

計算理論 第5回

1. 有限オートマトン
2. 正則表現と正則言語
3. 正則言語の性質

- ■ 言語が正則でないことの証明
■ 正則言語に関する閉包性

テキスト
4. 1~4. 2節

4. 文脈自由文法と言語
5. プッシュダウン・オートマトン
6. 文脈自由言語の性質
7. チューリングマシン

3

本日の学習目標

- 正則言語に対する反復補題とその根拠を説明できる
- 反復補題を用いて、正則言語でないことの証明ができる
- 与えられた言語が、正則言語かそうでないか見当がつく
- 正則言語についての3つ以上の閉包性を、例を用いて説明できる
- 有限オートマトンまたは正則表現を適宜使い分けて、正則言語の閉包性を証明できる

4

4. 正則言語の性質

4.1 言語が正則でないことの証明

- 正則言語でない言語は存在するのか？
 - 正則表現で表現できない言語はあるのか？
 - 有限オートマトンで認識できない言語はあるのか？
- 答えはYES
 - 正則言語でない言語が存在する
 - Σ 上の正則言語の数：加算無限
 - Σ 上の言語の数：非加算無限
 - 例： $L_{01} = \{0^n 1^n \mid n \geq 1\}$ は正則言語でない
- 正則言語でないことをどのように証明するか：本日の内容

5

4.1.1 正則言語に対する反復補題(1)

- $L_{01} = \{0^n 1^n \mid n \geq 1\}$ は正則言語でない
 - 正則言語であることは、
正則表現、有限オートマトンを構成すれば証明できる
 - 正則言語でないことの証明は？
反復補題（強力なツール）

6

4.1.1 正則言語に対する反復補題 (2)

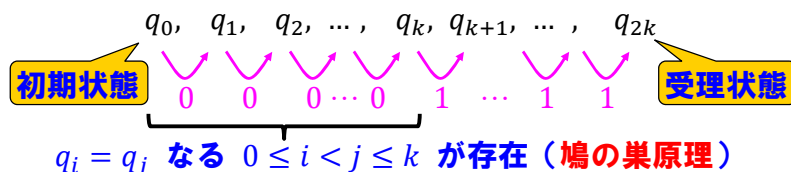
- $L_{01} = \{0^n 1^n \mid n \geq 1\}$ は正則言語でない

■ 正則言語と仮定 (背理法)

$T(M) = L_{01}$ なる有限オートマトン M が存在
状態数を k とする

- $w = 0^k 1^k \in L_{01}$ を考える

- M に w を入力したときの, M の状態変化

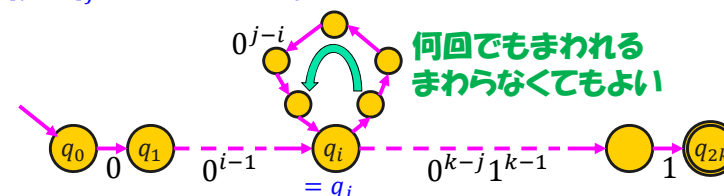
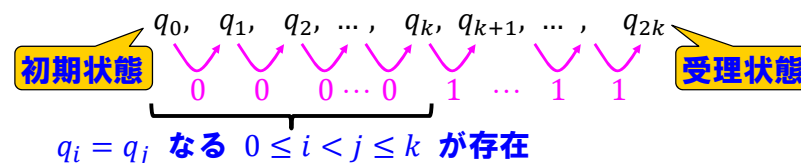


$k+1$ 羽の鳩が k 個の巣に入っている
⇒ 2 羽以上の鳩がいる巣が存在する

7

4.1.1 正則言語に対する反復補題 (3)

- M に w を入力したときの, M の状態変化



$0^{k-(j-i)+n(j-i)} 1^k$ ($n \geq 0$) も受理してしまう

$n \neq 1$ のとき, $0^{k-(j-i)+n(j-i)} 1^k \notin L_{01} (= \{0^n 1^n \mid n \geq 1\})$

$L_{01} = \{0^n 1^n \mid n \geq 1\}$ は正則言語でない

8

反復補題 (1)

- 言語 L が正則言語でないことの証明方法

■ アイデア

十分に長い語 $w \in L$ に対し、

w の一部を任意の回数繰り返して得られる語 w'

