4. 正則言語の性質(1):
(テキスト4.1,4.2)

4.1. 言語が正則でないことの証明

– 有限オートマトンは状態が有限個しかない。

→「有限個の状態しかないと区別できないもの」は区別できない。

(典型的な)鳩ノ巣原理(Pigeon Hole Principle):

n+1羽(以上)の鳩が n 個の巣に入っている。

このとき、どこかの巣には鳩が2羽以上入っている。

4. 正則言語の性質(1):
(テキスト4.1,4.2)

4.1. 言語が正則でないことの証明

例: 言語 $L=\{0^n1^n \mid n \geqq 1\}$

- n はどんなに大きくてもよい
- DFA A が m 状態なら、$n>m$ のときに 0^n1^n に関して A のふるまいは…?

例: 言語 $L=\{0^n1^n \mid n \geqq 1\}$ は正則ではない。

証明: L が正則であったと仮定して、矛盾を導く。

L は正則なので、L を受理する DFA A が存在する。A の状態集合を $q_1,q_2,\ldots,q_m$ とする(mは有限)。$n=m+1$のとき、鳩ノ巣原理から、

$$0,00,0^3,0^4,\ldots,0^n$$

の中には、「Aが遷移したときに同じ状態になる、長さの異なるペア」が存在する。これらを $0^i,0^j$ とおく。つまり A は$0^i,0^j$ のどちらを読み込んだときも同じ状態 q になる。

ここで入力0^i1^j を考える。$i \neq j$なので、これは L の要素ではない。しかし A は入力0^i1^jと入力0^j1^jを区別できない。したがって、両方とも受理するか、両方とも受理しないか、どちらかしかできない。これは A が L を受理する、という仮定に反する。

したがって L は正則ではない。

4. 正則言語の性質(1)
(テキスト4.1,4.2)

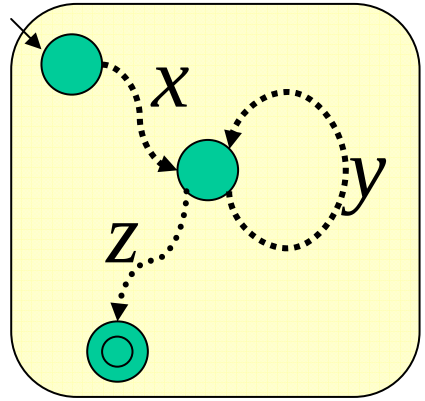
ある言語が正則でないことを示すのに使う標準的な補題

4.1. 言語が正則でないことの証明

正則言語に対する反復補題(Pumping Lemma):

– 正則言語 L に対し、以下の条件を満たす定数 n が存在する: $|w| \geqq n$ を満たす任意の文字列 $w \in L$ は、次の条件を満たす3個の部分列 $w = xyz$ に分解できる。

1. $y \neq \varepsilon$
2. $|xy| \leqq n$
3. すべての $k \geqq 0$ に対し、$xy^kz \in L$



4.1. 言語が正則でないことの証明

反復補題(Pumping Lemma):

- 正則言語 L に対し、以下の条件を満たす定数 n が存在する: $|w| \geqq n$ を満たす任意の文字列 $w \in L$ は、次の条件を満たす3個の部分列 $w = xyz$ に分解できる。

  (1) $y \neq \varepsilon$ (2) $|xy| \leqq n$ (3) $xy^kz \in L$ $(k \geqq 0)$

[証明] Lは正則言語なので、$L(A)=L$であるDFA Aが存在する。A の状態数を n とする。

長さ n 以上の L に属する任意の文字列 $w=a_1a_2\ldots a_m$ を考える。$(m \geqq n)$

A は文字列 $a_1a_2\ldots a_i$ を処理したあと、状態 p_i になるとする。(初期状態をq_0とすると$p_0=q_0$)

4.1. 言語が正則でないことの証明

反復補題(Pumping Lemma):

- 正則言語 L に対し、以下の条件を満たす定数 n が存在する: $|w| \geqq n$ を満たす任意の文字列 $w \in L$ は、次の条件を満たす3個の部分列 $w = xyz$ に分解できる。

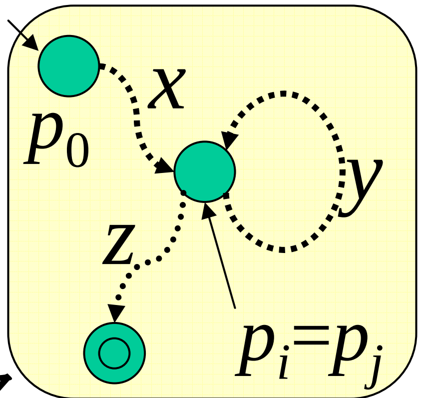
$$(1)\ y \neq \varepsilon \quad (2)\ |xy| \leqq n \quad (3)\ xy^k z \in L\ (k \geqq 0)$$

