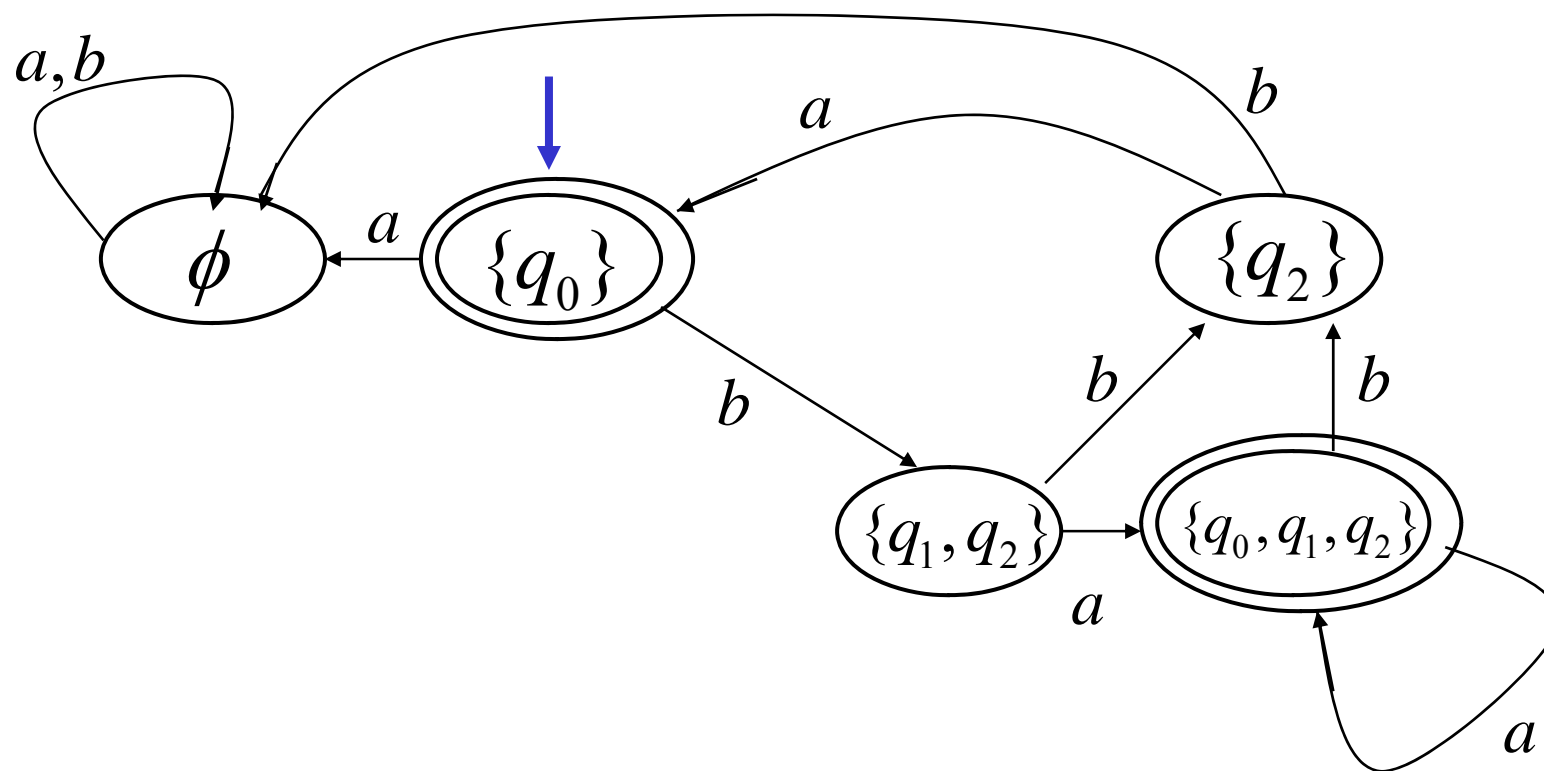
このNと同じ言語を受理するDFA

$$D = \{Q', \Sigma, \delta', q_0', \{q_0'\}, F'\}$$

を構成せよ。

補足

初期状態から、到達可能でない状態はDFAから削除できる。



NFA→DFAの証明

文字列 x の長さ $|x|$ に関する帰納法で

$$\delta'(q_0', x) = \{q_1, q_2, \dots, q_i\}$$

であるための必要十分条件が、

$$\delta(q_0, x) = \{q_1, q_2, \dots, q_i\}$$

であることを示す。

基礎

$$|x| = 0$$

このとき、 $x = \varepsilon$ であり、しかも

$$q_0' = \{q_0\} \text{ である。}$$

よって、明らかに成り立つ。

帰納

$|x| \leq m$ のとき成り立つと仮定する。

長さが $m + 1$ の文字列 xs ($s \in \Sigma$) に対して、
$$\delta'(q_0', xs) = \delta'(\delta'(q_0', x), s)$$

を考える。

帰納法の仮定より、

$$\delta'(q_0', x) = \{p_1, p_2, \dots, p_j\}$$

$$\Leftrightarrow \delta(q_0, x) = \{p_1, p_2, \dots, p_j\}$$

また、 δ' の定義から、

$$\delta'(\{p_1, p_2, \dots, p_j\}, s) = \{r_1, r_2, \dots, r_k\}$$

$$\Leftrightarrow \delta(\{p_1, p_2, \dots, p_j\}, s) = \{r_1, r_2, \dots, r_k\}$$

よって、

$$\begin{aligned}\delta'(q_0', xs) &= \{r_1, r_2, \dots, r_k\} \\ \Leftrightarrow \delta(q_0, xs) &= \{r_1, r_2, \dots, r_k\}\end{aligned}$$