$w' \in L$ となるはず

→ 矛盾を導く

■ 定理4.1として定式化

ポンプの補題 (pumping lemma), 繰返し定理とも呼ばれる

9

反復補題(2)

■ 定理4.1 (反復補題)

任意の正則言語 L に対し, ある正整数 n が存在し,

$|w| \geq n$ なる任意の $w \in L$ に対し,

$w = xyz$ ($y \neq \epsilon$, $|xy| \leq n$) と表せ,

任意の k ($k \geq 0$) に対し, $xy^k z \in L$

- 定理4.1は, ある言語が正則言語でないことの証明に有用

10

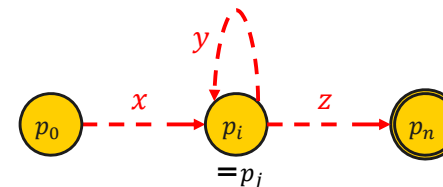
反復補題の証明 (1)

- L : 正則言語
- $A : L(A) = L$ なるオートマトン $A = (Q, \Sigma, \delta, q_0, F)$
- $n = |Q|$ とする.
- $w : |w| = m (\geq n)$ なる任意の語 $w \in L$
 $w = a_1 a_2 \dots a_m \ (a_i \in \Sigma)$
- $\hat{\delta}(q_0, a_1 a_2 \dots a_i) = p_i$ とする. ただし, $p_0 = q_0$
- このとき, $p_i = p_j$ なる $i, j \ (0 \leq i < j \leq m)$ が存在
- $x = a_1 a_2 \dots a_i, y = a_{i+1} a_{i+2} \dots a_j$ とする ($y \neq \epsilon, |xy| \leq n$)
 $w = \underbrace{a_1 a_2 \dots a_i}_{x} \underbrace{a_{i+1} a_{i+2} \dots a_j}_{y} \dots a_m \ (a_i \in \Sigma)$

11

反復補題の証明 (2)

- $p_i = p_j$ なる $i, j \ (0 \leq i < j \leq n)$ が存在
- $x = a_1 a_2 \dots a_i, y = a_{i+1} a_{i+2} \dots a_j$ とする ($0 < |y|, |xy| \leq n$)

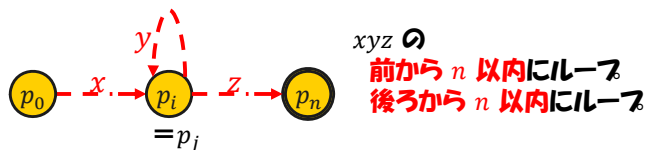


任意の $k (\geq 0)$ について, $xy^k z \in L$

12

反復補題の変形

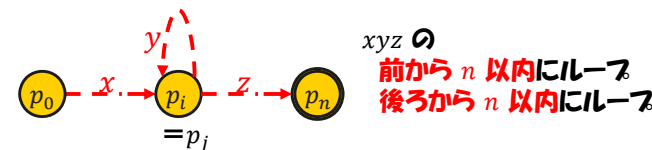
- 反復補題
 任意の正則言語 L に対し, ある正整数 n が存在し,
 $|w| \geq n$ なる任意の $w \in L$ に対し,
 $w = xyz \ (y \neq \epsilon, |xy| \leq n)$ と表せ,
 任意の $k (k \geq 0)$ に対し, $xy^k z \in L$
- 反復補題の変形
 任意の正則言語 L に対し, ある正整数 n が存在し,
 $|w| \geq n$ なる任意の $w \in L$ に対し,
 $w = xyz \ (y \neq \epsilon, |yz| \leq n)$ と表せ,
 任意の $k (k \geq 0)$ に対し, $xy^k z \in L$



13

反復補題の変形

- 反復補題
 任意の正則言語 L に対し, ある正整数 n が存在し,
 $|w| \geq n$ なる任意の $w \in L$ に対し,
 $w = xyz \ (y \neq \epsilon, |xy| \leq n)$ と表せ,
 任意の $k (k \geq 0)$ に対し, $xy^k z \in L$
- 反復補題の変形
 任意の正則言語 L に対し, ある正整数 n が存在し,
 $|w| \geq n$ なる任意の $w \in L$ に対し,
 $w = xyz \ (y \neq \epsilon, |yz| \leq n)$ と表せ,
 任意の $k (k \geq 0)$ に対し, $xy^k z \in L$



