# アルゴリズム論 1
# 第 1 回
# オリエンテーション, 正規表現

関川 浩

2017/04/12

1. オリエンテーション
   - 講義内容
   - 教科書・参考書・資料等
   - 成績評価方法
2. 正規表現
   - イントロダクション
   - 形式言語の導入
   - 言語の例
   - 正規表現の定義と性質

# 講義の概要

コンピュータによる計算の原理を数学的なモデルを用いて説明

## 前半

言語理論とオートマトンについて紹介

目標: 正規表現, 文脈自由文法, 有限オートマトン,
プッシュダウンオートマトンについて理解できること

## 後半

帰納的関数とチューリング機械による計算可能性を中心とした計算理論を説明後, 計算量理論について簡単に解説

目標: 計算可能性と計算量の基本が理解できること

より詳しくはシラバスを参照 (折にふれて見てほしい)

# 前半: 言語理論とオートマトン

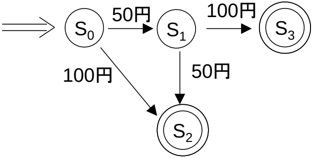
言語のモデルを計算機のモデル (オートマトン) で処理する過程を通して, 計算機や計算機で計算する原理を説明

- 第 1 回 (4 月 12 日): オリエンテーション, 正規表現
- 第 2 回 (4 月 19 日): 有限オートマトン
- 第 3 回 (4 月 26 日): 正規表現と有限オートマトンの関係
- 第 4 回 (5 月 10 日): 文脈自由文法 (1)
- 第 5 回 (5 月 17 日): 文脈自由文法 (2)
- 第 6 回 (5 月 24 日): プッシュダウンオートマトン (1)
- 第 7 回 (5 月 31 日): プッシュダウンオートマトン (2)

# オートマトンの例

自動販売機

- 100 円の品物を販売
- 受け付けるのは 50 円, 100 円硬貨のみ



# 後半: 計算可能性と計算量理論

- 計算機では解けない問題がある
- 計算機で解ける場合, その手間はどれくらいか?

第 8 回 (6 月 7 日): 帰納的関数
第 9 回 (6 月 14 日): チューリング機械と計算可能な関数 (1)
第 10 回 (6 月 21 日): チューリング機械と計算可能な関数 (2)
第 11 回 (6 月 28 日): 帰納的に可算な集合と決定問題
第 12 回 (7 月 5 日): 計算量理論の基本概念
第 13 回 (7 月 12 日): NP 完全性
第 14 回 (7 月 19 日): NP 完全問題の例

# 教科書・参考書・資料等

- 教科書は指定しない
- 参考書
  (1) 岩間一雄: オートマトン・言語と計算理論, コロナ社, 2003 年
  (2) 有川節夫, 宮野悟: オートマトンと計算可能性, 培風館, 1986 年

  前半は主に (1), 後半は主に (2)
- 資料 (講義で使用するスライド)
  講義三日前の日曜までには LETUS に upload する予定

# 成績評価方法

- 試験 (70%)
  第 1–14 回の全範囲から基本的な問題を中心に出題
- レポート (30%): 前半終了時 (第 7 回終了時)
  基本的な事項が理解できているかどうかを評価

# 第 1 回から第 3 回の内容

言語理論とオートマトンの主題
無限集合である言語をいかに表現するか

一つの方法:
言語が満たすルールをうまく書いて, それを言語の表現とする手法.
文法がその代表

第 1 回: 正規表現というシステムを紹介

第 2 回: 有限オートマトンという表現法を紹介

第 3 回: 正規表現と有限オートマトン (見掛けはかなり違う) は
言語の表現能力が等しいことを証明

有限オートマトン (正規表現) の能力の限界を説明

# 文章の意味を理解する

「文章の意味を理解する」とは?

**例: 数式の計算**

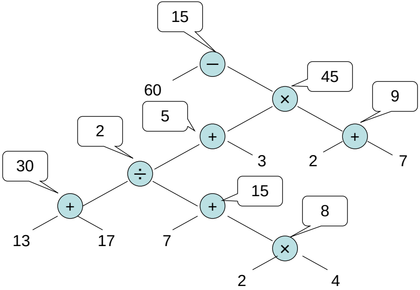
たとえば, 以下の数式を計算するには, その意味の理解が必要

$$60 - \left(\frac{13+17}{7+2\times 4} + 3\right) \times (2+7)$$

数式の意味とは? どういう表現が可能?

- 二つの数の四則演算しか知らない人に複雑な計算を教えることを考える
- 何らかの分かりやすい記述でうまく教えることができたとき, その記述を意味ととらえてみる

$$60 - \left(\frac{13+17}{7+2\times 4} + 3\right) \times (2+7)$$




頂点, 枝, 根, 葉, 子, 子孫, 親, 先祖, . . .

各頂点の値を決めるのに必要な知識は二つの数の演算のみ
$\Longrightarrow$ 式の意味として上の図の木を対応させる

導出木 (どうしゅつき):
　文法 (対応の規則) にしたがって作成された木

# 文章の正しさを解析する

- 文章の意味が定まるためには, その文章が正しいことが条件
- 正しい文章とは?

