形式言語とオートマトン

2020年度講義資料 (1)

新潟大学工学部

青戸等人

目次

目次

科目の概要

- 計算論では，モデル化と抽象化によって，計算のさまざまな側面に理論的な焦点を当てる.

科目の概要

- 計算論では，モデル化と抽象化によって，計算のさまざまな側面に理論的な焦点を当てる.
- この分野の最も基礎的でかつ応用範囲も広いトピックである形式言語・オートマトンについて，特に，正規言語と文脈自由言語について講義する.

科目の概要

- 計算論では，モデル化と抽象化によって，計算のさまざまな側面に理論的な焦点を当てる.
- この分野の最も基礎的でかつ応用範囲も広いトピックである形式言語・オートマトンについて，特に，正規言語と文脈自由言語について講義する.
- 具体的には，有限オートマトン，正規言語，正規表現，プッシュダウン・オートマトン，文脈自由言語，ポンピング補題などについて講義する.

科目のねらい

- 形式言語・オートマトンの理論は，どのような計算手段で記号列の(無限)集合が扱えるか，計算手段によって何が取り扱え，また，取り扱えないか，そして，そのような記号列の集合がどのように特徴づけられるかを対象とする学問である.

科目のねらい

- 形式言語・オートマトンの理論は，どのような計算手段で記号列の(無限)集合が扱えるか，計算手段によって何が取り扱え，また，取り扱えないか，そして，そのような記号列の集合がどのように特徴づけられるかを対象とする学問である.
- この理論は，計算論やコンパイラの基礎として重要なばかりでなく，情報分野における理論的なアプローチとその応用可能性を学ぶ上での格好のトピックである.

科目のねらい

- 形式言語・オートマトンの理論は，どのような計算手段で記号列の(無限)集合が扱えるか，計算手段によって何が取り扱え，また，取り扱えないか，そして，そのような記号列の集合がどのように特徴づけられるかを対象とする学問である.
- この理論は，計算論やコンパイラの基礎として重要なばかりでなく，情報分野における理論的なアプローチとその応用可能性を学ぶ上での格好のトピックである.
- このような観点から，形式言語・オートマトンの理論の基礎を習得することを目標とする.

学習の到達目標

1. 有限オートマトンとその計算について理解していること.
2. 正規言語とその閉包性について理解していること.
3. 決定性有限オートマトンと非決定性有限オートマトン，その関係について理解していること.
4. 正規表現および有限オートマトンとの関係について理解していること.
5. ポンピング補題を理解し，言語の非正規性を示せること.
6. 文脈自由文法について理解すること.
7. プッシュダウン・オートマトンおよび文脈自由言語との関係について理解していること.

授業計画

1回 イントロダクション
2回 有限オートマトン
3回 正規言語
4回 正規演算と正規演算の閉包性(1)
5回 非決定性有限オートマトン
6回 正規演算の閉包性(2)
7回 有限オートマトンの最小化
8回 正規表現
9回 正規表現と有限オートマトンの等価性
10回 非正規言語と正規言語に対するポンピング補題
11回 正規言語に関する決定可能問題
12回 文脈自由文法とチョムスキー標準形
13回 プッシュダウン・オートマトン
14回 文脈自由文法とプッシュダウン・オートマトンの等価性
15回 非文脈自由言語と非文脈自由言語に対するポンピング補題

教科書について

- 教科書
  「計算理論の基礎」[1]　オートマトンと言語
  M.Sipser著
  太田和夫，田中圭介監訳
  共立出版

教科書について

- 教科書
  「計算理論の基礎」[1]　オートマトンと言語
  M.Sipser著
  太田和夫，田中圭介監訳
  共立出版
- 原著は「Introduction to the Theory of Computation」
- 和訳は3分冊[1]〜[3]になっている.
  講義で用いるのは分冊[1].
  また，配布する講義資料にて，分冊[2]の一部の内容にも触れる予定.

教科書について

- 教科書
  「計算理論の基礎」[1]　オートマトンと言語
  M.Sipser著
  太田和夫，田中圭介監訳
  共立出版
- 原著は「Introduction to the Theory of Computation」
- 和訳は3分冊[1]〜[3]になっている.
  講義で用いるのは分冊[1].
  また，配布する講義資料にて，分冊[2]の一部の内容にも触れる予定.
- “Theory of Computation”—日本語では「計算の理論」. 情報分野の基礎理論の1つ.

