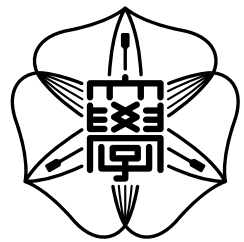
計算理論

Thomas Zeugmann

北海道大学 大学院情報科学研究科
アルゴリズム研究室

http://www-alg.ist.hokudai.ac.jp/~thomas/ToC/

第 1 回：形式言語の導入
(日本語訳：大久保 好章，湊 真一)



動機 I

本講義は，魅惑的かつ重要なテーマ：

計算理論 (*the theory of computation*).

を学ぶためのものである．

動機 I

本講義は，魅惑的かつ重要なテーマ：

計算理論 (*the theory of computation*).

を学ぶためのものである．
それは，計算機ハードウェア，ソフトウェア，そして，アプリケーションの基礎となる数学的性質を扱う．これから，何が計算できて，何が計算できないかを判断していこう．もし，それが計算可能であるなら，どのような計算モデルで，どのくらい高速に，どれだけの記憶容量 (メモリ) で可能なのかを理解することも試みる．

動機 II

計算理論は，工学技術の実践との多くの関わりを持ち，また，真の科学として，哲学的な側面をも含む.

動機 II

計算理論は，工学技術の実践との多くの関わりを持ち，また，真の科学として，哲学的な側面をも含む.

形式言語 (formal language) は，計算機科学において基礎となる重要なものであることから，まずはそれらについて詳しく見ていこう.

動機 II

計算理論は，工学技術の実践との多くの関わりを持ち，また，真の科学として，哲学的な側面をも含む．

形式言語 (formal language) は，計算機科学において基礎となる重要なものであることから，まずはそれらについて詳しく見ていこう．

最初は，形式言語理論 (*formal language theory*) について学ぼう．

動機 III

一般的には，形式言語理論は，言語 (*language*) と呼ばれる文字列の集合と，それらを生成・識別する機構に関わるものである．文字列集合を生成する機構は，通常，文法 (*grammar*) と呼ばれ，また，文字列集合を識別する機構は，受理器 (*acceptors*) あるいはオートマトン (*automata*) と呼ばれる．

動機 III

一般的には，形式言語理論は，言語 (*language*) と呼ばれる文字列の集合と，それらを生成・識別する機構に関わるものである．文字列集合を生成する機構は，通常，文法 (*grammar*) と呼ばれ，また，文字列集合を識別する機構は，受理器 (*acceptors*) あるいはオートマトン (*automata*) と呼ばれる．

言語の生成・受理に関する数学理論は 1950 年代後半に提案され，それ以来，広範囲に渡って研究されている．今日では，計算機言語と自然言語の両者に関する緻密な理論が存在する．

動機 IV

ここでは，形式言語理論の最も基礎的な部分，すなわち，正規言語，文脈自由言語，そして，帰納可算的言語に絞って議論する．形式言語理論は何についてのものであるのか，および，基礎的な証明技法にはどのようなものがあるのかといった基礎理解を得るにはこれで十分であろう．

動機 IV

ここでは，形式言語理論の最も基礎的な部分，すなわち，正規言語，文脈自由言語，そして，帰納可算的言語に絞って議論する．形式言語理論は何についてのものであるのか，および，基礎的な証明技法にはどのようなものがあるのかといった基礎理解を得るにはこれで十分であろう．

共通の基礎を得るために，必要となる数学的な予備知識を復習する．

予備知識 I

任意の集合 M について，M の 要素数 (*cardinality*) および べき集合 (*power set*) をそれぞれ $\mathrm{card}(M)$，$\wp(M)$ と表す.

予備知識 I

任意の集合 M について，M の 要素数 (*cardinality*) および べき集合 (*power set*) をそれぞれ $\mathrm{card}(M)$，$\wp(M)$ と表す.

任意のふたつの集合 X, Y について，X と Y の和集合・積集合・差集合をそれぞれ $X \cup Y$，$X \cap Y$，$X \setminus Y$ と表す.

予備知識 I

任意の集合 M について，M の 要素数 (*cardinality*) および べき集合 (*power set*) をそれぞれ $\mathrm{card}(M)$，$\wp(M)$ と表す.

任意のふたつの集合 X, Y について，X と Y の和集合・積集合・差集合をそれぞれ $X \cup Y$，$X \cap Y$，$X \setminus Y$ と表す.