また、

$$\delta'(q_0', x) \in F' \Leftrightarrow \delta(q_0, x) \cap F \neq \phi$$

である。

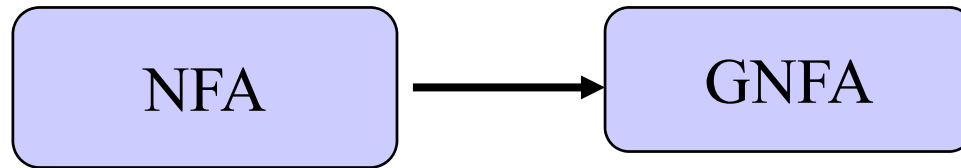
したがって、

$$L(D) = L(N)$$

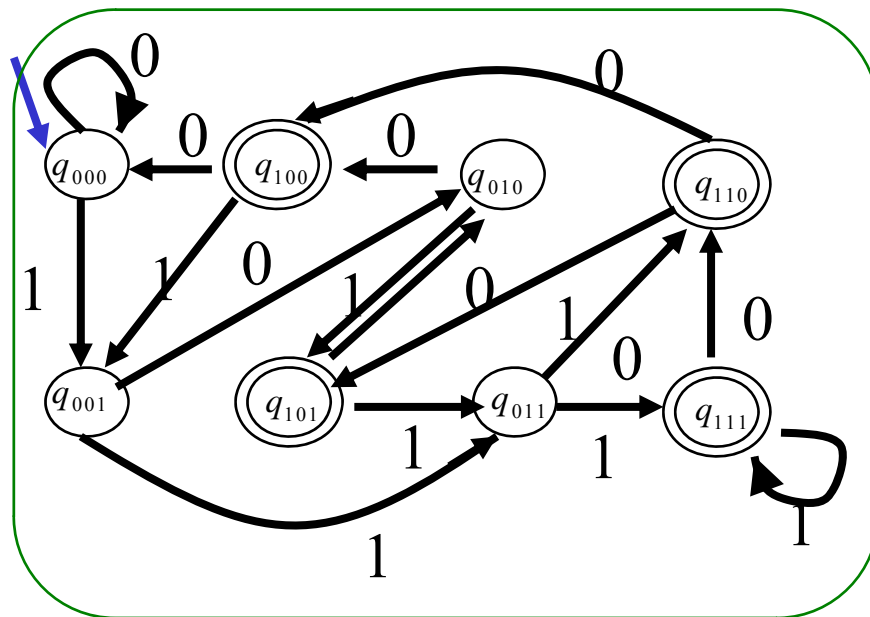
である。

QED

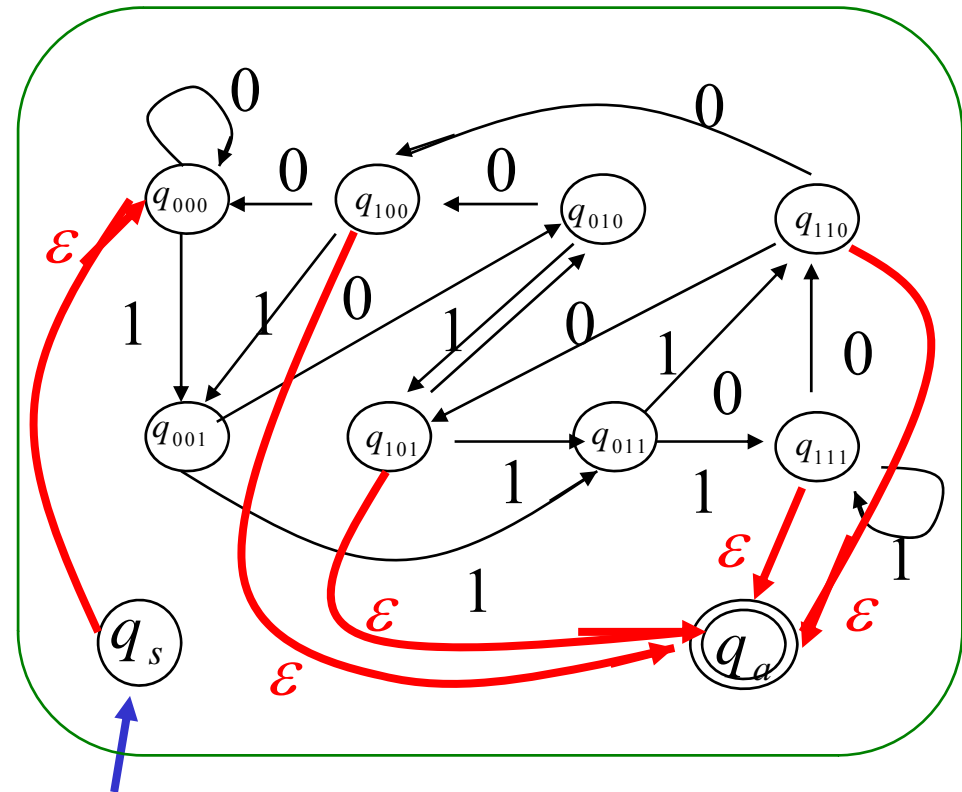
NFAから拡張NFAへ



NFAの受理状態が複数あるのに対して、GNFAの受理状態は一つである。



NFA



拡張NFA

ここでは、NFA→GNFAを形式的に示す。

任意のNFAを $N = (Q, \Sigma, \delta, q_0, F)$ とする。
このとき、 N と同じ言語を受理するGNFA
 $G = (Q', \Sigma, \delta', q_s, q_a)$ を構成する。

1. 状態の追加

$$Q' = Q \cup \{q_s, q_a\}$$

2. 状態遷移関数 δ' の決定

- ・通常の状態

$q \in Q, s \in \Sigma$ に対して、

$\delta(q, s) = P = \{q_x \mid q_x \text{は} q \text{から遷移可能な関数}\}$

のとき、

$q \in Q' - \{q_s, q_a\}, s \in \Sigma$ に対して、 $\delta'(q, q_x) = s$

- ・初期状態

$$\delta'(q_s, q_0) = \varepsilon$$

- ・受理状態

$q \in F$ に対して、

$$\delta'(q, q_a) = \varepsilon$$

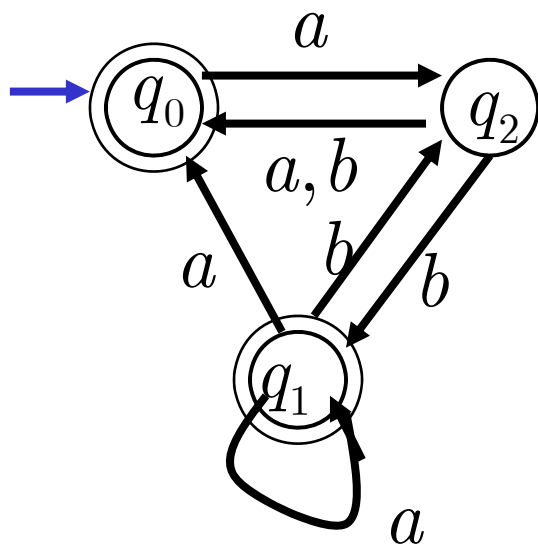
- ・上で定められていない定義域

$$\delta'(q, q') = \phi$$

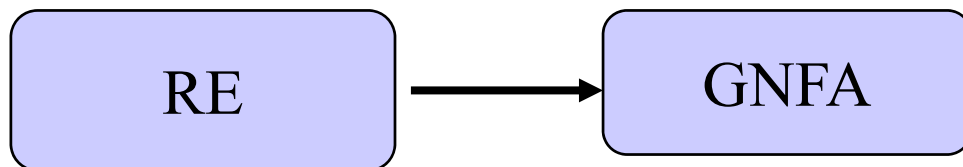
練習

$\Sigma = \{a, b\}$ とする。

次のNFAに対して、同じ言語を受理するGNFAを
状態遷移図と形式的定義の両方で与えよ。



正規表現からGNFAへ



方針:

任意の正規表現 r に対して、

$$L(r) = L(G)$$

となるような

$$G = (Q, \Sigma, \delta, q_s, q_t)$$

を構成する。

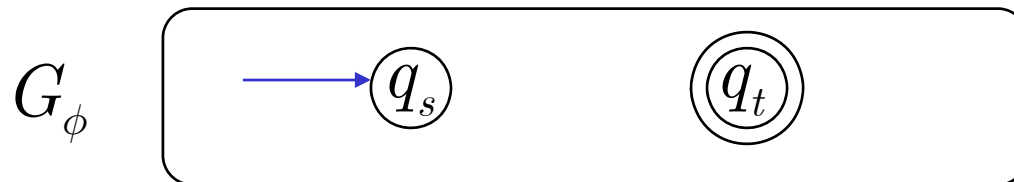
正規表現に基づいて、再帰的に構成していく。

演算数による帰納法

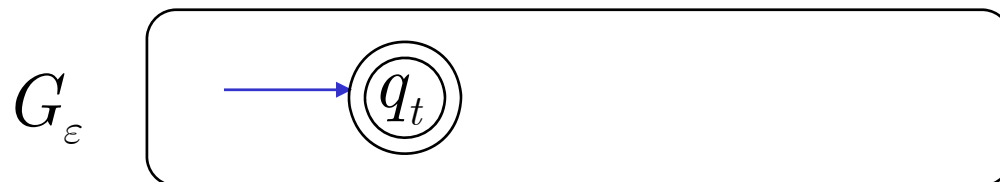
アルファベットを Σ とする。

基礎

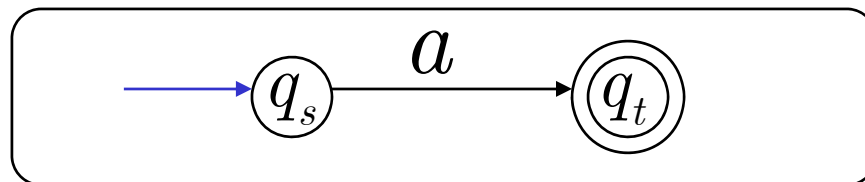
$r = \phi$ のとき、



$r = \varepsilon$ のとき、



$r = a(a \in \Sigma)$



帰納

演算数が m 以下のどんな正規表現 r' も対応するGNFAがあるとする。

演算数が $m + 1$ である正規表現 r を考える。

正規表現の定義より、3つの場合が存在する。

(1) 場合1

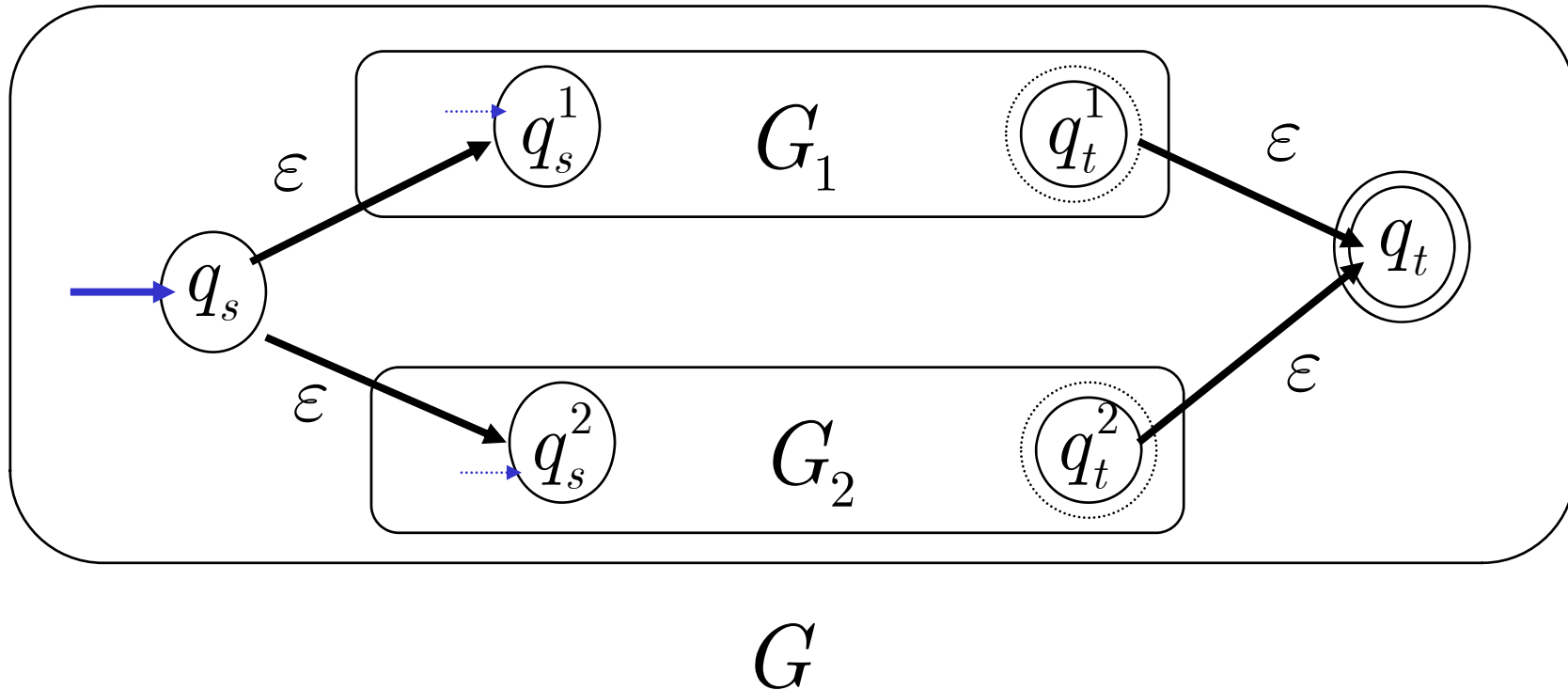
$r = r_1 + r_2$ の形にできるとき。

ここで、 r_1, r_2 は、正規表現だが、演算数は m 以下である。

よって、 $L(r_1) = L(G_1), L(r_2) = L(G_2)$

となる2つのGNFA G_1, G_2 が存在する。

これらを用いて $L(r)$ を受理するGNFA G を以下のように構成できる。



(2) 場合2

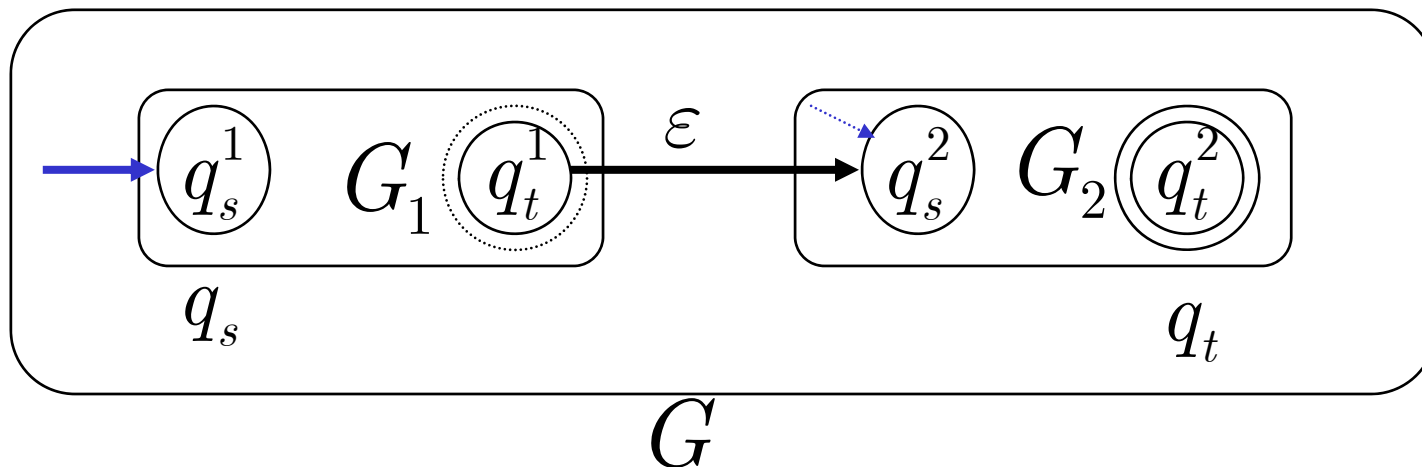