講義の情報

- 講義ホームページ

  http://www.nue.ie.niigata-u.ac.jp/~aoto/lecture/Automata/

- 講義の補足資料を，いくつか用意します．講義ホームページに置きます．このスライドも置きます．
- 期末試験実施予定です．教科書，ノート等参照可です．
  オンライン実施に伴い，成績評価の割合を変更しました．
  ($\Rightarrow$ レポート50%，期末試験50%)
- 講義時限：火曜と金曜の1限(全15回)． 今年はすべてオンラインで実施します．動画をアップする予定です．受講する時間は柔軟に変更して構いません．
- オフィスアワー
  Slack 上で受け付けの予定です(が，大学のセキュリティガイドラインで利用できないようでしたら，別の方法を考えます)．詳細は学務情報システムから連絡します．

講義の復習とレポート

- 復習：毎回の講義はそれ以前の講義内容の理解を前提とします．講義について行くには，毎回，講義内容を余さず消化することが重要です．不明な点はすぐに解決するとともに，演習問題を解いて理解を確実にしておくこと．
- 講義の復習として，講義で進んだところまでの演習問題を，次回の講義前までに解いておくとよい．

講義の復習とレポート

- 復習： 毎回の講義はそれ以前の講義内容の理解を前提とします．講義について行くには，毎回，講義内容を余さず消化することが重要です．不明な点はすぐに解決するとともに，演習問題を解いて理解を確実にしておくこと．
- 講義の復習として，講義で進んだところまでの演習問題を，次回の講義前までに解いておくとよい．
- 教科書の演習問題のうち，その回の講義までで，きちんと解けていなければいけない問題を選んで，レポート課題にします．

講義の復習とレポート

- 復習： 毎回の講義はそれ以前の講義内容の理解を前提とします．講義について行くには，毎回，講義内容を余さず消化することが重要です．不明な点はすぐに解決するとともに，演習問題を解いて理解を確実にしておくこと．
- 講義の復習として，講義で進んだところまでの演習問題を，次回の講義前までに解いておくとよい．
- 教科書の演習問題のうち，その回の講義までで，きちんと解けていなければいけない問題を選んで，レポート課題にします．
- レポート課題で，解答が教科書に載っていないものについては，解答(例)を公開します．解答のチェックは，自分でやってください．講師の方では採点しません．

講義の復習とレポート

- 復習： 毎回の講義はそれ以前の講義内容の理解を前提とします．講義について行くには，毎回，講義内容を余さず消化することが重要です．不明な点はすぐに解決するとともに，演習問題を解いて理解を確実にしておくこと．
- 講義の復習として，講義で進んだところまでの演習問題を，次回の講義前までに解いておくとよい．
- 教科書の演習問題のうち，その回の講義までで，きちんと解けていなければいけない問題を選んで，レポート課題にします．
- レポート課題で，解答が教科書に載っていないものについては，解答(例)を公開します．解答のチェックは，自分でやってください．講師の方では採点しません．
- レポート課題は，学務情報システムに登録します． ノート等に解答して，解答例等で答え合わせ(丸付け)した後，その写真をとって，締切までに学務情報システムから写真を提出してください．

目次

教科書目次から(1)

- 分冊[1]の内容：
  - 0. 序論…(数学的な) 準備
  - 1. 正規言語
  - 2. 文脈自由言語

教科書目次から(1)

- 分冊[1]の内容：
  - 0. 序論…(数学的な）準備
  - 1. 正規言語
  - 2. 文脈自由言語
- この講義のメインの内容は，「正規言語」と「文脈自由言語」の2種類.

教科書目次から(1)

- 分冊[1]の内容：
  - 0. 序論…(数学的な）準備
  - 1. 正規言語
  - 2. 文脈自由言語
- この講義のメインの内容は，「正規言語」と「文脈自由言語」の2種類.
- ここでいう「言語」とは
  - 言語とは，文字列の集合のこと(専門用語).
  - 有限種類の記号，例：$\{a, b, c\}$
  - （無限かもしれない）文字列の集合，例：$\{a, aa, aaba, \ldots\}$
  - 無限集合でも，有限メモリの計算機，(有限の)計算時間で扱いたい←困難さ

教科書目次から(2)

- 言語の階層：

$$L_1 = \{\mathtt{a}, \mathtt{ba}, \mathtt{bba}, \ldots\}$$
$$L_2 = \{\mathtt{ab}, \mathtt{aabb}, \mathtt{aaabbb}, \ldots\}$$
$$L_3 = \quad \text{“a}\text{と}\text{b}\text{を使った文字列全部”}$$