さらに，空集合 (*empty set*) を $\emptyset$ と書く.

予備知識 I

任意の集合 M について，M の 要素数 (*cardinality*) および べき集合 (*power set*) をそれぞれ $\mathrm{card}(M)$，$\wp(M)$ と表す．

任意のふたつの集合 X, Y について，X と Y の和集合・積集合・差集合をそれぞれ $X \cup Y$，$X \cap Y$，$X \setminus Y$ と表す．

さらに，空集合 (*empty set*) を $\emptyset$ と書く．

$\mathbb{N} = \{0, 1, 2, \ldots\}$ で，すべての自然数の集合を表す．また，$\mathbb{N}^+ = \mathbb{N} \setminus \{0\}$ とする．

予備知識 II

可算個の集合 $X_0, X_1, \ldots$ について，すべての X_i の和集合を $\bigcup_{i\in\mathbb{N}} X_i$ と表す．すなわち，

$$\bigcup_{i\in\mathbb{N}} X_i \quad = \quad X_0 \cup X_1 \cup \cdots \cup X_n \cup \cdots . \tag{1}$$

予備知識 II

可算個の集合 $X_0, X_1, \ldots$ について，すべての X_i の和集合を $\bigcup_{i\in\mathbb{N}} X_i$ と表す．すなわち，

$$\bigcup_{i\in\mathbb{N}} X_i \quad = \quad X_0 \cup X_1 \cup \cdots \cup X_n \cup \cdots . \tag{1}$$

同様に，すべての X_i の積集合を表すには $\bigcap_{i\in\mathbb{N}} X_i$ と書く．つまり，

$$\bigcap_{i\in\mathbb{N}} X_i = X_0 \cap X_1 \cap \cdots \cap X_n \cap \cdots . \tag{2}$$

予備知識 III

次の概念は有用である．X, Y を任意の集合，$f\colon X \to Y$ を関数とする．任意の $y \in Y$ について，以下を定義する．

$$f^{-1}(y) = \{x \mid x \in X,\ f(x) = y\}\,. \tag{3}$$

$f^{-1}(y)$ を，y の原像 (*pre-images*) の集合とする．

予備知識 III

次の概念は有用である．X, Y を任意の集合，$f\colon X \to Y$ を関数とする．任意の $y \in Y$ について，以下を定義する．

$$f^{-1}(y) = \{x \mid x \in X,\ f(x) = y\}. \tag{3}$$

$f^{-1}(y)$ を，y の原像 (*pre-images*) の集合とする．
さらに，次の定義が必要となる．

定義 1

X, Y を任意の集合，$f\colon X \to Y$ を関数とする．

(1) すべての $x, y \in X$ について，$f(x) = f(y)$ ならば $x = y$ であるとき，f は単射 ***(injective)*** であると言われる．

(2) すべての $y \in Y$ について，$f(x) = y$ なる $x \in X$ が存在するとき，f は全射 ***(surjective)*** であると言われる．

(3) f が，単射かつ全射であるとき，全単射 ***(bijective)*** であると言われる．

予備知識 IV

X と Y を任意の集合とする．全単射 f: X → Y が存在するとき，X と Y は同じ要素数であると言う．もし，集合 X が自然数の集合 $\mathbb{N}$ と同じ要素数であるとき，X は 可算無限 (*countably infinite*) であると言われる．

予備知識 IV

X と Y を任意の集合とする．全単射 $f\colon X \to Y$ が存在するとき，X と Y は同じ要素数であると言う．もし，集合 X が自然数の集合 $\mathbb{N}$ と同じ要素数であるとき，X は 可算無限 (*countably infinite*) であると言われる．

集合 X が有限，あるいは，可算無限であるとき，X は高々可算無限 (*at most countably infinite*) である．もし，$X \neq \emptyset$ かつ X が高々可算無限であるとき，全射 $f\colon \mathbb{N} \to X$ が存在する．すなわち，

$$X = \{f(0), f(1), f(2), \ldots\}.$$

つまり，直観的には，X のすべての要素を列挙することが可能である (ここで，繰り返しは許されるものとする)．

予備知識 V

次の定理は基礎となる重要なものである．

定理 1 (カントールの定理)