16

4.1.2 反復補題の応用 (1)

- $L_{01} = \{0^n 1^n \mid n > 0\}$ は正則言語でない

(証明)

L_{01} を正則言語とする (背理法)

反復補題 (定理4.1) の正整数 n に対し, $w = 0^{n/2} 1^{n/2}$ とする (n は偶数)

$|w| \geq n$ なので, $w = xyz$ ($y \neq \epsilon, |xy| \leq n$) と表せ,

任意の k ($k \geq 0$) に対し, $xy^k z \in L_{01}$

場合分けして, すべての場合で矛盾を導く

- $y = 0^t$ ($t > 0$) のとき
- $y = 1^t$ ($t > 0$) のとき
- $y = 0^s 1^t$ ($s, t > 0$) のとき

17

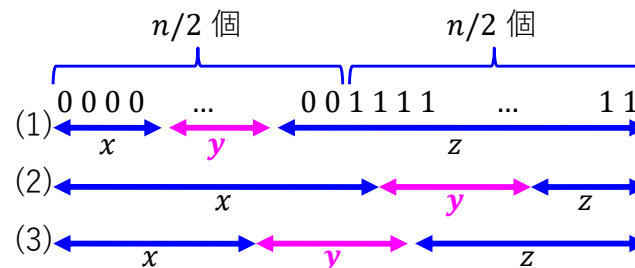
反復補題の応用 (1) 補足

- 場合分けの考え方

$w = 0^{n/2} 1^{n/2}$ に対し,

$w = xyz$ ($y \neq \epsilon, |xy| \leq n$) と表す

- $y = 0^t$ ($t > 0$) のとき
- $y = 1^t$ ($t > 0$) のとき
- $y = 0^s 1^t$ ($s, t > 0$) のとき



18

反復補題の応用 (2)

- $L_{01} = \{0^n 1^n \mid n > 0\}$ は正則言語でない

(証明のつづき)

$w = 0^{n/2} 1^{n/2}$ とする

$w = xyz$ ($y \neq \epsilon, |xy| \leq n$) と表せ, $xy^k z \in L_{01}$ ($k \geq 0$)

- $y = 0^t$ ($t > 0$) のとき

$xy^0 z = 0^{n-t} 1^n \notin L_{01}$ となり矛盾

- $y = 1^t$ ($t > 0$) のとき

$xy^0 z = 0^n 1^{n-t} \notin L_{01}$ となり矛盾

- $y = 0^s 1^t$ ($s, t > 0$) のとき

$xy^2 z = 0^{n-s} (0^s 1^t)^2 1^{n-t} = 0^{n-1} 1^{n+1} \notin L_{01}$ となり矛盾

- (1), (2), (3) より, $L_{01} = \{0^n 1^n \mid n > 0\}$ は正則言語でない

19

4.1.2 反復補題の応用 (3)

- $L_{01} = \{0^n 1^n \mid n > 0\}$ は正則言語でない

(別の証明)

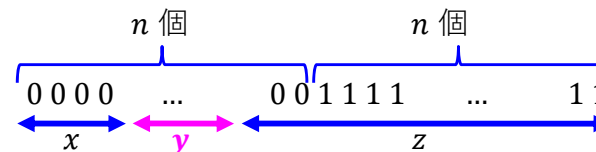
L_{01} を正則言語とする (背理法)

反復補題 (定理4.1) の正整数 n に対し, $w = 0^n 1^n$ とする

$|w| \geq n$ なので, $w = xyz$ ($y \neq \epsilon, |xy| \leq n$) と表せ,

任意の k ($k \geq 0$) に対し, $xy^k z \in L_{01}$

- $y \neq \epsilon, |xy| \leq n$ より, $y = 0^t$ ($t > 0$) この場合しかない!
あとは同様にして矛盾を導く



20

反復補題の応用 (4) 例4. 2

- 例4. 2 $L_{eq} = \{w \in \{0, 1\}^* \mid w \text{ は同じ個数の } 0 \text{ と } 1 \text{ を含む}\}$
は正則言語でない

(証明)

L_{eq} を正則言語とする (背理法)

