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。

1/32

4. Regular Languages (1):
(Text 4.1,4.2)

4.1. Non-regular language

– Finite automaton has finite states.

→ It cannot distinguish infinite objects.

(Typical) Pigeon Hole Principle:
There are n+1 or more pigeons are in n nests.
Then, there are at least two pigeons in some nest.

2/32

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 $L=\{0^n1^n \mid n≧1\}$

- 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…?

3/32

4. Regular Languages (1):
(Text 4.1,4.2)

4.1. Non-regular language

Ex: Language $L=\{0^n1^n \mid n≧1\}$

- n can be any integer
- When DFA A has m states, what if the transition of A on the input 0^n1^n for $n>m$…?

4/32

例: 言語 $L=\{0^n1^n \mid n≧1\}$ 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。$i≠j$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。

5/32

Ex: The language $L=\{0^n1^n \mid n≧1\}$ is not a regular language.

Proof: To derive contradictions, we assume that L is regular.

Since L is regular, there is a DFA A accepting L. Let $q_1,q_2,\ldots,q_m$ be the set of states (finite m). Suppose $n=m+1$. Then, by the pigeon hole principle, among the inputs

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

there is a pair 0^i, 0^j with $i≠j$ such that A translates to the same state, say q.

Now, consider the input 0^i1^j. Then, since $i≠j$, that is not in L. However, A cannot distinguish $0^i0^j \notin L$ with $0^j1^j \in L$. Therefore, A has to accept both of them, or reject both of them. This contradicts that A accepts the language L.

Hence, L is not a regular language.

6/32

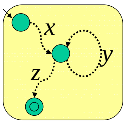
4. 正則言語の性質(1)
(テキスト4.1,4.2)

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、$xy^kz\in L$



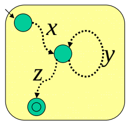
4. Regular Languages (1)
(Text 4.1,4.2)

Basic lemma to show a language is not regular.

4.1. Non-regular language

Pumping Lemma for a regular language:

– For any regular language L, there is a constant n that satisfies the following condition: Any string $w\in L$ with $|w|\geqq n$ can be decomposed to three substrings $w=xyz$.
  1. $y\neq\varepsilon$
  2. $|xy|\leqq n$
  3. For all $k\geqq 0$, $xy^kz\in L$



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. Non-regular Languages

Pumping Lemma:

– For any regular language L, there is a constant n that satisfies the following condition: Any string $w\in L$ with $|w|\geqq n$ can be decomposed to three substrings $w=xyz$.
  1. $y\neq\varepsilon$
  2. $|xy|\leqq n$
  3. For all $k\geqq 0$, $xy^kz\in L$

[Proof] Since L is regular, there is a DFA A with $L(A)=L$. Let n be the number of states of A.

Let $w=a_1a_2\ldots a_m$ be any word in L with $m\geqq n$.

Let p_i be the state of A after reading the substring $a_1a_2\ldots a_i$ (p_0 is the initial state).

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。

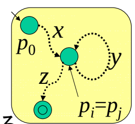
  (1) $y\neq\varepsilon$ (2) $|xy|\leqq n$ (3) $xy^kz\in L$ ($k\geqq 0$)

[証明] 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$p_0=q_0$)
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^kz ($k\geqq 0$)を受理する。



4.1. Non-regular Languages

Pumping Lemma:

– For any regular language L, there is a constant n that satisfies the following condition: Any string $w\in L$ with $|w|\geqq n$ can be decomposed to three substrings $w=xyz$.
  1. $y\neq\varepsilon$
  2. $|xy|\leqq n$
  3. For all $k\geqq 0$, $xy^kz\in L$

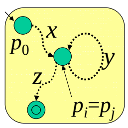
[Proof] A is in state p_i after reading the substring $a_1a_2\ldots a_i$

By pigeon hole principle, there is the same states p_i, p_j with $i<j$ among $p_0,p_1,\ldots,p_m$. Letting

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

It can be $x=\varepsilon$ /$z=\varepsilon$, but we have $y\neq\varepsilon$

A accepts xy^kz for each $k\geqq 0$.



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。

13/32

Ex: Language $L=\{0^n1^n \mid n\geqq 1\}$ is not regular.

Proof by Pumping lemma: To derive contradictions, we suppose that L is regular. Since L is regular, there exists a constant m s.t.

any string w with $|w|\geqq m$ in L can be decomposed three substrings x,y,z with the following conditions:

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

We let $w=0^m1^m$. Then we have three substrings $x,y,$ and z with the above conditions. Since $|xy|\leqq m$, $y\neq\varepsilon$, we have $y=0^i$ $(i\geqq 1)$.

Since $xyz=0^m1^m$, $xyyz=0^{m+i}1^m$. By the pumping lemma, we have $xyyz\in L$. However, $xyyz\notin L$, which is a contradiction.

Hence L is not regular.

14/32

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。

15/32

4. Property of Regular Languages (1): (Text 4.1,4.2)

4.2. Closure property of regular languages

- A set is close under an operation:
  - If all regular languages are still regular if they are changed by an operation, we say
    - regular languages are closed under the operation.