すべての可算無限集合 X について，集合 $\wp(X)$ は可算無限ではない．

予備知識 V

次の定理は基礎となる重要なものである．

定理 1 (カントールの定理)

すべての可算無限集合 X について，集合 $\wp(X)$ は可算無限ではない．

X は可算無限なので，全単射 $f\colon \mathbb{N} \to X$ が存在する．$\wp(X)$ が可算無限であると仮定すると，全単射 $g\colon \mathbb{N} \to \wp(X)$ が存在するはずである．

予備知識 V

次の定理は基礎となる重要なものである．

定理 1 (カントールの定理)

すべての可算無限集合 X について，集合 $\wp(X)$ は可算無限ではない．

X は可算無限なので，全単射 $f\colon \mathbb{N} \to X$ が存在する．$\wp(X)$ が可算無限であると仮定すると，全単射 $g\colon \mathbb{N} \to \wp(X)$ が存在するはずである．
ここで，対角集合 D を次の通り定義する：

$$D = \{f(j) \mid j \in \mathbb{N},\ f(j) \notin g(j)\}.$$

構成法より，$D \subseteq X$ となり，よって，$D \in \wp(X)$ である．故に，$D = g(d)$ なる $d \in \mathbb{N}$ が存在する．

予備知識 VI

いま，$f(d)$ を考える．f の定義から，$f(d) \in X$ であることがわかる．$D \subseteq X$ であるから，$f(d) \in D$ あるいは $f(d) \notin D$ のいずれかの場合が考えられる．

予備知識 VI

いま，$f(d)$ を考える．f の定義から，$f(d) \in X$ であることがわかる．$D \subseteq X$ であるから，$f(d) \in D$ あるいは $f(d) \notin D$ のいずれかの場合が考えられる．

場合 1. $f(d) \in D$.

D および d の定義から，直ちに

$$f(d) \in D \iff f(d) \notin g(d) \iff f(d) \notin D\,,$$

を得る．なぜなら $g(d) = D$ であるから．この矛盾は，場合 1 が起こり得ないことを示す．

予備知識 VII

場合 2. $f(d) \notin D$.

構成法から，次が成り立つとき，かつ，そのときに限り $f(d) \notin D$ である：$f(d) \in D$ ならば，かつそのときに限り，$f(d) \in g(d)$ である．よって，再び矛盾となる．

以上より，場合 2 も起こり得ない．よって，$\wp(X)$ が可算無限であるとの仮定は成立し得ない． ∎

関係 I

X, Y を任意の非空の集合とし，
$X \times Y = \{(x, y) \mid x \in X \text{ かつ } y \in Y\}$
とする．
すべての $\rho \subseteq X \times Y$ は，二項関係 (*binary relation*) であると言われる．

$(x, y) \in \rho$ と書く代わりに，$x \rho y$ なる表記も度々用いられる．
$X = Y$ の場合は，特に重要である．$\rho \subseteq X \times X$ のとき，ρ は X 上の二項関係とも言われる．

関係 II

定義 2

$X \neq \emptyset$ を任意の集合，ρ を X 上の任意の二項関係とする．

(1) すべての $x \in X$ について，$(x,x) \in \rho$ であるとき，ρ は反射的 *(reflexive)* であると言われる．

(2) すべての $x,y \in X$ について，$(x,y) \in \rho$ ならば $(y,x) \in \rho$ であるとき，ρ は対称的 *(symmetric)* であると言われる．

(3) すべての $x,y,z \in X$ について，$(x,y) \in \rho$ かつ $(y,z) \in \rho$ ならば $(x,z) \in \rho$ であるとき，ρ は推移的 *(transitive)* であると言われる．

(4) $(x,y) \in \rho$ かつ $(y,x) \in \rho$ ならば $x = y$ であるとき，ρ は反対称的 *(antisymmetric)* であると言われる．

関係 II

定義 2

$X \neq \emptyset$ を任意の集合，ρ を X 上の任意の二項関係とする．

(1) すべての $x \in X$ について，$(x,x) \in \rho$ であるとき，ρ は反射的 *(reflexive)* であると言われる．

(2) すべての $x,y \in X$ について，$(x,y) \in \rho$ ならば $(y,x) \in \rho$ であるとき，ρ は対称的 *(symmetric)* であると言われる．