反復補題の正整数 n に対し, $w \in L_{eq}$, $|w| \geq n$ とする

$w = xyz$ ($y \neq \epsilon$, $|xy| \leq n$) と表せ, $xy^kz \in L_{eq}$ ($k \geq 0$)

y に含まれる 0 と 1 の個数が異なれば,
 $xy^0z \notin L_{eq}$ となり矛盾

21

反復補題の応用 (5) 例4. 2 (つづき)

- 例4. 2 $L_{eq} = \{w \in \{0, 1\}^* \mid w \text{ は同じ個数の } 0 \text{ と } 1 \text{ を含む}\}$
は正則言語でない

(証明)

L_{eq} を正則言語とする (背理法)

反復補題の正整数 n に対し, $w = 0^n 1^n$ とする

*反復補題は $|w| \geq n$ なる任意の w に対して成立

$w = xyz$ ($y \neq \epsilon$, $|xy| \leq n$) と表せることより,

$$y = 0^s \quad (s > 0)$$

$xy^0z = 0^{n-s} 1^n \notin L_{eq}$ となり, 矛盾

22

反復補題の応用 (6) 例4. 3

- 例4. 3 $L_{pr} = \{w \in 1^* \mid w \text{ の長さが素数}\}$
は正則言語でない

(証明)

L_{pr} を正則言語とする (背理法)

反復補題の正整数 n に対し, $w = 1^p$ ($p \geq n + 2$) とする

反復補題の y に対し, $y = 1^m$ ($1 \leq m \leq n$) とする

このとき, $xz = 1^{p-m}$ である

$$xy^{p-m}z = 1^{p-m+m(p-m)} = 1^{(m+1)(p-m)}$$

$m + 1 \geq 2$, $p - m \geq 2$ より, $(m + 1)(p - m)$ は合成数

$xy^{p-m}z = 1^{(m+1)(p-m)} \notin L_{pr}$ となり, 矛盾

23

4. 2 正則言語に関する閉包性

- 正則言語 の閉包性

正則言語が演算 α に関して閉じている:

正則言語に演算 α を施して得られる言語も正則言語

- (1) 二つの正則言語の和集合は正則
- (2) 二つの正則言語の共通集合は正則
- (3) 正則言語の補集合は正則
- (4) 二つの正則言語の差集合は正則
- (5) 正則言語の反転は正則
- (6) 正則言語のスター閉包は正則
- (7) 二つの正則言語の接続は正則
- (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

29

4.2.1 ブール演算のもとでの正則言語の閉包性 (1)

■ 正則言語の和集合に関する閉包性

定理4.4 L と M が正則言語であるとき、 $L \cup M$ もまた正則言語である

(証明)

- L, M が正則言語なので、 $L(R) = L, L(S) = M$ となる正則表現 R, S が存在する
- $R + S$ は正則表現であり、
 $L(R + S) = L(R) \cup L(S) = L \cup M$
- 従って、 $L \cup M$ も正則言語である

正則表現を用いた証明
オートマトンを用いた証明でもよい

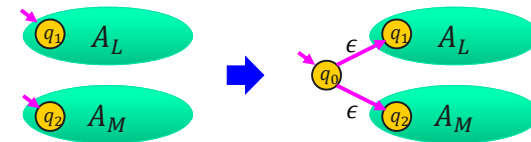
30

ブール演算のもとでの正則言語の閉包性 (2)

定理4.4 L と M が正則言語であるとき、 $L \cup M$ もまた正則言語である

(別の証明)

- L, M が正則言語なので、 $L(A_L) = L, L(A_M) = M$ となる DFA A_L, A_M が存在する
- DFA A_L, A_M から、 $L \cup M$ を受理する ϵ -NFA を構成する



オートマトンを用いた証明
正則表現、オートマトンどちらで証明してもよい

31

ブール演算のもとでの正則言語の閉包性 (3)

■ 正則言語の補集合に関する閉包性

定理4.5 L がアルファベット Σ 上の正則言語であるとき、 $\bar{L} = \Sigma^* - L$ もまた正則言語である

(証明)