教科書目次から(2)

- 言語の階層：

$L_1 = \{\mathtt{a}, \mathtt{ba}, \mathtt{bba}, \ldots\}$
$L_2 = \{\mathtt{ab}, \mathtt{aabb}, \mathtt{aaabbb}, \ldots\}$
$L_3 =$ “aとbを使った文字列全部”

  - L_1より，L_2の方が，複雑にみえる．(?)
  - L_2より，L_3の方が，複雑にみえる．(?)
  - L_2より，C言語のプログラムの集合の方が，複雑にみえる(?)

教科書目次から(2)

- 言語の階層：

  $L_1 = \{\mathtt{a}, \mathtt{ba}, \mathtt{bba}, \ldots\}$
  $L_2 = \{\mathtt{ab}, \mathtt{aabb}, \mathtt{aaabbb}, \ldots\}$
  $L_3 =$ “aとbを使った文字列全部”

  - L_1より，L_2の方が，複雑にみえる．(?)
  - L_2より，L_3の方が，複雑にみえる．(?)
  - L_2より，C言語のプログラムの集合の方が，複雑にみえる(?)
  - 言語の分類，言語の特徴，コンピュータによって，どのようなクラスの言語ならどの程度に扱えるのか?
  - 「正規言語」 $\subset$ 「文脈自由言語」

教科書目次から(3)

- 正規言語
  - (限定的なので)“良い”性質を持っている
    - 正規言語は計算機上で“表現”できる.
    - 異なる表現をしても同じ言語かプログラムで判定できる.
  - どの正規言語も“有限オートマトン”というマシンで特徴づけられる
  - どの正規言語も“正規表現”という表現で特徴づけられる
  - さまざまな応用，例えば，
    - 正規表現による文字列照合
    - 字句解析(コンパイラの機能の一部)

教科書目次から(4)

- 文脈自由言語
  - 正規言語より高い表現力 (よりいろいろな言語を含む)
  - ほとんどのプログラミング言語は（ほぼ）文脈自由言語
  - プログラムの構文解析
    - 入力文字列からプログラムとしての構造を抽出
    - 構文解析器（パーザー）：コンパイラの機能の一部
    - パーザージュネレータ (yacc)：文法記述からパーザープログラムを自動生成
    - 広く用いられるLR構文解析法は，正規言語と密接な関係
  - どの文脈自由言語も“プッシュダウンオートマトン”というマシンで特徴づけられる
  - どの文脈自由言語も“文脈自由文法”という表現で特徴づけられる
  - “良い”性質は，正規言語より少ない.

教科書目次から(5)

- [2],[3]分冊の話題
- [2]：計算可能性の理論
  - 「正規言語」 $\subset$ 「文脈自由言語」$\subset \ldots$
  - コンピュータで計算できるとは
  - 判定可能性，帰納的集合，帰納的列挙可能集合

教科書目次から(5)

- [2],[3]分冊の話題
- [2]：計算可能性の理論
  - 「正規言語」 $\subset$ 「文脈自由言語」$\subset \ldots$
  - コンピュータで計算できるとは
  - 判定可能性，帰納的集合，帰納的列挙可能集合
- [3]：複雑さの理論
  - プログラムの効率の良さとは.
  - (時間)計算量，NP完全性
  - 近似アルゴリズム，確率的アルゴリズム，暗号

目次

集合

- $\in$と$\subseteq$の区別
  $1 \in \{1,2,3\}$と $\{1,2\} \subseteq \{1,2,3\}$
  - 1は集合$\{1,2,3\}$の元(または，要素ともいう)
  - 集合$\{1,2\}$は集合$\{1,2,3\}$の部分集合
- 集合の等しさは要素が同じかで決まる：
  $\{1,2\} = \{1,1,2\} = \{2,1\}$