$r = r_1 r_2$ の形にできるとき。

ここで、 r_1, r_2 は、正規表現だが、演算数は m 以下である。

よって、 $L(r_1) = L(G_1), L(r_2) = L(G_2)$

となる2つのGNFA G_1, G_2 が存在する。

これらを用いて $L(r)$ を受理するGNFA G を以下のように構成できる。



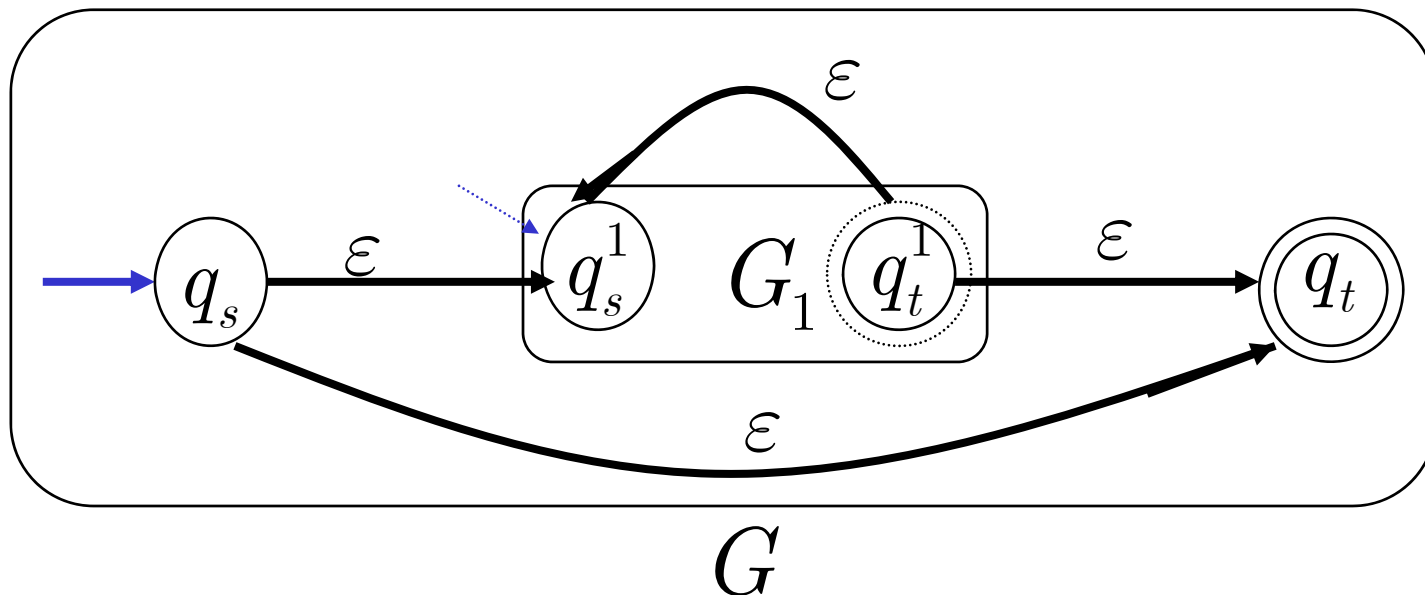
(3) 場合3

$r = r_1^*$ の形にできるとき。

ここで、 r_1 は、正規表現だが、
演算数は m 以下である。

よって、 $L(r_1) = L(G_1)$ なるGNFA G_1 が存在する。

これらを用いて $L(r)$ を受理するGNFA G を
以下のように構成できる。



以上より、
任意の正規表現はGNFAに変換可能である。

QED

例

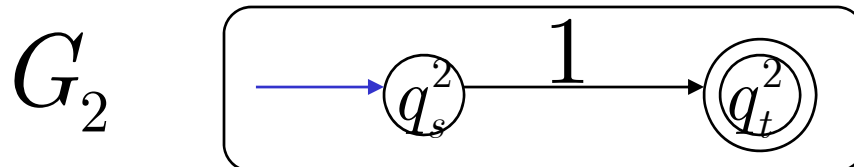
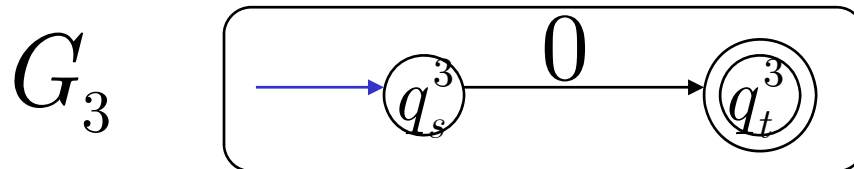
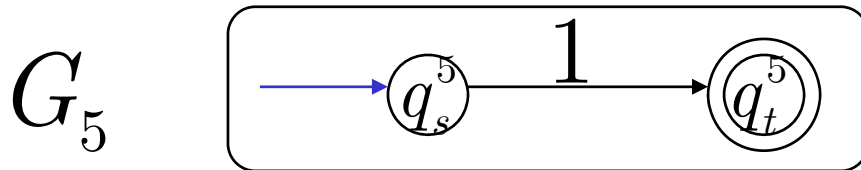
$\Sigma = \{0, 1\}$ 上の正規表現 $r = 01^* + 1$ を受理する GNFA G を構成する。

$r_1 = 01^*, r_2 = 1$ とおけば、 $r = r_1 + r_2$ である。

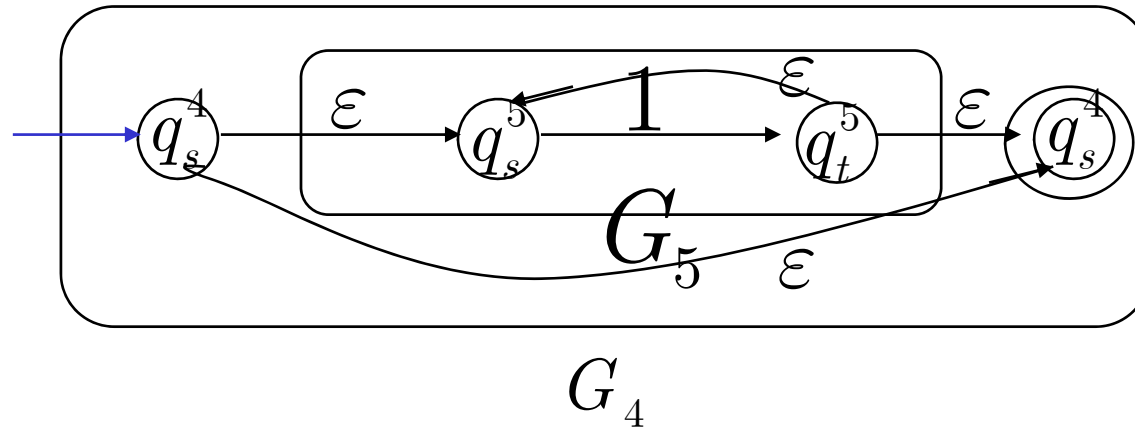
$r_3 = 0, r_4 = 1^*$ とおけば、 $r_1 = r_3 r_4$ である。