    That is called closure property.

16/32

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

17/32

4.2. Closure property of regular languages

- Regular languages are closed under the following operations:

① For any R.L. L_1 and L_2, $L_1\cup L_2$ is regular

② For any R.L. L_1 and L_2, $L_1\cap L_2$ is regular

③ The complement of a regular language is regular

④ For any R.L. L_1 and L_2, L_1-L_2 is regular

⑤ The reverse of a regular language is regular

⑥ For any R.L. L_1, $L_1{}^*$ is regular

⑦ The concatenation of R.L.s L_1 and L_2 is regular

⑧ A homomorphism of a regular language is regular

⑨ The inverse of homomorphism of a regular language is regular

4 methods for proof

Out of range

18/32

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. Closure property of R.L.

① For any R.L. L_1 and L_2, $L_1 \cup L_2$ is regular.

[Proof method 1] Using regular expressions

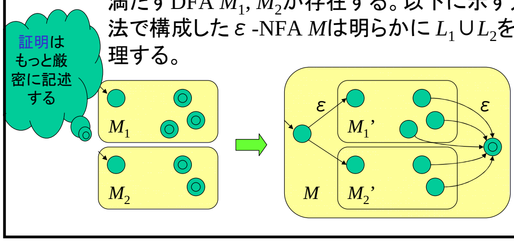
Since L_1 and L_2 are regular, there are two regular expressions E_1 and E_2 with $L(E_1)=L_1$, $L(E_2)=L_2$. Then $((E_1)+(E_2))$ is also regular expression, and clearly, $L(((E_1)+(E_2)))=L_1 \cup L_2$.

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。

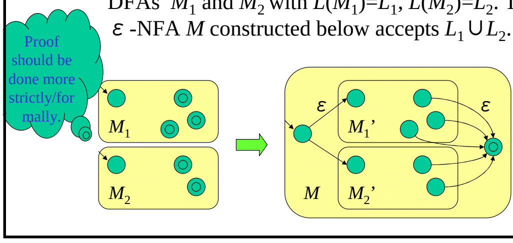


4.2. Closure property of R.L.

① For any R.L. L_1 and L_2, $L_1 \cup L_2$ is regular.

[Proof method 2] Using automata

Since L_1 and L_2 are regular languages, there are two DFAs M_1 and M_2 with $L(M_1)=L_1$, $L(M_2)=L_2$. The ε-NFA M constructed below accepts $L_1 \cup L_2$.



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 $\overline{L}=\{ w \mid w \notin L\}$

[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. Closure property of R.L.

③ The complement of a regular language is regular

[Definition] The complement of a language L:

$$\overline{L} = \{w \mid w \notin L\}$$

[Proof] (Method 2)

Since L is regular, there is a DFA $A=(Q, \Sigma, \delta, q, F)$ with $L(A)=L$. Then, the DFA $A'=(Q, \Sigma, \delta, q, Q-F)$, which is obtained by swapping F and $Q-F$, accepts the complement of L.

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

[証明手法3]

ド･モルガンの定理より、

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

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

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

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. Closure property of R.L.

② For any R.L. L_1 and L_2, $L_1 \cap L_2$ is regular

[Proof method 3]

By "De Morgan's Law",

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

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

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

Hence , if L_1 and L_2 are regular, by ①,③, so is $L_1 \cap L_2$.

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)]

Mの状態=(M_1の状態,M_2の状態)

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. Closure property of R.L.

④ For any R.L. L_1 and L_2, $L_1 - L_2$ is regular

(Since $L_1 - L_2 = L_1 \cap \overline{L_2}$, method 3 also works.)

[Proof method 4 (product construction)]

① Let M_1 and M_2 be the DFAs that accept L_1, L_2.

② DFA M, which accepts $L_1 - L_2$, reads the input and simulates simultaneously
- the transfer of M_1 for the input
- the transfer of M_2 for the input

state of M = (state of M_1,state of M_2)

③ When input is end, if M_1 accepts and M_2 does not accept, M accepts.

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 $L^R=\{ w \mid w^R \in L\}$

[証明]

A: DFA

A^R: ε -NFA

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. Closure property of R.L.

⑤ The reverse of a regular language is regular

[Definition]

The reverse of a string $w=x_1x_2\ldots x_k$: $w^R=x_k\ldots x_2x_1$.

The reverse of a language L: $L^R=\{ w \mid w^R \in L\}$

[Proof]

A: DFA

A^R: ε -NFA

For the DFA A accepting L,

① make the accepting state of A unique,

② reverse all transfers of A,

③ exchange the (unique) accepting state and initial state

ε -NFA A^R accepts L^R.

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.

4.2. Closure property of R.L.

For regular languages L_1 and L_2,

⑥ L_1* is regular.

⑦ The concatenation of L_1 and L_2 is regular.

For the regular expressions E_1 and E_2 for L_1 and L_2,

⑥ $(E_1)^*$

⑦ $(E_1)(E_2)$

guarantee the claims.