[証明] A は文字列 $a_1 a_2 \ldots a_i$ を処理したあと、状態 p_i になるとする。(初期状態をq_0とすると$p_0=q_0$)
鳩ノ巣原理により、$p_0, p_1, \ldots, p_m$ の中には同じ状態 p_i, p_j が存在する。($i<j$ としてよい)

- $x = a_1, a_2, \ldots, a_i$
- $y = a_{i+1}, \ldots, a_j$
- $z = a_{j+1}, \ldots, a_m$

$x=\varepsilon$ や$z=\varepsilon$ はありえるが$y \neq \varepsilon$

と定義すると A は $xy^k z$ $(k \geqq 0)$を受理する。



例: 言語 $L=\{0^n1^n \mid n \geqq 1\}$ は正則ではない。

反復補題による証明: L が正則であると仮定して、矛盾を導く。

L は正則なので、反復補題より、以下の条件を満たす定数 m が存在する: $|w| \geqq m$ を満たす任意の文字列 $w \in L$ は、次の条件を満たす3個の部分列 $w = xyz$ に分解できる。

(1) $y \neq \varepsilon$ (2) $|xy| \leqq m$ (3) $xy^kz \in L$ $(k \geqq 0)$

ここで文字列$w=0^m1^m$を考える。wを上記の条件を満たすような部分列xyzに分解する。$|xy| \leqq m$, $y \neq \varepsilon$ なので、$y=0^i$ $(i \geqq 1)$ となる。

$xyz = 0^m1^m$ なので $xyyz = 0^{m+i}1^m$ である。反復補題から、$xyyz \in L$ となるが、実際には $xyyz \notin L$ であるので矛盾。

したがって L は正則ではない。

4. 正則言語の性質(1):
(テキスト4.1,4.2)

4.2. 正則言語に関する閉包性

- 閉包性…集合/言語が演算に関して閉じていること。
  - 正則言語にある操作/演算を加えて、新しい言語を作ったとき、それがまた正則になっているなら、
    - 正則言語はその操作/演算に関して閉じている

    という。この性質を閉包性という。

4.2. 正則言語に関する閉包性

– 正則言語は以下の閉包性を持つ。

① 正則言語 L_1, L_2 について $L_1 \cup L_2$ は正則

② L_1, L_2 について $L_1 \cap L_2$ は正則

③ 正則言語の補集合は正則

④ L_1, L_2 について $L_1 - L_2$ は正則

⑤ 正則言語の反転は正則

⑥ L_1 について $L_1{}^*$ は正則

⑦ L_1, L_2 の連接は正則

⑧ 正則言語の準同型の像は正則

⑨ 正則言語の逆準同型の像は正則

正則言語における4つの証明手法

この授業では範囲外

4.2. 正則言語に関する閉包性

① 正則言語 L_1, L_2 について $L_1 \cup L_2$ は正則

[証明手法1] 正則表現を使ったもの

L_1, L_2は正則言語なので、$L(E_1)=L_1, L(E_2)=L_2$ を満たす正則表現が存在する。$((E_1)+(E_2))$は正則表現で、かつ明らかに $L(((E_1)+(E_2)))=L_1 \cup L_2$ が成立する。