### 正しい例 (前スライドの例)

$$60 - \left( \frac{13 + 17}{7 + 2 \times 4} + 3 \right) \times (2 + 7)$$

### 正しくない例

$$60 - \left( \frac{13 + 17}{7 + 2 \times 4} + 3 \right) \times (2+)$$

# 例: カッコ文 (1/2)

**手続き (※)**

数式からカッコのみを残して数値や演算記号を除く

例 $((12+34)\times(56-7))\times(89+10) \longmapsto (()())()$

正しいカッコの並び $\overset{\text{def}}{\Longleftrightarrow}$ 「手続き (※)」で得られるカッコの並び

- 正しい例

  $(()())()$

- 正しくない例

  $((()())()$ $\Longrightarrow$ 左カッコが多い
  $(()()))(()$ $\Longrightarrow$ 途中で右カッコが多過ぎ

カッコの並びをカッコ文と呼ぶことにする

# 例: カッコ文 (2/2)

## 問題

カッコ文が正しいか否かを判定せよ

## 解答

- カウンタを用意．初期値は 0
- カッコの並びを左から見ていき，
  - 左カッコがあればカウンタを $+1$
  - 右カッコがあればカウンタを $-1$
- カッコ文が正しい
  $\Longleftrightarrow$ カウンタがつねに 0 以上，かつ，最終値が 0

与えられた文章が正しいか否かを判定するための計算機のモデルとしてオートマトンを導入する (次回)

# 形式言語の導入 (1/3)

- アルファベット
  記号の有限集合 ($\neq \emptyset$)

  例　$\{0, 1\}$, $\{a, b, c\}$, $\ldots$

- アルファベット $\Sigma$ 上の記号列 (あるいは単に列)
  $\Sigma$ に属する記号を, 重複を許して有限個, 一列に並べたもの

  例　$\Sigma = \{0, 1\}$ のときの 01001

- 列の長さ
  列に含まれる記号の数. 列 $x$ の長さを $|x|$ と書く

  例　列 $x = 01001$ のとき $|x| = 5$

- 空列
  長さ 0 の列. アルファベットによらず $\varepsilon$ で表す
  (長さ 0 の列は普通には表現できない)

# 形式言語の導入 (2/3)

- 列 $x$ の逆
  $x$ を逆順に読んだ列. $x^{\mathrm{R}}$ と書く

  例　$x = 0010$ のとき, $x^{\mathrm{R}} = 0100$

- 列 $x$ と $y$ の連接
  $x$ の後ろに $y$ をつなげた列. $xy$ と書く. $xx$ を $x^2$ とも書く

  例　$x = 101$, $y = 00$ なら $xy = 10100$
  任意の列 $x$ に対し, $x\varepsilon = \varepsilon x = x$

- 列 $y$ が列 $x$ の部分列
  ある列 $z_1$, $z_2$ が存在して $x = z_1 y z_2$ と書けること

- $\Sigma^*$
  アルファベット $\Sigma$ 上のすべての列の集合

  例　$\Sigma = \{0, 1\}$ のとき, $\Sigma^* = \{\varepsilon, 0, 1, 00, 01, 10, 11, 000, \ldots\}$

# 形式言語の導入 (3/3)

- アルファベット $\Sigma$ 上の言語
  $\Sigma$ 上の列の集合のこと

  例　$\Sigma = \{0, 1\}$ のとき, 以下の集合は言語

  $$\{0, 01, 10, 11\} \text{ (有限集合)}, \qquad \{0^n 1^n \mid n \geq 0\} \text{ (無限集合)}$$

- 言語 $L_1$ と $L_2$ の連接
  $\{x_1 x_2 \mid x_1 \in L_1$ かつ $x_2 \in L_2\}$ のこと
  $L_1 L_2$ (または $L_1 \cdot L_2$) と書く
  $L$ を言語としたとき,

  $$L^0 \stackrel{\text{def}}{=} \{\varepsilon\}, \quad L^1 \stackrel{\text{def}}{=} L, \quad L^2 \stackrel{\text{def}}{=} LL, \quad \ldots$$

- 言語 $L^*$ ($L$ は言語)
  $L^0 \cup L^1 \cup L^2 \cup L^3 \cup \cdots$ のこと

# 言語の例 1 (1/3)

## 問題

$M = \{a, bc, bcc, bccc\}$ のとき, $M^*$ はどんな言語か

## 解答

$x \in M^* \iff x = \varepsilon$, あるいは, $x$ は以下の 4 条件を満たす

1. $x$ の最初は $a$ または $b$
2. $x$ に $a$ が現れたら次は (もしあれば) $a$ または $b$
3. $x$ に $b$ が現れたら次は $c$
4. $c$ は 4 個以上は続かない