集合

- $\in$と$\subseteq$の区別
  $1 \in \{1,2,3\}$と $\{1,2\} \subseteq \{1,2,3\}$
  - 1は集合$\{1,2,3\}$の元(または，要素ともいう)
  - 集合$\{1,2\}$は集合$\{1,2,3\}$の部分集合
- 集合の等しさは要素が同じかで決まる：
  $\{1,2\} = \{1,1,2\} = \{2,1\}$
- 表記
  - $\mathcal{N}$ (0を含まない)自然数全体の集合 $\{1,2,\ldots\}$
  - $\mathcal{Z}$ 整数全体の集合 $\{\ldots,-2,-1,0,1,2,\ldots\}$
  - $\mathcal{Z}_m = \{0,1,2,\ldots,m-1\}$
  - $\{\mathsf{TRUE},\mathsf{FALSE}\}$　ブール値
  - $\emptyset$ 空集合 $\{\}$とも書く
  - $A \cap B$ 積集合
  - $A \cup B$ 和集合
  - $\overline{A}$ 補集合

列と組

- 列：要素を順番に並べたもの
  - $7, 21, 57$
  - $(7, 21, 57)$
  - $1, 1, 1, 1, \ldots$　　無限列
  - $(1, 2) \neq (2, 1)$

列と組

- 列：要素を順番に並べたもの
  - $7, 21, 57$
  - $(7, 21, 57)$
  - $1, 1, 1, 1, \ldots$　　無限列
  - $(1, 2) \neq (2, 1)$
- 組：有限の列
  - $(7, 21, 57)$ 3個組
  - $(1, 2)$ 2個組(普通，対という)

列と組

- 列：要素を順番に並べたもの
  - $7, 21, 57$
  - $(7, 21, 57)$
  - $1, 1, 1, 1, \ldots$　　無限列
  - $(1, 2) \neq (2, 1)$
- 組：有限の列
  - $(7, 21, 57)$ 3個組
  - $(1, 2)$ 2個組(普通，対という)
- 直積(Cartesian product)：組全部を集めた集合
  - $\{1, 2\} \times \{x, y, z\} = \{(1, x), (1, y), (1, z), (2, x), (2, y), (2, z)\}$
  - $\{1, 2\} \times \{x, y, z\} \times \{1, 2\} = \{(1, x, 1), (1, x, 2), \ldots\}$
  - $A \times A \times A = A^3$
  - $\mathcal{N} \times \mathcal{N} = \{(i, j) \mid 1 \leq i, j\}$

関数と関係

- 関数
  - $+ : \mathcal{N} \times \mathcal{N} \to \mathcal{N}$
  - $\mathrm{abs} : \mathcal{Z} \to \mathcal{Z}$
  - $f : A \to B$のとき，Aを定義域，Bを値域とよぶ.
  - 単射関数，全射関数，全単射関数

関数と関係

- 関数
  - $+ : \mathcal{N} \times \mathcal{N} \to \mathcal{N}$
  - $\mathrm{abs} : \mathcal{Z} \to \mathcal{Z}$
  - $f : A \to B$のとき，Aを定義域，Bを値域とよぶ.
  - 単射関数，全射関数，全単射関数
- 対応関係を表で書いて関数を記述することがある(教科書例0.8,例0.9)
- 関数は入出力の組を集合で書くことでも表わせる.

$$\mathrm{abs} = \{(0,0),(1,1),(-1,1),(2,2),(-2,2),\ldots\}$$

このとき，$\mathrm{abs} \subseteq \mathcal{Z} \times \mathcal{Z}$

関数と関係

- 関係
  - $A \subseteq \mathbb{Z} \times \mathbb{Z}$なる集合$A$を，$\mathbb{Z}$上の2項関係とよぶ.
    つまり，$\mathbb{Z}$上の2項関係は，整数の対を要素にもつ集合
  - $< = \{(0,1),(0,2),\ldots,(1,2),(1,3),\ldots,\}$
    $<$は，$\mathbb{Z}$上の2項関係

関数と関係

- 関係
  - $A \subseteq \mathcal{Z} \times \mathcal{Z}$なる集合$A$を，$\mathcal{Z}$上の2項関係とよぶ.
    つまり，$\mathcal{Z}$上の2項関係は，整数の対を要素にもつ集合
  - $< = \{(0,1),(0,2),\ldots,(1,2),(1,3),\ldots,\}$
    $<$は，$\mathcal{Z}$上の2項関係
  - 関係は，述語とも見なせる.

$$0 < 1 = \mathsf{TRUE}$$
$$0 < 2 = \mathsf{TRUE}$$
$$0 < 0 = \mathsf{FALSE}$$
$$1 < 0 = \mathsf{FALSE}$$
$$\ldots\ldots\ldots\ldots$$

$$< : \mathcal{Z} \times \mathcal{Z} \to \{\mathsf{TRUE}, \mathsf{FALSE}\}$$

- 述語　$f : A \to \{\mathsf{TRUE}, \mathsf{FALSE}\}$なる関数

関数と関係

- 関係
  - $A \subseteq \mathcal{Z} \times \mathcal{Z}$なる集合$A$を，$\mathcal{Z}$上の2項関係とよぶ.
    つまり，$\mathcal{Z}$上の2項関係は，整数の対を要素にもつ集合
  - $< = \{(0,1),(0,2),\ldots,(1,2),(1,3),\ldots,\}$
    $<$は，$\mathcal{Z}$上の2項関係
  - 関係は，述語とも見なせる.