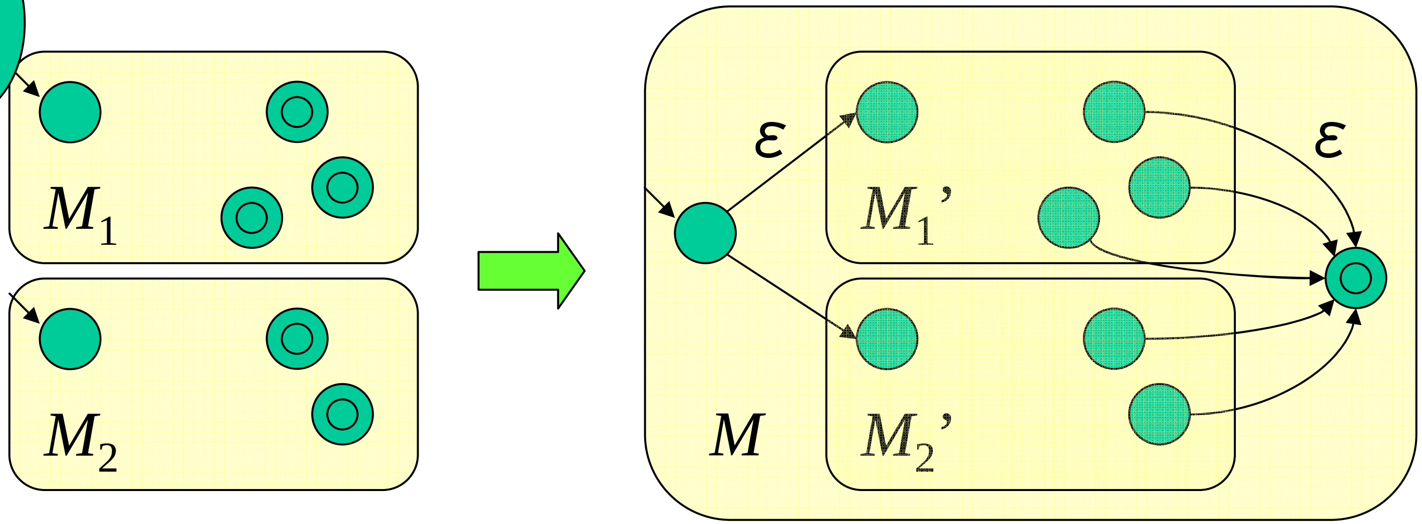
4.2. 正則言語に関する閉包性

① 正則言語 L_1, L_2 について $L_1 \cup L_2$ は正則

[証明手法2]オートマトンを使ったもの

L_1, L_2 は正則言語なので、$L(M_1)=L_1$, $L(M_2)=L_2$を満たすDFA M_1, M_2が存在する。以下に示す方法で構成したε-NFA Mは明らかに$L_1 \cup L_2$を受理する。

証明はもっと厳密に記述する



4.2. 正則言語に関する閉包性

③ 正則言語の補集合は正則

[補集合とは] 言語 L の補集合 $\overline{L}$={ w | $w \notin L$}

[証明](手法2)

言語 L が正則なら、L を受理するDFA $A=(Q, \Sigma, \delta, q, F)$ が存在する。このとき、A の受理状態とそれ以外を入れ替えた DFA $\overline{A}=(Q, \Sigma, \delta, q, Q-F)$ は $\overline{L}$ を受理する。

4.2. 正則言語に関する閉包性

② L_1, L_2 について $L_1 \cap L_2$ は正則

$$\overline{A \cup B} = \overline{A} \cap \overline{B}$$

$$\overline{A \cap B} = \overline{A} \cup \overline{B}$$

[証明手法3]

ド・モルガンの定理より、

$$L_1 \cap L_2 = \overline{\overline{L_1 \cap L_2}} = \overline{\overline{L_1} \cup \overline{L_2}}$$

したがって L_1, L_2 が正則なら①,③より、
$L_1 \cap L_2$ も正則

4.2. 正則言語に関する閉包性

④ L_1, L_2 について $L_1 - L_2$ は正則
($L_1 - L_2 = L_1 \cap \overline{L_2}$ なので手法3でもOK)

[証明手法4(直積構成法)]

① L_1, L_2 を受理する DFA を M_1, M_2 とする。

② $L_1 - L_2$を受理するDFA Mは、入力を読みながら、

- その入力に対する M_1 の状態遷移
- その入力に対する M_2 の状態遷移

を同時に模倣する。

③ 入力を読み終えた時点で M_1 が受理かつ M_2 が受理でないなら M は受理。

4.2. 正則言語に関する閉包性

⑤ 正則言語の反転は正則

[定義]

文字列 $w=x_1x_2\ldots x_k$ の反転(Reverse) $w^R=x_k\ldots x_2x_1$

言語 L の反転 $L^R=\{ w \mid w^R \in L\}$

A: DFA

A^R: ε-NFA

[証明]

L を受理するDFA A に対し、

① Aの受理状態を一つにし、

② Aの遷移をすべて逆転し、

③受理状態と初期状態を入れ替えた

ε-NFA A^RはL^Rを受理する。

4.2. 正則言語に関する閉包性

⑥ L_1 について L_1* は正則

⑦ L_1, L_2 の連接は正則

L_1, L_2 を表現する正則表現 E_1, E_2 に対し、

⑥ (E_1)*

⑦ $(E_1)(E_2)$

でOK.

4. 正則言語の性質(1):
演習問題(4)

[問題] 次の $\Sigma=\{0,1\}$上の言語 L が正則でないことを証明せよ。鳩ノ巣原理を直接使った証明でも、反復補題を使った証明でもよい。

$$L=\left\{ww^R \mid w\in\Sigma^*\right\}$$

[余談] 言語 L は日本語では「回文」英語では ``palindrome'' と呼ばれる。

例: たけやぶやけた、Madam, I'm Adam.