# 言語の例 1 (2/3)

## 証明 (1/2)

$(\Longrightarrow)$

任意の $x \in M^*$ を取ると, ある $i$ が存在して $x \in M^i$

以下, $i$ についての帰納法

- $i = 0$ なら $x = \varepsilon$ で成立
- $i = n - 1$ で成立するとして $i = n$ のとき
  $x = yz$, $y \in M^{n-1}$, $z \in M = \{a, bc, bcc, bccc\}$ と書ける
  - $y = \varepsilon$ のとき ($n = 1$ のとき) は明らか
  - $y \neq \varepsilon$ のとき ($1 < n$ のとき)
    $y$ が条件 1 から 4 を満たすことに注意すれば, $x$ が条件を満たすことが確認できる

## 言語の例 1 (3/3)

### 証明 (2/2)

($\Longleftarrow$)

$x \neq \varepsilon$ のとき, $x$ の以下の場所に区切を入れる

- 列の最初と最後
- $a$ と $a$ の間
- $a$ と $b$ の間
- $c$ と $a$ の間
- $c$ と $b$ の間

例　$x = abcabccc$ なら $|a|bc|a|bccc|$

$x$ が 4 条件を満たしていれば, 区切記号と区切記号の間は

$$a, \quad bc, \quad bcc, \quad bccc$$

のいずれか ■

# 言語の例 2

## 問題

長さが奇数の, $\{0,1\}$ 上の列全体からなる言語を表現せよ

## 解答 1

長さが偶数の列全体は $\{00,01,10,11\}^*$
よって, 長さが奇数の列全体は, $\{00,01,10,11\}^*\{0,1\}$

## 解答 2

$\{0,1\}$ 上の列全体から偶数長の列を除く

$$\{0,1\}^* \setminus \{00,01,10,11\}^*$$

## 注意

集合 $A$, $B$ に対し, $A \setminus B \stackrel{\mathrm{def}}{=} \{x \mid x \in A$ かつ $x \notin B\}$

# 正規表現の定義

アルファベット $\Sigma$ 上の正規表現 $S$ と, $S$ が表現する言語 $L(S) \subseteq \Sigma^*$ は, 以下のように定義される

(1) $\emptyset$ は正規表現で, $L(\emptyset) = \emptyset$

(2) $a \in \Sigma$ なら $a$ は正規表現で, $L(a) = \{a\}$

(3) $R_1$ と $R_2$ が正規表現なら,

$$(R_1 + R_2), \qquad (R_1 \cdot R_2) \ ((R_1 R_2) \text{ とも書く}), \qquad (R_1^*)$$

も正規表現で,

$$L((R_1 + R_2)) = L(R_1) \cup L(R_2),$$
$$L((R_1 \cdot R_2)) = L(R_1)L(R_2),$$
$$L((R_1^*)) = (L(R_1))^*$$

(4) 1–3 で得られるもののみが正規表現

# 正規表現の定義に関する注意

## 注意

- 定義の (1) 「$\emptyset$ は正規表現で, $L(\emptyset) = \emptyset$」において,
  - 最初と二番目の $\emptyset$ は正規表現に利用する特別な記号
  - 最後の $\emptyset$ は空集合

  を表す
- 演算の優先度は以下の順で高いことにする

$$* \qquad \cdot \qquad +$$

  たとえば, $(0(1^*)) + 0$ は $01^* + 0$ と書いてよい

# 例題

## 問題

次の正規表現はどのような言語を表現しているか

1. $R_1 = \emptyset^*$
2. $R_2 = ((0+\varepsilon)^* + (001+11)^*)\emptyset$
3. $R_3 = (0+1)^*000(0+1)^*$

## 解答

1. $L(R_1) = L(\emptyset^*) = (L(\emptyset))^* = \emptyset^* = \emptyset^0 \cup \emptyset^1 \cup \emptyset^2 \cup \cdots$
   $= \{\varepsilon\} \cup \emptyset \cup \emptyset \cup \cdots = \{\varepsilon\}$
2. $R'_2 = (0+\varepsilon)^* + (001+11)^*$ とすると,
   $$L(R_2) = L(R'_2\emptyset) = L(R'_2)L(\emptyset) = L(R'_2)\emptyset = \emptyset$$
3. $L((0+1)^*) = (L(0+1))^* = (L(0) \cup L(1))^* = (\{0\} \cup \{1\})^* = \{0,1\}^*$
   $\Longrightarrow L(R_3)$ は, 0 がどこかに 3 個以上連続している $\{0,1\}$ 上の列の全体

# 基本的な性質

## 定理 1

言語 $L_1$ と $L_2$ が正規表現で表現できるなら, $L_1 \cup L_2$, $L_1 L_2$, $L_1^*$ も正規表現で表現できる

## 証明

$L_1 \cup L_2$ のみ示す (ほかも正規表現の定義の (3) を使えば同様).
正規表現 $R_1$, $R_2$ が存在して, $L_1 = L(R_1)$, $L_2 = L(R_2)$ と書ける.
正規表現の定義の (3) より, $R_1 + R_2$ も正規表現で,

$$L(R_1 + R_2) = L(R_1) \cup L(R_2) = L_1 \cup L_2 \qquad \blacksquare$$

## 注意

$L_1 \cap L_2$, $\Sigma^* \setminus L_1$ については第 3 回で