- L が正則言語なので、 $L(A) = L$ となる DFA $A = (Q, \Sigma, \delta, q_0, F)$ が存在する
- DFA $B = (Q, \Sigma, \delta, q_0, Q - F)$ を考える
- $L(B) = \Sigma^* - L(A) = \Sigma^* - L = \bar{L}$
- 従って、 $\bar{L} = \Sigma^* - L$ も正則言語である

32

ブール演算のもとでの正則言語の閉包性 (4)

■ 例4.7 正則言語でないことの証明に閉包性を利用

$L_{neq} = \{w \in \{0, 1\}^* \mid w \text{ は異なる個数の } 0 \text{ と } 1 \text{ を含む}\}$
は正則言語でない

(証明)

- 背理法: L_{neq} が正則言語と仮定
- 事実: $L_{eq} = \{w \in \{0, 1\}^* \mid w \text{ は同じ個数の } 0 \text{ と } 1 \text{ を含む}\}$ は正則言語でない
- DFA $L_{eq} = \overline{L_{neq}} = \Sigma^* - L_{neq}$ である
- L_{neq} が正則言語なら、定理4.5より、
 L_{eq} も正則言語 矛盾!
- 従って、 L_{neq} は正則言語でない

33

ブール演算のもとでの正則言語の閉包性 (5)

■ 正則言語の共通部分に関する閉包性

定理4.8 L と M が正則言語であるとき、 $L \cap M$ もまた正則言語である

(証明)

- $L \cap M = \overline{\overline{L} \cup \overline{M}}$ (ド・モルガンの法則)
- 補集合, 和集合に関して閉じているので、
共通部分に関しても閉じている

34

ブール演算のもとでの正則言語の閉包性 (6)

■ 正則言語の共通部分に関する閉包性

定理4.8 L と M が正則言語であるとき、 $L \cap M$ もまた正則言語である

(別の証明)

- L, M が正則言語なので、 $L(A_L) = L, L(A_M) = M$ となる DFA A_L, A_M が存在する
 - $A_L = (Q_L, \Sigma, \delta_L, q_L, F_L), A_M = (Q_M, \Sigma, \delta_M, q_M, F_M)$ とする
- DFA A_L, A_M から、 $L \cap M$ を受理する DFA A を構成する
 - $A = (Q_L \times Q_M, \Sigma, \delta, (q_L, q_M), F_L \times F_M)$
 - $\delta((p, q), a) = (\delta_L(p, a), \delta_M(q, a))$

35

ブール演算のもとでの正則言語の閉包性 (7)

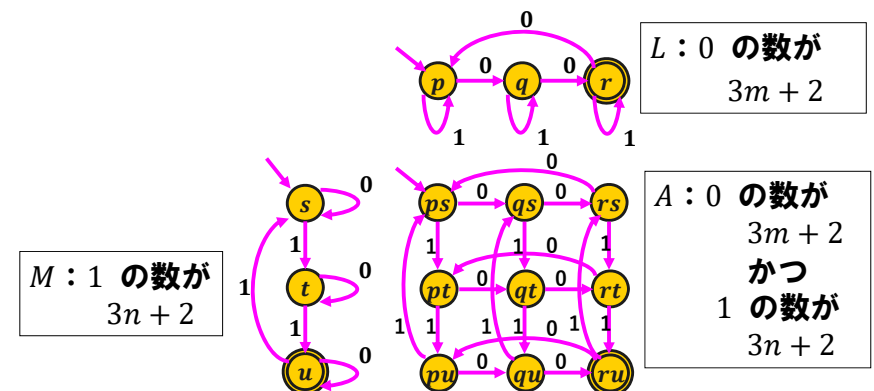
■ DFA A_L, A_M から、 $L \cap M$ を受理する DFA A を構成する

- $A = (Q_L \times Q_M, \Sigma, \delta, (q_L, q_M), F_L \times F_M)$
 - $\delta((p, q), a) = (\delta_L(p, a), \delta_M(q, a))$
- A の構成法のアイデア
 - A の状態: (A_L の状態, A_M の状態) の2項組
 - 第1項で A_L の動作を模倣 (A_L と同じ状態に到達)
 - 第2項で A_M の動作を模倣 (A_M と同じ状態に到達)
 - A_L, A_M を並列に模倣
 - A は $L(A_L) \cap L(A_M)$ を受理する

