

形式言語とオートマトン

2018年度講義資料 (1)

新潟大学工学部

青戸等人

目次

講義概要

オートマトン，計算可能性，複雑さ (教科書0.1節)

数学的概念や用語 (教科書0.2節)

定義，定理，証明 (教科書0.3節)

今回の復習について

目次

講義概要

オートマトン，計算可能性，複雑さ (教科書0.1節)

数学的概念や用語 (教科書0.2節)

定義，定理，証明 (教科書0.3節)

今回の復習について

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트로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/](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。
- ▶ 講義時限：火曜と金曜の2限(全15回)。
- ▶ オフィスアワー

週1回、青戸の居室で質問等を受け付ける時間を作ります。研究室のゼミ等のスケジュールと勘案して時間を設定しますので、まだ未定です。ホームページ上で連絡します。(大概の回では、講義の直後の時間も、質問を受け付けできるかと思います。)

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いよい。

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、次の講義前までに解いておくことよい。
- ▶ **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、レポート問題にします。**
- ▶ レポート問題で、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。**
- ▶ (たぶん)学務情報システムを使って、レポートの提出をしてもらいます。丸付け済みの答案の写真をアップロードしてもらおうつもりです。

目次

講義概要

オートマトン，計算可能性，複雑さ (教科書0.1節)

数学的概念や用語 (教科書0.2節)

定義，定理，証明 (教科書0.3節)

今回の復習について

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: $\{a, aa, aaba, \dots\}$
 - ▶ 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：

$$L_1 = \{a, ba, bba, \dots\}$$

$$L_2 = \{ab, aabb, aaabbb, \dots\}$$

$$L_3 = \text{“aとbを使った文字列全部”}$$

教科書目次から(2)

▶ 言語の階層：

$$L_1 = \{a, ba, bba, \dots\}$$

$$L_2 = \{ab, aabb, aaabbb, \dots\}$$

$$L_3 = \text{“aとbを使った文字列全部”}$$

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

教科書目次から(2)

▶ 言語の階層：

$$L_1 = \{a, ba, bba, \dots\}$$

$$L_2 = \{ab, aabb, aaabbb, \dots\}$$

$$L_3 = \text{“aとbを使った文字列全部”}$$

- ▶ L_1 より, L_2 の方が, 複雑に見える. (?)
- ▶ L_2 より, L_3 の方が, 複雑に見える. (?)
- ▶ L_2 より, C言語のプログラムの集合の方が, 複雑に見える(?)
- ▶ 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
 - ▶ パーザージェネレータ (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」 $\subset \dots$
 - ▶ 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」 $\subset \dots$
 - ▶ 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

講義概要

オートマトン，計算可能性，複雑さ (教科書0.1節)

数学的概念や用語 (教科書0.2節)

定義，定理，証明 (教科書0.3節)

今回の復習について

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 $\{1, 2, \dots\}$
 - ▶ $\mathcal{Z}$ 整数全体の集合 $\{\dots, -2, -1, 0, 1, 2, \dots\}$
 - ▶ $\mathcal{Z}_m = \{0, 1, 2, \dots, m-1\}$
 - ▶ $\{\text{TRUE}, \text{FALSE}\}$ 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, ... 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, ... 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, ... 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
 - ▶ $\{1, 2\} \times \{x, y, z\} = \{(1, x), (1, y), (1, z), (2, x), (2, y), (2, z)\}$
 - ▶ $\{1, 2\} \times \{x, y, z\} \times \{1, 2\} = \{(1, x, 1), (1, x, 2), \dots\}$
 - ▶ $A \times A \times A = A^3$
 - ▶ $\mathcal{N} \times \mathcal{N} = \{(i, j) \mid 1 \leq i, j\}$

関数と関係

▶ 関数

- ▶ $+: \mathcal{N} \times \mathcal{N} \rightarrow \mathcal{N}$
- ▶ $\text{abs} : \mathcal{Z} \rightarrow \mathcal{Z}$
- ▶ $f : A \rightarrow B$ 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
 - ▶ $+: \mathcal{N} \times \mathcal{N} \rightarrow \mathcal{N}$
 - ▶ $\text{abs} : \mathcal{Z} \rightarrow \mathcal{Z}$
 - ▶ $f : A \rightarrow B$ 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.