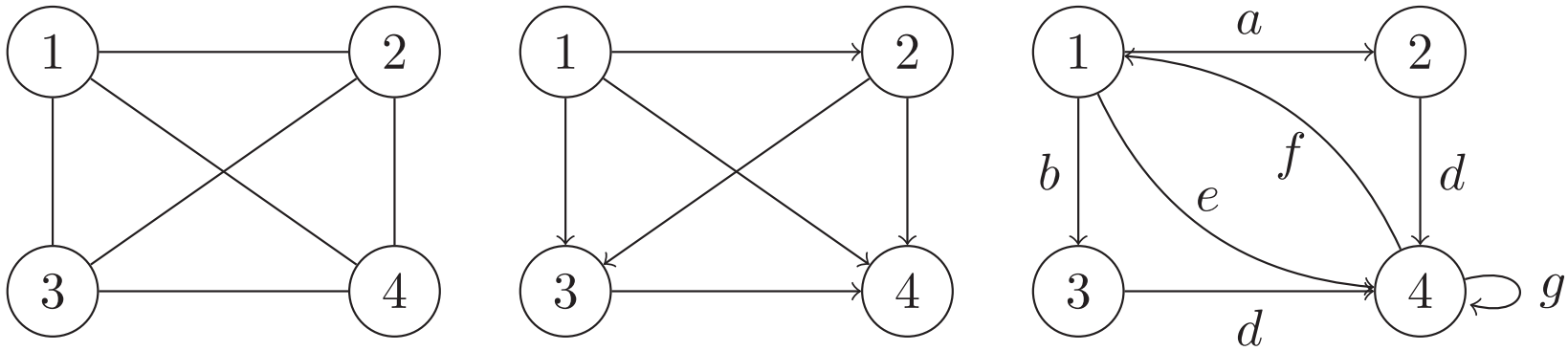
$$0 < 1 = \mathsf{TRUE}$$
$$0 < 2 = \mathsf{TRUE}$$
$$0 < 0 = \mathsf{FALSE}$$
$$1 < 0 = \mathsf{FALSE}$$
$$\ldots\ldots\ldots\ldots$$