36

ブール演算のもとでの正則言語の閉包性 (8)

$L \cap M$ を認識する決定性有限オートマトン A の例



37

ブール演算のもとでの正則言語の閉包性 (9)

■ 正則言語の集合差に関する閉包性

定理4.10 L と M が正則言語であるとき, $L - M$ もまた正則言語である

(証明)

- $L - M = L \cap \bar{M}$
- 補集合, 共通部分に関して閉じているので, 集合差に関しても閉じている

38

ブール演算のもとでの正則言語の閉包性 (10)

■ 正則言語の集合差に関する閉包性

定理4.10 L と M が正則言語であるとき, $L - M$ もまた正則言語である

(別の証明)

- L, M が正則言語なので, $L(A_L) = L, L(A_M) = M$ となる DFA A_L, A_M が存在する
 - $A_L = (Q_L, \Sigma, \delta_L, q_L, F_L), A_M = (Q_M, \Sigma, \delta_M, q_M, F_M)$ とする
- DFA A_L, A_M から, $L - M$ を受理する DFA A を構成する
 - $A = (Q_L \times Q_M, \Sigma, \delta, (q_L, q_M), F_L \times (Q_M - F_M))$
 - $\delta((p, q), a) = (\delta_L(p, a), \delta_M(q, a))$

39

4.2.2 反転 (1)

■ 文字列 w の反転 w^R

- $w = a_1 a_2 \cdots a_n$ のとき, $w^R = a_n a_{n-1} \cdots a_1$

■ 言語 L の反転 L^R

- $L^R = \{w^R \mid w \in L\}$
- $L = \{001, 10, 111\}$ のとき, $L^R = \{100, 01, 111\}$

40

4.2.2 反転 (2)

定理4.11 L が正則言語であるとき, L^R もまた正則言語である

(証明)

- L が正則言語なので, $L(A) = L$ となる DFA A が存在する
- DFA A から ϵ -NFA A^R を作る
 - (1) A の状態遷移図において, 有向辺の向きをすべて逆向きにする
 - (2) A の初期状態をただ一つの受理状態とする
 - (3) 新しい初期状態 p_0 を作り, p_0 から A のすべての受理状態へ ϵ -遷移を導入する
- $L(A^R) = L^R$ は帰納法で証明できる

テキストで証明の詳細を
見ておくこと

41

4.2.3 準同型写像 (1)

■ 準同型写像 (ホモモルフィズム)

- $h: \Sigma \rightarrow \Gamma^*$
- 文字列の各文字を特定の文字列で置き換え

■ 例4.3

- $\Sigma = \{0, 1\}$, $\Gamma = \{a, b\}$, $h(0) = ab$, $h(1) = \epsilon$ とする
 - このとき, $h(0011) = abab$
- #### ■ 言語 L の準同型写像 $h(L)$
- $h(L) = \{h(w) \mid w \in L\}$

42

4.2.3 準同型写像 (2)

定理4.14 L がアルファベット Σ 上の正則言語であり, h が Σ 上の準同型写像であるとき, $h(L)$ も正則言語である

(証明)

- L が正則言語なので, $L(R) = L$ となる正則表現 R が存在する
- 正則表現 R から正則表現 $h(R)$ を作る
 - R において, 各記号 $a \in \Sigma$ を $h(a)$ によって置き換える
- $L(h(R)) = h(L)$ は帰納法で証明できる

テキストで証明の詳細を
見ておくこと

43

本日のまとめ

1. 有限オートマトン
 2. 正則表現と正則言語
 3. 正則言語の性質
- ■ 言語が正則でないことの証明
■ 正則言語に関する閉包性
4. 文脈自由文法と言語
 5. プッシュダウン・オートマトン
 6. 文脈自由言語の性質
 7. チューリングマシン

テキスト
4.1~4.2節

44

本日の学習目標

目標を達成できたか
確認してみよう
(復習も含めて)

- 正則言語に対する反復補題とその根拠を説明できる
- 反復補題を用いて, 正則言語でないことの証明ができる
- 与えられた言語が, 正則言語かそうでないか見当がつく
- 正則言語についての3つ以上の閉包性を, 例を用いて説明できる
- 有限オートマトンまたは正則表現を適宜使い分けて, 正則言語の閉包性を証明できる

45