$$\text{abs} = \{(0, 0), (1, 1), (-1, 1), (2, 2), (-2, 2), \dots\}$$

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

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
- ▶ $< = \{(0, 1), (0, 2), \dots, (1, 2), (1, 3), \dots, \}$
 $<$ は, $\mathbb{Z}$ 上の2項関係

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
- ▶ $< = \{(0, 1), (0, 2), \dots, (1, 2), (1, 3), \dots\}$
 $<$ は、 $\mathbb{Z}$ 上の2項関係
- ▶ 関係は、述語とも見なせる。

$$0 < 1 = \text{TRUE}$$

$$0 < 2 = \text{TRUE}$$

$$0 < 0 = \text{FALSE}$$

$$1 < 0 = \text{FALSE}$$

.....

$$< : \mathbb{Z} \times \mathbb{Z} \rightarrow \{\text{TRUE}, \text{FALSE}\}$$

- ▶ 述語 $f : A \rightarrow \{\text{TRUE}, \text{FALSE}\}$ なる関数

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
- ▶ $< = \{(0, 1), (0, 2), \dots, (1, 2), (1, 3), \dots, \}$
 $<$ は、 $\mathbb{Z}$ 上の2項関係
- ▶ 関係は、述語とも見なせる。

$$0 < 1 = \text{TRUE}$$

$$0 < 2 = \text{TRUE}$$

$$0 < 0 = \text{FALSE}$$

$$1 < 0 = \text{FALSE}$$

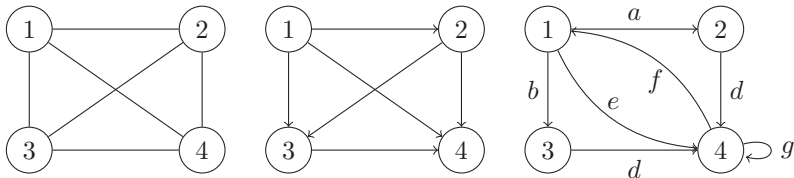
.....

$$< : \mathbb{Z} \times \mathbb{Z} \rightarrow \{\text{TRUE}, \text{FALSE}\}$$

- ▶ 述語 $f : A \rightarrow \{\text{TRUE}, \text{FALSE}\}$ 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

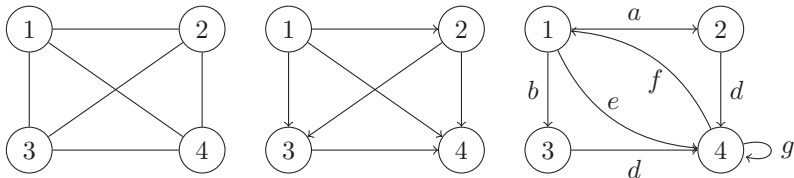


- ▶ 頂点(節点), 辺, パス(経路)
- ▶ (有向)グラフの記述

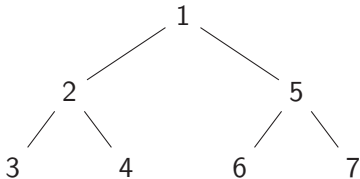
$$G = (\{1, 2, 3, 4\}, \{(1, 2), (1, 3), (1, 4), (2, 3), (2, 4), (3, 4)\})$$

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
 - ▶ $\Sigma_1 = \{0, 1\}$
 - ▶ $\Sigma_2 = \{a, b, \dots, z\}$
- ▶ 文字列
 - ▶ Σ_1 上の文字列の例 : 01, 000, 10101, ...
 - ▶ Σ_2 上の文字列の例 : abc, alphabet, abracadabra, ...