$$< : \mathcal{Z} \times \mathcal{Z} \to \{\mathsf{TRUE}, \mathsf{FALSE}\}$$

- 述語　$f : A \to \{\mathsf{TRUE}, \mathsf{FALSE}\}$なる関数
- 反射律，対称律，推移律，同値関係

グラフ

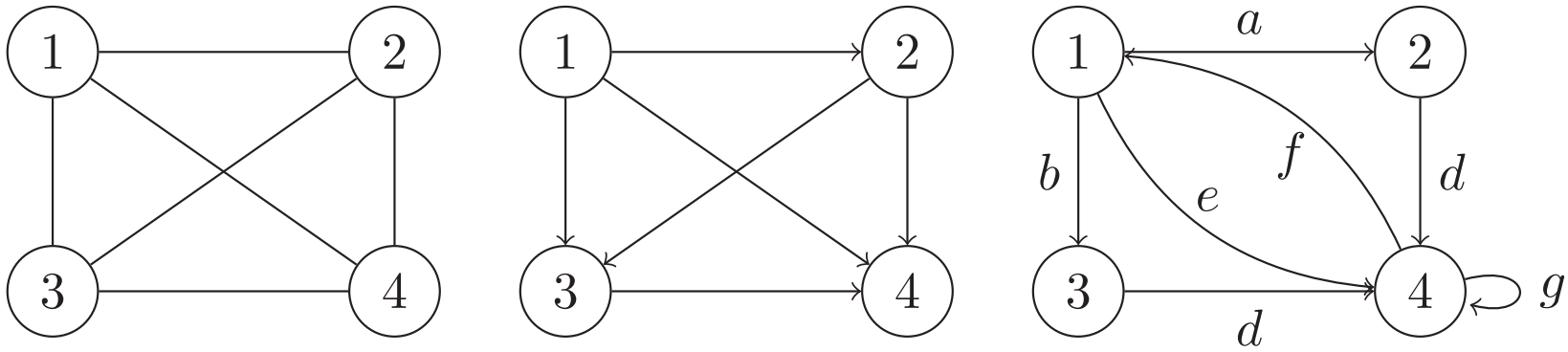
- 無向グラフ，有向グラフ，ラベル付き有向グラフ



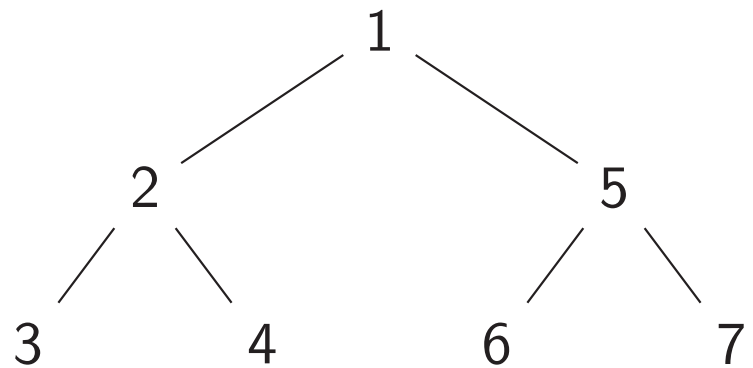
- 頂点(節点)，辺，パス(経路)
- (有向)グラフの記述
  $G = (\{1, 2, 3, 4\}, \{(1, 2), (1, 3), (1, 4), (2, 3), (2, 4), (3, 4)\})$

グラフ

- 無向グラフ，有向グラフ，ラベル付き有向グラフ



- 頂点(節点)，辺，パス(経路)
- (有向)グラフの記述
  $G = (\{1, 2, 3, 4\}, \{(1, 2), (1, 3), (1, 4), (2, 3), (2, 4), (3, 4)\})$
- 木(tree)，根(root)，葉(leaf)



文字列と言語

- アルファベット　使える記号を集めた集合
- 文字(記号)　アルファベットの要素
- アルファベットの例
  - $\Sigma_1 = \{0, 1\}$
  - $\Sigma_2 = \{\mathtt{a}, \mathtt{b}, \ldots, \mathtt{z}\}$
- 文字列
  - Σ_1上の文字列の例：$\mathtt{01}, \mathtt{000}, \mathtt{10101}, \ldots$
  - Σ_2上の文字列の例：$\mathtt{abc}, \mathtt{alphabet}, \mathtt{abracadabra}, \ldots$

文字列と言語

- アルファベット　使える記号を集めた集合
- 文字(記号)　アルファベットの要素
- アルファベットの例
  - $\Sigma_1 = \{0, 1\}$
  - $\Sigma_2 = \{\mathtt{a}, \mathtt{b}, \ldots, \mathtt{z}\}$
- 文字列
  - Σ_1上の文字列の例：$\mathtt{01}, \mathtt{000}, \mathtt{10101}, \ldots$
  - Σ_2上の文字列の例：$\mathtt{abc}, \mathtt{alphabet}, \mathtt{abracadabra}, \ldots$
- 空文字列　(見えるように) εと書く.

文字列と言語

- アルファベット　使える記号を集めた集合
- 文字(記号)　アルファベットの要素
- アルファベットの例
  - $\Sigma_1 = \{0, 1\}$
  - $\Sigma_2 = \{\mathtt{a}, \mathtt{b}, \ldots, \mathtt{z}\}$
- 文字列
  - Σ_1上の文字列の例：$\mathtt{01}, \mathtt{000}, \mathtt{10101}, \ldots$
  - Σ_2上の文字列の例：$\mathtt{abc}, \mathtt{alphabet}, \mathtt{abracadabra}, \ldots$
- 空文字列　(見えるように) εと書く.
- $|w|$　文字列wの長さ，例：$|\mathtt{abc}| = 3$，　$|\varepsilon| = 0$