$r_5 = 1$ とおけば、 $r_4 = r_5^*$ である。

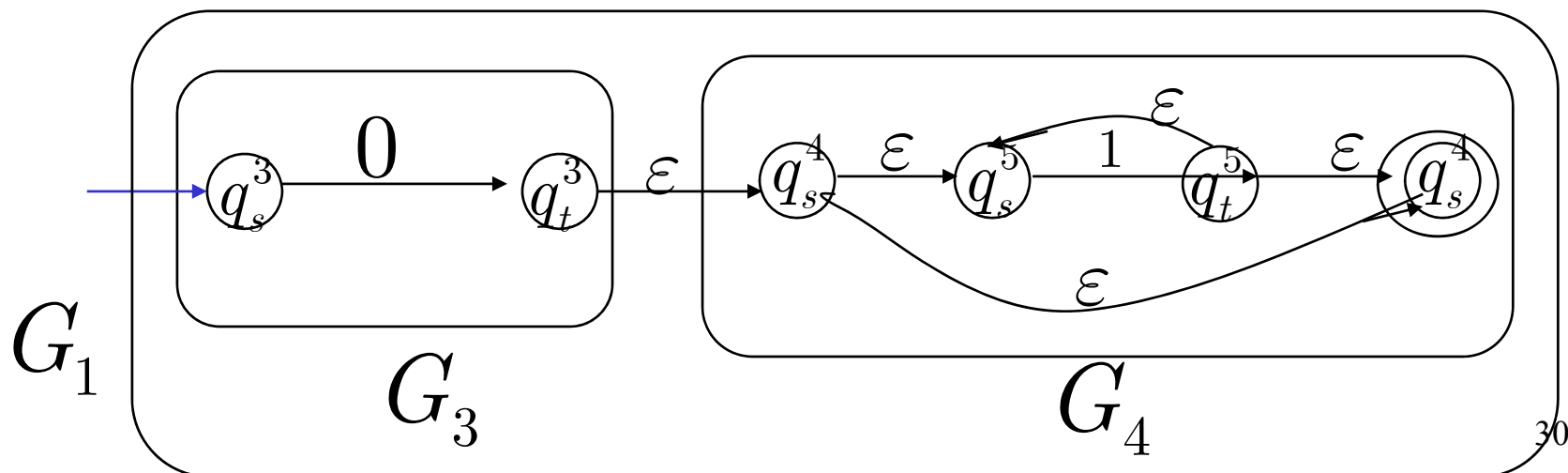
よって、まず基礎が以下のように構成できる。



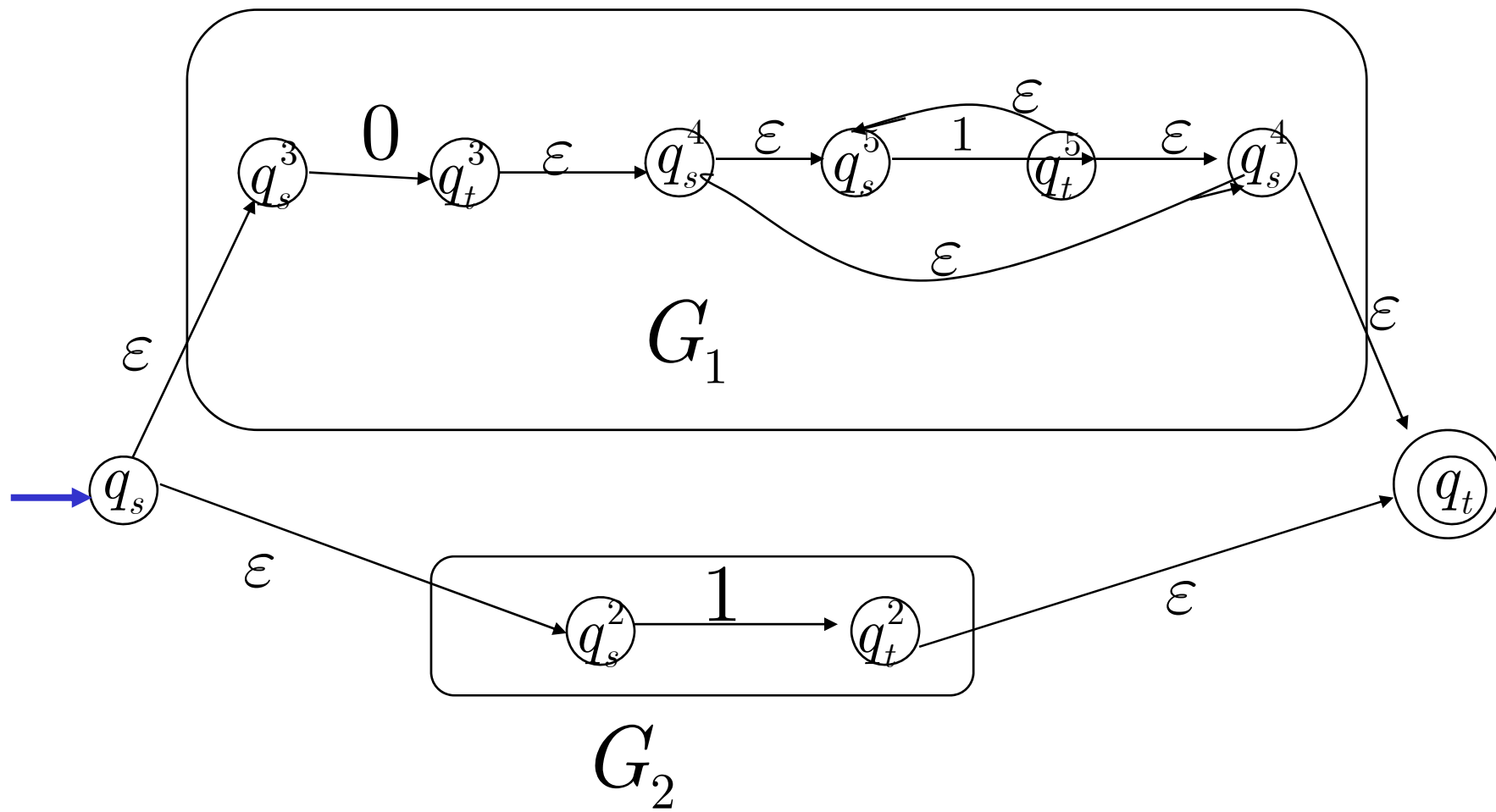
$r_4 = r_5^*$ を受理するGNFA G_4 は、 G_5 を用いて



$r_1 = r_3 r_4$ を受理するGNFA G_1 は、 G_4 を用いて



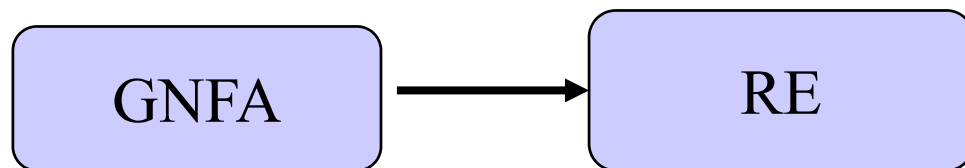
$r = r_1 + r_2$ を受理するGNFA G は、 G_1 を用いて



練習

$\Sigma = \{0, 1\}$ 上の正規表現 $r = (10 + 0)^*$ を受理する
GNFA G を構成せよ。

GNFAから正規表現へ



方針:

任意のGNFA $G = (Q, \Sigma, \delta, q_s, q_t)$ に対して、

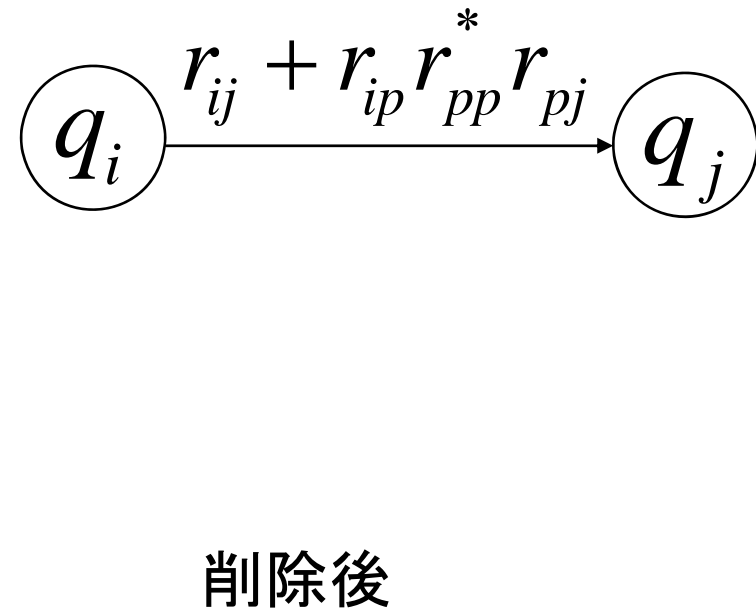
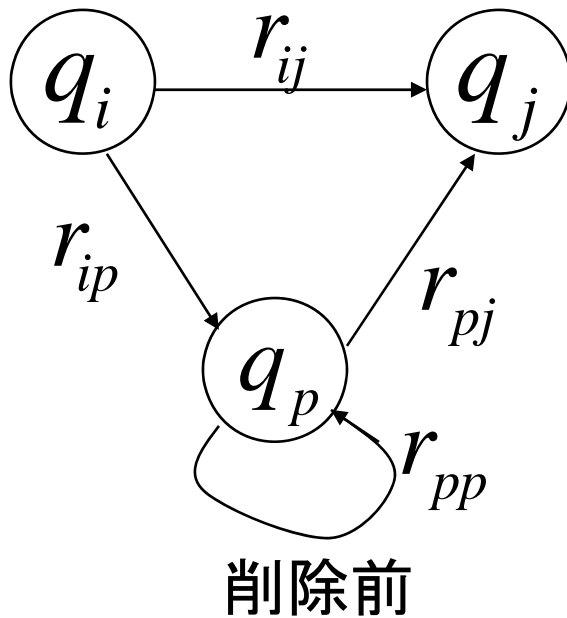
$$L(G) = L(r)$$

となるような r を導く。

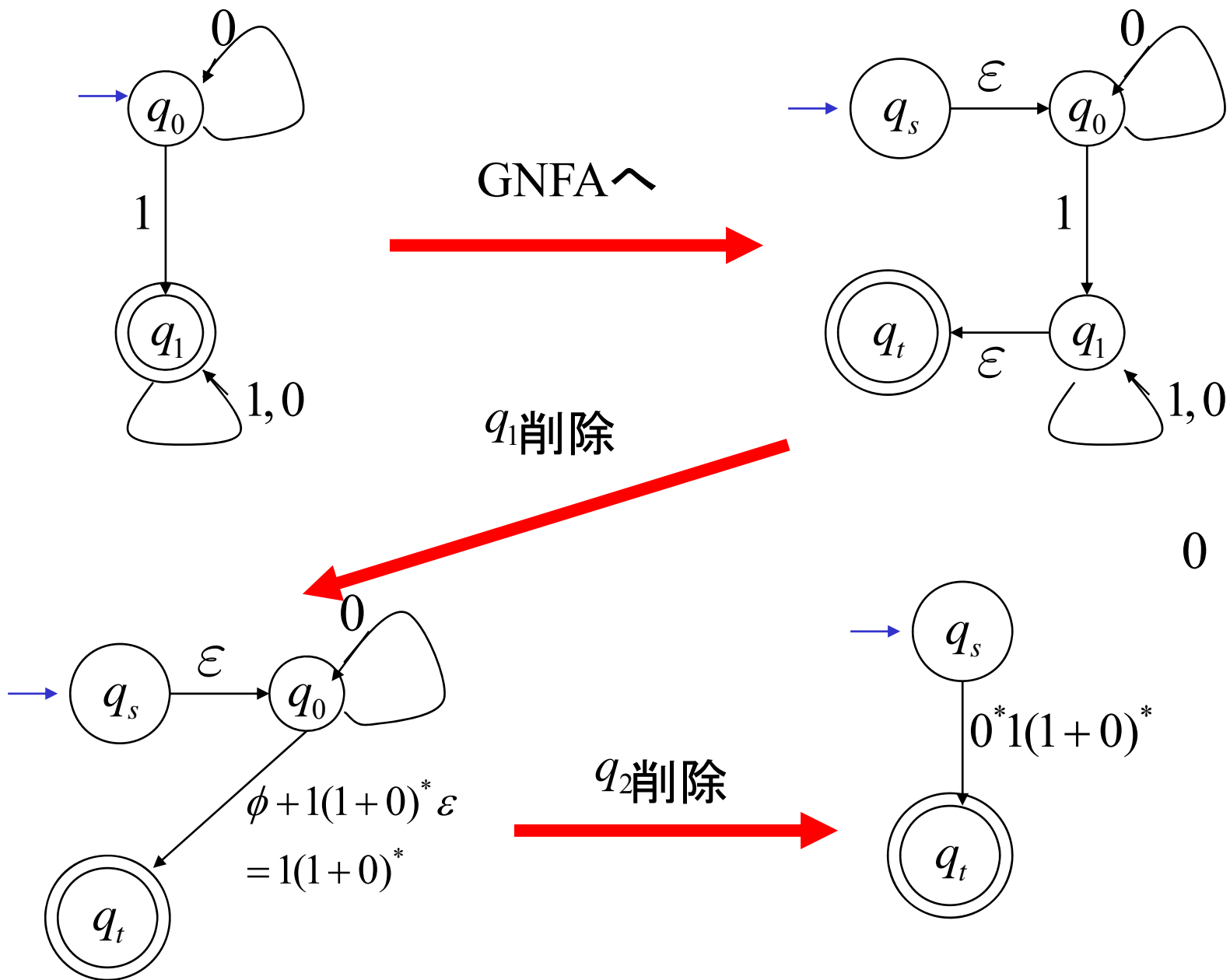
G の状態数を減少させることにより、
最終的には2状態のGNFAを構成していく。