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
 - ▶ $\Sigma_1 = \{0, 1\}$
 - ▶ $\Sigma_2 = \{a, b, \dots, z\}$
- ▶ 文字列
 - ▶ Σ_1 上の文字列の例: 01, 000, 10101, ...
 - ▶ Σ_2 上の文字列の例: abc, alphabet, abracadabra, ...
- ▶ 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
 - ▶ $\Sigma_1 = \{0, 1\}$
 - ▶ $\Sigma_2 = \{a, b, \dots, z\}$
- ▶ 文字列
 - ▶ Σ_1 上の文字列の例: 01, 000, 10101, ...
 - ▶ Σ_2 上の文字列の例: abc, alphabet, abracadabra, ...
- ▶ 空文字列 (見えるように) ε と書く.
- ▶ $|w|$ 文字列 w の長さ, 例: $|abc| = 3, |\varepsilon| = 0$

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
 - ▶ $\Sigma_1 = \{0, 1\}$
 - ▶ $\Sigma_2 = \{a, b, \dots, z\}$
- ▶ 文字列
 - ▶ Σ_1 上の文字列の例: 01, 000, 10101, ...
 - ▶ Σ_2 上の文字列の例: abc, alphabet, abracadabra, ...
- ▶ 空文字列 (見えるように) ε と書く.
- ▶ $|w|$ 文字列 w の長さ, 例: $|abc| = 3$, $|\varepsilon| = 0$
- ▶ 文字列の連結 $w_1 = 001$, $w_2 = 100$ なら, $w_1w_2 = 001100$
- ▶ 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
 - ▶ $\Sigma_1 = \{0, 1\}$
 - ▶ $\Sigma_2 = \{a, b, \dots, z\}$
- ▶ 文字列
 - ▶ Σ_1 上の文字列の例: 01, 000, 10101, ...
 - ▶ Σ_2 上の文字列の例: abc, alphabet, abracadabra, ...
- ▶ 空文字列 (見えるように) ε と書く.
- ▶ $|w|$ 文字列 w の長さ, 例: $|abc| = 3$, $|\varepsilon| = 0$
- ▶ 文字列の連結 $w_1 = 001$, $w_2 = 100$ なら, $w_1 w_2 = 001100$
- ▶ 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
 - ▶ $\Sigma_1 = \{0, 1\}$
 - ▶ $\Sigma_2 = \{a, b, \dots, z\}$
- ▶ 文字列
 - ▶ Σ_1 上の文字列の例: 01, 000, 10101, ...
 - ▶ Σ_2 上の文字列の例: abc, alphabet, abracadabra, ...
- ▶ 空文字列 (見えるように) ε と書く.
- ▶ $|w|$ 文字列 w の長さ, 例: $|abc| = 3$, $|\varepsilon| = 0$
- ▶ 文字列の連結 $w_1 = 001$, $w_2 = 100$ なら, $w_1w_2 = 001100$
- ▶ 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
 - ▶ $\Sigma_1 = \{0, 1\}$
 - ▶ $\Sigma_2 = \{a, b, \dots, z\}$
- ▶ 文字列
 - ▶ Σ_1 上の文字列の例: 01, 000, 10101, ...
 - ▶ Σ_2 上の文字列の例: abc, alphabet, abracadabra, ...
- ▶ 空文字列 (見えるように) ε と書く.
- ▶ $|w|$ 文字列 w の長さ, 例: $|abc| = 3$, $|\varepsilon| = 0$
- ▶ 文字列の連結 $w_1 = 001$, $w_2 = 100$ なら, $w_1w_2 = 001100$
- ▶ 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 $11000^R = 00011$
- ▶ 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) $\{\text{TRUE}, \text{FALSE}\}$
($\{1, 0\}$ 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

講義概要

オートマトン，計算可能性，複雑さ (教科書0.1節)

数学的概念や用語 (教科書0.2節)

定義，定理，証明 (教科書0.3節)

今回の復習について

定義, 定理, 証明

- ▶ 定義
- ▶ 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.
⇒ 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

講義概要

オートマトン，計算可能性，複雑さ (教科書0.1節)

数学的概念や用語 (教科書0.2節)

定義，定理，証明 (教科書0.3節)

今回の復習について

今回の復習について

- ▶ 0章の演習問題は、特に解かなくてよいです.
- ▶ スライドの p.17以降の内容で、不明な点があれば、離散数学の教科書を復習するか、この後か次回の講義の後などに、質問してください.