文字列と言語

- アルファベット　使える記号を集めた集合
- 文字(記号)　アルファベットの要素
- アルファベットの例
  - $\Sigma_1 = \{0, 1\}$
  - $\Sigma_2 = \{\mathtt{a}, \mathtt{b}, \ldots, \mathtt{z}\}$
- 文字列
  - Σ_1上の文字列の例：$0, 000, 10101, \ldots$
  - Σ_2上の文字列の例：$\mathtt{abc}, \mathtt{alphabet}, \mathtt{abracadabra}, \ldots$
- 空文字列　(見えるように) εと書く.
- $|w|$　文字列wの長さ，例：$|\mathtt{abc}| = 3$，　$|\varepsilon| = 0$
- 文字列の連結 $w_1 = 001$, $w_2 = 100$なら，$w_1 w_2 = 001100$.
  $w\varepsilon = w = \varepsilon w$がいつでも成立.
- 同じ文字列の連結 $w = 01$なら，$w^3 = 010101$

文字列と言語

- アルファベット　使える記号を集めた集合
- 文字(記号)　アルファベットの要素
- アルファベットの例
  - $\Sigma_1 = \{0, 1\}$
  - $\Sigma_2 = \{\mathtt{a}, \mathtt{b}, \ldots, \mathtt{z}\}$
- 文字列
  - Σ_1上の文字列の例：$0, 000, 10101, \ldots$
  - Σ_2上の文字列の例：$\mathtt{abc}, \mathtt{alphabet}, \mathtt{abracadabra}, \ldots$
- 空文字列　(見えるように) εと書く.
- $|w|$　文字列wの長さ，例：$|\mathtt{abc}| = 3$，　$|\varepsilon| = 0$
- 文字列の連結 $w_1 = 001$, $w_2 = 100$なら，$w_1 w_2 = 001100$.
  $w\varepsilon = w = \varepsilon w$がいつでも成立.
- 同じ文字列の連結 $w = 01$なら，$w^3 = 010101$
- 部分文字列 11は，001100の部分文字列

文字列と言語

- アルファベット　使える記号を集めた集合
- 文字(記号)　アルファベットの要素
- アルファベットの例
  - $\Sigma_1 = \{0, 1\}$
  - $\Sigma_2 = \{\mathtt{a}, \mathtt{b}, \ldots, \mathtt{z}\}$
- 文字列
  - Σ_1上の文字列の例：$0, 000, 10101, \ldots$
  - Σ_2上の文字列の例：$\mathtt{abc}, \mathtt{alphabet}, \mathtt{abracadabra}, \ldots$
- 空文字列　(見えるように) εと書く.
- $|w|$　文字列wの長さ，例：$|\mathtt{abc}| = 3$，　$|\varepsilon| = 0$
- 文字列の連結 $w_1 = 001$, $w_2 = 100$なら，$w_1 w_2 = 001100$.
  $w\varepsilon = w = \varepsilon w$がいつでも成立.
- 同じ文字列の連結 $w = 01$なら，$w^3 = 010101$
- 部分文字列 11は，001100の部分文字列
- 文字列の反転 $11000^{\mathcal{R}} = 00011$

文字列と言語

- アルファベット　使える記号を集めた集合
- 文字(記号)　アルファベットの要素
- アルファベットの例
  - $\Sigma_1 = \{0, 1\}$
  - $\Sigma_2 = \{\mathtt{a}, \mathtt{b}, \ldots, \mathtt{z}\}$
- 文字列
  - Σ_1上の文字列の例：$0, 000, 10101, \ldots$
  - Σ_2上の文字列の例：$\mathtt{abc}, \mathtt{alphabet}, \mathtt{abracadabra}, \ldots$
- 空文字列　(見えるように) εと書く.
- $|w|$　文字列wの長さ，例：$|\mathtt{abc}| = 3$，　$|\varepsilon| = 0$
- 文字列の連結 $w_1 = 001$, $w_2 = 100$なら，$w_1 w_2 = 001100$.
  $w\varepsilon = w = \varepsilon w$がいつでも成立.
- 同じ文字列の連結 $w = 01$なら，$w^3 = 010101$
- 部分文字列 11は，001100の部分文字列
- 文字列の反転 $11000^{\mathcal{R}} = 00011$
- 言語　文字列の集合のこと

BOOLE論理

- 命題論理
- 真理値(ブール値) $\{\mathrm{TRUE}, \mathrm{FALSE}\}$
  ($\{1, 0\}$を使ったりもする)
- 否定($\neg$)，論理積($\wedge$)，論理和($\vee$)，排他的論理和($\oplus$)，含意($\rightarrow$)，等価($\leftrightarrow$)

