

# 正規表現

正規言語を表すための便利な方法

例:

$$\begin{aligned}ab^* &= \{a\}\{b\}^* \\ (a + ab^*)^* + b &= (\{a\} \cup \{a\}\{b\}^*)^* \cup \{b\}\end{aligned}$$

UNIX のコマンド grep, awk, perl で使う正規表現とは、表記法が若干違うが、本質的に同じ概念である。

## 正規表現の定義

言語のアルファベット  $\Sigma$  を  $\{a, b, c, \dots, z\}$  とする  $\alpha, \beta$  を正規表現とし、それぞれ言語  $L(\alpha), L(\beta)$  を表すとする

| 正規表現                       | 対応する言語                    |
|----------------------------|---------------------------|
| $0$                        | 空集合                       |
| $1$                        | $\{\epsilon\}$            |
| $a$ (アルファベット $\Sigma$ の要素) | $\{a\}$                   |
| $(\alpha)$                 | $L(\alpha)$               |
| $\alpha^*$                 | $L(\alpha)^*$             |
| $\alpha \cdot \beta$       | $L(\alpha)L(\beta)$       |
| $\alpha + \beta$           | $L(\alpha) \cup L(\beta)$ |

正規表現の長さは常に有限とする

$\alpha \cdot \beta$  は通常  $\alpha\beta$  と省略する

演算の優先順位は上の表で上にあるほど高い。例えば

$$\alpha + \beta\gamma^* = \alpha + (\beta(\gamma)^*)$$

$$\alpha + \beta\gamma^* \neq ((\alpha + \beta)\gamma)^*$$

である

## 正規表現で表せる言語は正規言語

| 正規表現                       | 対応する言語         |
|----------------------------|----------------|
| 0                          | 空集合            |
| 1                          | $\{\epsilon\}$ |
| $a$ (アルファベット $\Sigma$ の要素) | $\{a\}$        |

上記の言語が正規言語であることはあきらか

$\alpha, \beta$ : 正規表現

$L(\alpha), L(\beta)$ :  $\alpha, \beta$  で表される言語

$M_\alpha, M_\beta$ :  $L(\alpha) = T(M_\alpha), L(\beta) = T(M_\beta)$  となるオートマトン

$L(\alpha^*) = L(\alpha)^*$  を受理する  $\epsilon$ -NFA を  $M_\alpha$  から構成できる

$L(\alpha\beta) = L(\alpha)L(\beta)$  を受理する  $\epsilon$ -NFA を  $M_\alpha$  と  $M_\beta$  から構成できる

$L(\alpha + \beta) = L(\alpha) \cup L(\beta)$  を受理する  $\epsilon$ -NFA を  $M_\alpha$  と  $M_\beta$  から構成できる

上記の事実から、すべての正規表現について、それが表す言語を受理する  $\epsilon$ -NFA が存在する。したがって正規表現で表される言語は正規言語である。

## 正規言語は正規表現で表せる

$M = (\{q_1, \dots, q_n\}, T, t, q_1, F)$  を NFA とする

$R_{i,j}^k$  を  $q_i$  から  $q_j$  に途中  $q_1$  から  $q_k$  だけを経由して遷移する記号列の集合を表す正規表現とする

$T(M)$  を表す正規表現は  $F = \{q_{\ell_1}, \dots, q_{\ell_m}\}$  なら

$$R_{1,\ell_1}^n + R_{1,\ell_2}^n + \dots + R_{1,\ell_m}^n$$

である。なぜなら、 $R_{1,\ell_1}^n$  は出発状態から終了状態の一つ  $q_{\ell_1}$  に至る記号列を表す正規表現だから

$q_i$  を  $q_j$  に直接遷移させる記号が  $a_1, \dots, a_m$  であるとする

$$\begin{aligned} R_{i,j}^0 &= a_1 + a_2 + \dots + a_m \quad (\text{if } i \neq j), \\ R_{i,j}^0 &= 1 + a_1 + a_2 + \dots + a_m \quad (\text{if } i = j) \end{aligned}$$

$$R_{i,j}^{k+1} = R_{i,j}^k + R_{i,k+1}^k (R_{k+1,k+1}^k)^* R_{k+1,j}^k$$

$\{q_1, \dots, q_{k+1}\}$  を経由して  $q_i$  から  $q_j$  に至る記号列の集合は、 $\{q_1, \dots, q_k\}$  を経由して  $q_i$  から  $q_j$  に至る記号列の集合と、

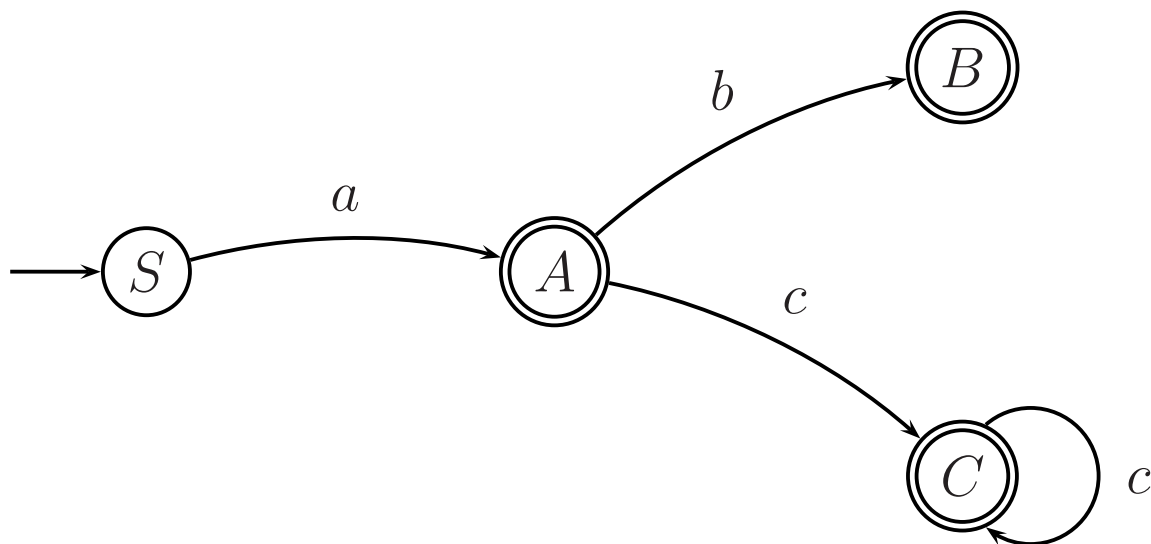
$\{q_1, \dots, q_k\}$  を経由して  $q_i$  から  $q_{k+1}$  に至り、 $q_{k+1}$  を1回以上経由して、 $q_j$  に至る記号列の集合の和集合に等しい

注意：前述の方法で作成した正規表現は不必要に長い  
が、短い正規表現をオートマトンから作る方法は参考  
書「オートマトン 言語理論 計算論 I」の 3.2.2 節に書  
いてある

## 演習問題

演習問題 36 正規表現  $a(a + b^*a) + bb^*$  を受理する状  
態数最小の DFA のグラフを書け (どのような方法で  
答えを導いてもよいが、答えの根拠を明示すること)

演習問題 37 以下のオートマトンが受理する言語を表  
す正規表現を書け (どのような方法で答えを導いても  
よいが、答えの根拠を明示すること)



演習問題 38 今日の授業でわかりにくい所や要望を書  
いて下さい

## 演習問題解説

前回の DFA の状態数を最小化する問題は、最初から状態数最小なので、与えられたオートマトンが答えである。ダミー状態を付け加えないで、区別できる状態の組を調べると間違った答えが出る。4 点満点