状態の削除法

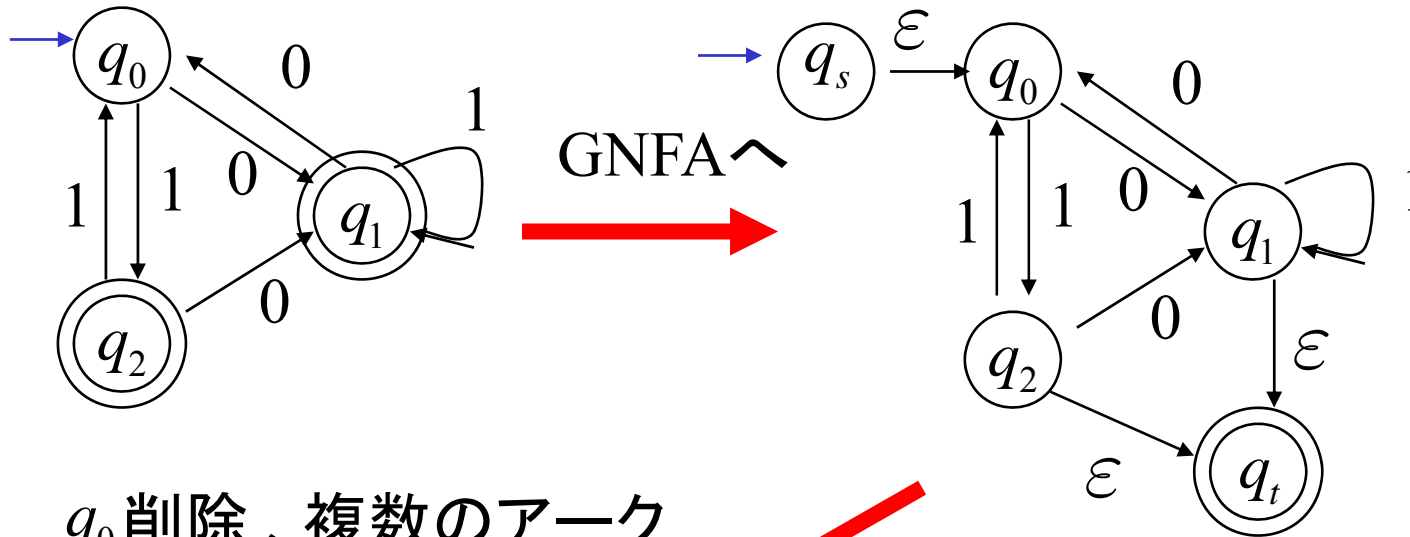
1. 削除する状態 $q_p \in Q$ を決める。
2. すべての組 $q_i, q_j \in Q - \{q_p\}$ に対して、次のようなアークを構成する。