A	$\neg A$
1	0
0	1

A	B	$A \wedge B$
1	1	1
1	0	0
0	1	0
0	0	0

A	B	$A \vee B$
1	1	1
1	0	1
0	1	1
0	0	0

A	B	$A \oplus B$
1	1	0
1	0	1
0	1	1
0	0	0

A	B	$A \rightarrow B$
1	1	1
1	0	0
0	1	1
0	0	1

A	B	$A \leftrightarrow B$
1	1	1
1	0	0
0	1	0
0	0	1

目次

定義，定理，証明

- 定義　専門用語が何を意味するかの約束(当然ながら，約束が変われば，話の辻褄(推論のつながり)が全部変わってしまうので，定義はとても重要．)

定義，定理，証明

- 定義　専門用語が何を意味するかの約束(当然ながら，約束が変われば，話の辻褄(推論のつながり)が全部変わってしまうので，定義はとても重要．)
- 数学的命題　論理式で記述できる言明

定義，定理，証明

- 定義　専門用語が何を意味するかの約束(当然ながら，約束が変われば，話の辻褄(推論のつながり)が全部変わってしまうので，定義はとても重要．)
- 数学的命題　論理式で記述できる言明
- 定理　特に重要であることを強調したい数学的命題を，定理とよぶ.
- 補題　定理の証明のために補助的に示す数学的命題を補題とよぶ

定義，定理，証明

- 定義　専門用語が何を意味するかの約束(当然ながら，約束が変われば，話の辻褄(推論のつながり)が全部変わってしまうので，定義はとても重要．)
- 数学的命題　論理式で記述できる言明
- 定理　特に重要であることを強調したい数学的命題を，定理とよぶ.
- 補題　定理の証明のために補助的に示す数学的命題を補題とよぶ
- 証明　数学的命題が正しいことを示す一連の論理的な推論
- 証明方法のいろいろ
  構成的証明，背理法，帰納法など，よく使われるので名前がついている.

定義，定理，証明

- 定義　専門用語が何を意味するかの約束(当然ながら，約束が変われば，話の辻褄(推論のつながり)が全部変わってしまうので，定義はとても重要．)
- 数学的命題　論理式で記述できる言明
- 定理　特に重要であることを強調したい数学的命題を，定理とよぶ．
- 補題　定理の証明のために補助的に示す数学的命題を補題とよぶ
- 証明　数学的命題が正しいことを示す一連の論理的な推論
- 証明方法のいろいろ
  構成的証明，背理法，帰納法など，よく使われるので名前がついている．
  $\Longrightarrow$ 証明の原理などの深い話は，2年後期の「数理論理学」のトピック

定義，定理，証明

- 定義　専門用語が何を意味するかの約束(当然ながら，約束が変われば，話の辻褄(推論のつながり)が全部変わってしまうので，定義はとても重要．)
- 数学的命題　論理式で記述できる言明
- 定理　特に重要であることを強調したい数学的命題を，定理とよぶ．
- 補題　定理の証明のために補助的に示す数学的命題を補題とよぶ
- 証明　数学的命題が正しいことを示す一連の論理的な推論
- 証明方法のいろいろ
  構成的証明，背理法，帰納法など，よく使われるので名前がついている．
  $\Longrightarrow$ 証明の原理などの深い話は，2年後期の「数理論理学」のトピック

この講義では，あまり証明の細部には立ち入らず，
アイデアの説明中心でいく予定．

目次

最後に：今回の復習，勉強の仕方

- 0章の演習問題は，特に解かなくてよいです．
- 本スライドの p.17以降の内容で，不明な点があれば，青戸に質問するか，離散数学の教科書を復習してください．

最後に：今回の復習，勉強の仕方

- 0章の演習問題は，特に解かなくてよいです．
- 本スライドの p.17以降の内容で，不明な点があれば，青戸に質問するか，離散数学の教科書を復習してください．
- この講義は難しい予備知識は必要になりませんが，基本的なことや初歩的なことを幅広く知っていないとつまづく可能性があるかと思います．

最後に：今回の復習，勉強の仕方

- 0章の演習問題は，特に解かなくてよいです．
- 本スライドの p.17以降の内容で，不明な点があれば，青戸に質問するか，離散数学の教科書を復習してください．
- この講義は難しい予備知識は必要になりませんが，基本的なことや初歩的なことを幅広く知っていないとつまづく可能性があるかと思います．
- 忘れていたり，知らなかったりしたときに，すぐ聞いて解決しておくのが，つまづかないコツかと思います．