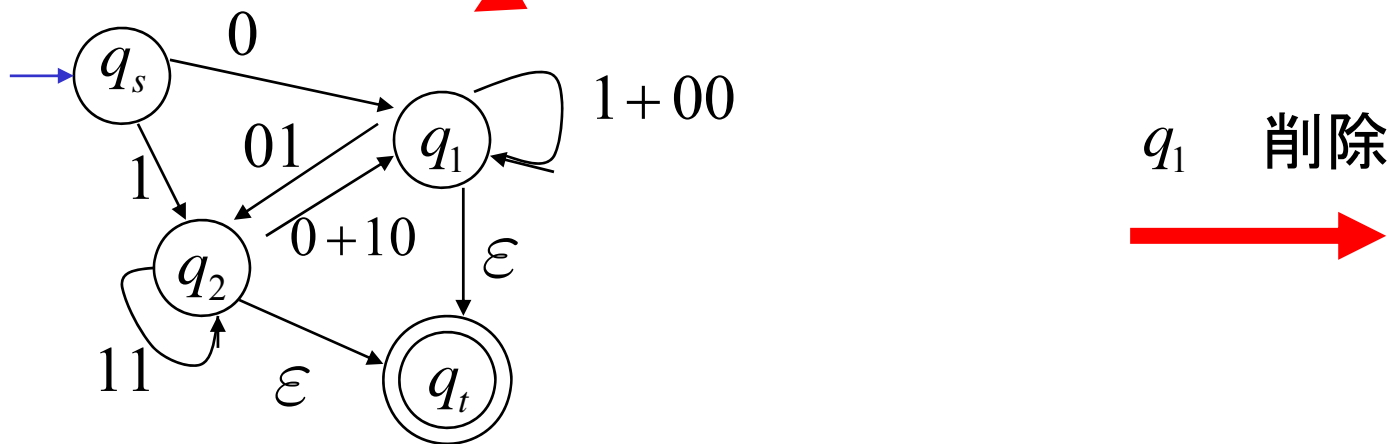
例

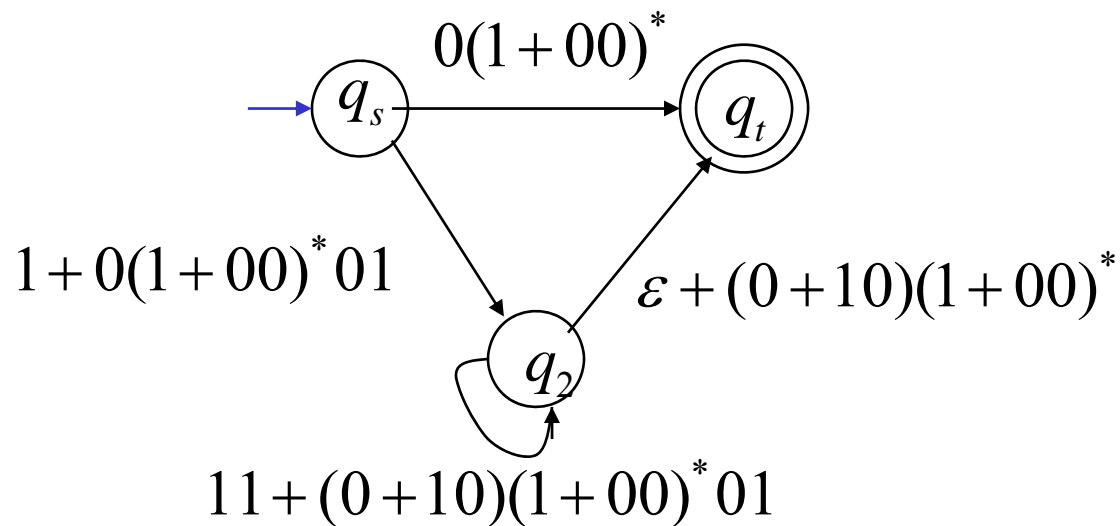



例2

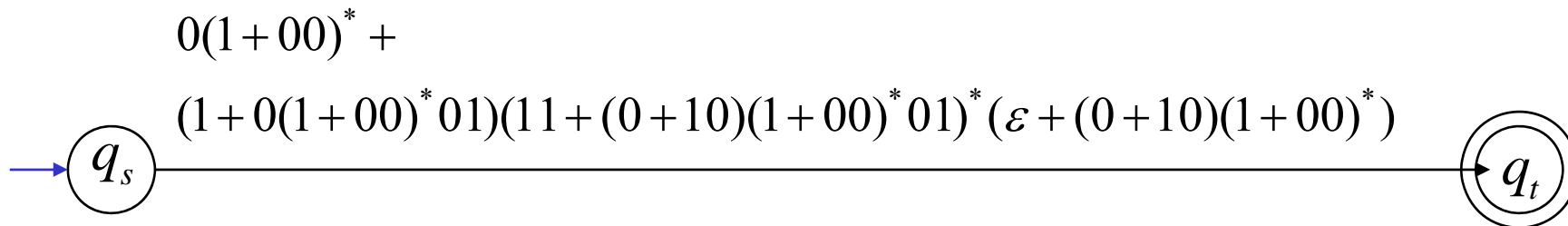


q_0 削除、複数のアークを生成することに注意。



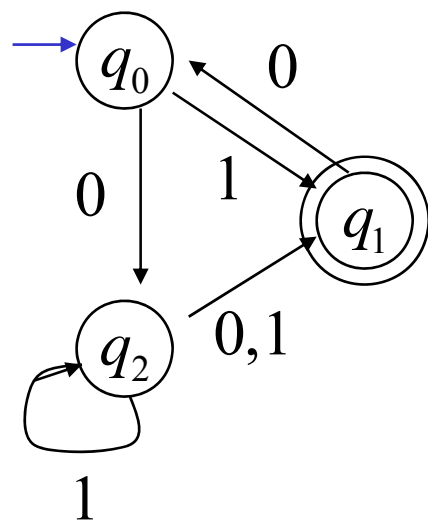



 q_2 消除



練習

次のDFAが受理する言語を正規表現で示せ。



2-2. 正規言語の性質

ここでは、DFAの限界を示す。

実際、次のような言語は、正規言語ではない。

$$B = \{0^n 1^n \mid n \geq 0\}$$

$$C = \{w \mid w \text{は } 0 \text{ と } 1 \text{ を同じ数だけ含む文字列}\}$$

正規言語でないことを示すための、有用な補題(定理)がある。

ポンピング補題

ポンピング補題

A が正規言語であるならば、ある数 p (ポンピング長)が存在して、 p より長い任意の文字列 $s \in A$ に対して、次を満たすように s を

$$s = xyz$$

に分割できる。

1. 各 $i \geq 0$ について、 $xy^iz \in A$
2. $|y| \geq 1, (y \neq \varepsilon)$
3. $|xy| \leq p$

ポンピング補題の意味

正規言語 A を認識するDFAを $M_A = (Q, \Sigma, \delta, q_0, F)$ とする。

実は、 p を M_A の状態数 ($p = |Q|$) とすると補題が成り立つ。

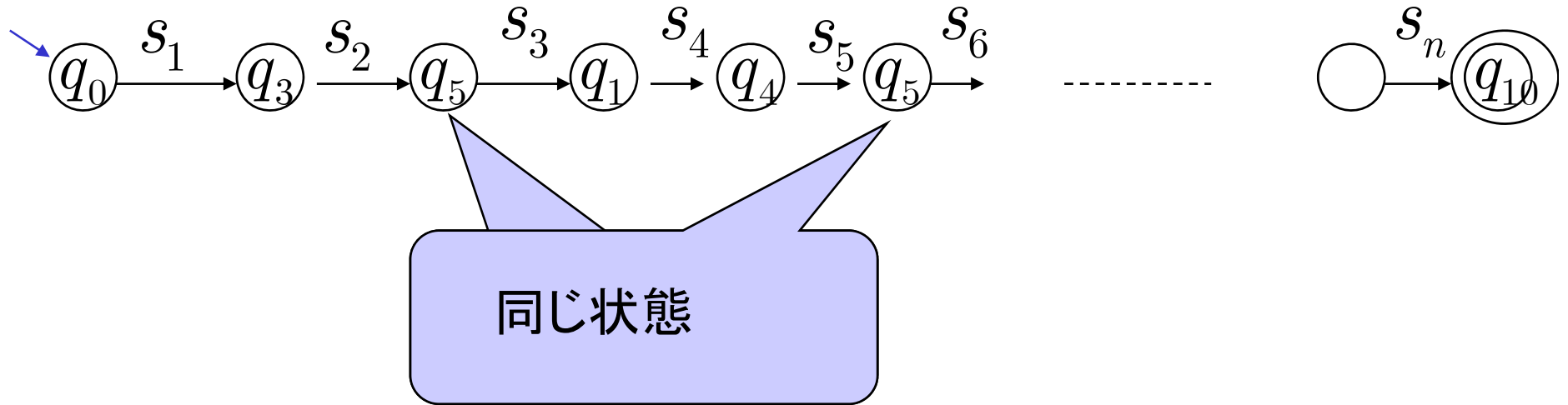
入力が状態数を超えているとき、必ず状態遷移中に2度以上おとづれる状態が存在しているはずである。

(このことは、鳩ノ巣原理と呼ばれます。)

例えば、 $s = s_1 s_2 s_3 s_4 \cdots s_n \in A$ による M_A の状態遷移が $q_0 q_3 q_5 q_1 q_4 q_5 \cdots q_{10}$ であったとする。

ここで、 $q_{10} \in F$ としています。

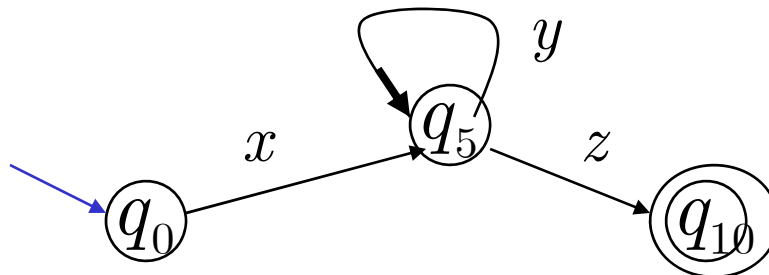
例の状態遷移



この例では、

$$x = s_1 s_2 \quad y = s_3 s_4 s_5 \quad z = s_6 \cdots s_n$$

とすればよい。このとき、 $xy^i z \in A$ である。



ポンピング補題の証明

正規言語 A を認識するDFAを $M_A = (Q, \Sigma, \delta, q_0, F)$ とし、
 $p = |Q|$ とする。

また、 $n \geq p$ とし、 $s = s_1 s_2 s_3 s_4 \cdots s_n \in A$ とする。

s を入力としたときの、 M_A の状態遷移の系列を
 $r_1 r_2 \cdots r_{n+1}$ とする。すなわち、 $1 \leq i \leq n$ に対して、
 $r_{i+1} = \delta(r_i, s_i)$

である。

鳩ノ巣原理より、列 $r_1 r_2 \cdots r_{n+1}$ の最初の $p + 1$ 個の中に
同じ状態が2回以上現れる。

2回現れた状態の一回目を r_j とし、
2回目を r_k とする。

ここで、

$$x = s_1 \cdots s_{j-1} \quad y = s_j \cdots s_{k-1} \quad z = s_k \cdots s_n$$

とおく。

このとき、補題の3条件をすべて満たしている。

QED

非正規な言語

次の言語は正規言語ではない。

$$B = \{0^n 1^n \mid n \geq 0\}$$

$$\begin{aligned} B &= \{w \mid w \text{は } 0 \text{ の列のあとに同じ長さの } 1 \text{ の列が続く}\} \\ &= \{\varepsilon, 01, 0011, 000111, 00001111, \dots\} \end{aligned}$$

証明

ポンピング補題を用いて背理法で証明する。

Bは正規言語であると仮定する。(背理法の仮定)

Bは正規言語であるので、あるポンプ長 p が存在する。

$s = 0^p 1^p$ とする。

このとき、ポンピング補題より、

$$s = xyz$$

と分割できる。

このとき、 y として次の3つの場合が考えられる。

(1) 0だけ (2) 1だけ (3) 0の列と1の列の連結

しかし、次のようにいずれの場合も矛盾が生じる。

(1) y が0だけのとき

このとき、ポンピング補題より、

$xy^2z = xyyz$ も B に含まれなければならない。

($xyyz \in B$)

しかし、 $xyyz$ は0の数が多いので $xyyz \notin B$ であり、
矛盾が生じる。

(2) y が1だけのとき

(1)と同様に矛盾が導ける。

(3) y が0の列と1の列の連結のとき

$y = 0^j1^k$ とする。

このとき、 $xyyz = x0^j1^k0^j1^kz$ であるが、

1の前に0があり $xyyz \notin B$ である。

以上、すべての場合で矛盾が生じるので、
 B は正規言語ではない。

QED ⁴⁷