(3) すべての $x,y,z \in X$ について，$(x,y) \in \rho$ かつ $(y,z) \in \rho$ ならば $(x,z) \in \rho$ であるとき，ρ は推移的 *(transitive)* であると言われる．

(4) $(x,y) \in \rho$ かつ $(y,x) \in \rho$ ならば $x = y$ であるとき，ρ は反対称的 *(antisymmetric)* であると言われる．

(1) から (3) を満たす任意の二項関係は，同値関係 (*equivalence relation*) と呼ばれる．任意の $x \in X$ について，x により生成される同値類 (*equivalence class*) を $[x]$ と書く．すなわち，
$[x] = \{y \mid y \in X \text{ かつ } (x,y) \in \rho\}$．

関係 III

(1)，(3) および (4) を満たす任意の二項関係を半順序 (*partial order*) と呼ぶ．このとき，(X, ρ) は半順序集合 (*partially ordered set*) であると言われる．

関係 III

(1)，(3) および (4) を満たす任意の二項関係を半順序 (*partial order*) と呼ぶ．このとき，(X, ρ) は半順序集合 (*partially ordered set*) であると言われる．

定義 3

$\rho \subseteq X \times Y$ および $\tau \subseteq Y \times Z$ を二項関係とする．ρ と τ の**合成 (*composition*)** は，以下の通り定義される二項関係 $\zeta \subseteq X \times Z$ である：

$$\begin{aligned} \zeta &= \rho\tau \\ &= \{(x,z) \mid (x,y) \in \rho,\ (y,z) \in \tau \text{ なる } y \in Y \text{ が存在する } \}. \end{aligned}$$

関係 IV

いま，$X \neq \emptyset$ を任意の集合とする．恒等関係 (*equality*) と呼ばれる特別な二項関係 ρ^0 があり，次の通り定義される．

$\rho^0 = \{(x, x) \mid x \in X\}$.

また，$\rho \subseteq X \times X$ を任意の二項関係としたとき，各 $i \in \mathbb{N}$ について，$\rho^{i+1} = \rho^i \rho$ を帰納的に定義する．

関係 IV

いま，$X \neq \emptyset$ を任意の集合とする．恒等関係 (*equality*) と呼ばれる特別な二項関係 ρ^0 があり，次の通り定義される．
$\rho^0 = \{(x, x) \mid x \in X\}$.
また，$\rho \subseteq X \times X$ を任意の二項関係としたとき，各 $i \in \mathbb{N}$ について，$\rho^{i+1} = \rho^i \rho$ を帰納的に定義する．

定義 4

$X \neq \emptyset$ を任意の集合，ρ を X 上の任意の二項関係とする．ρ の反射的推移的閉包 (*reflexive-transitive closure*) は二項関係 $\rho^* = \bigcup_{i \in \mathbb{N}} \rho^i$ である．

関係 IV

いま，$X \neq \emptyset$ を任意の集合とする．恒等関係 *(equality)* と呼ばれる特別な二項関係 ρ^0 があり，次の通り定義される．
$\rho^0 = \{(x, x) \mid x \in X\}$.
また，$\rho \subseteq X \times X$ を任意の二項関係としたとき，各 $i \in \mathbb{N}$ について，$\rho^{i+1} = \rho^i \rho$ を帰納的に定義する．

定義 4

$X \neq \emptyset$ を任意の集合，ρ を X 上の任意の二項関係とする．ρ の反射的推移的閉包 *(reflexive-transitive closure)* は二項関係 $\rho^* = \bigcup_{i \in \mathbb{N}} \rho^i$ である．

次に，文字列および文字列の集合を扱うための形式化を行なう．

アルファベット

アルファベット (*alphabet*) と呼ばれる非空の有限集合を Σ と表す．Σ 中の要素は，これ以上分解できない記号であると仮定され，これらを文字 (*letters*)，あるいは，記号 (*symbols*) と呼ぶ．

例：
$\Sigma = \{0, 1\}$ は文字 0 と 1 を含むアルファベットである．
$\Sigma = \{a, b, c\}$ 文字 a，b および c を含むアルファベットである．

コンパイル等においては，例えば **begin** や **end** を含むアルファベットを考えることもある．ただし，**begin** や **end** は分解できないものと仮定される．

アルファベット

アルファベット (*alphabet*) と呼ばれる非空の有限集合を Σ と表す．
Σ 中の要素は，これ以上分解できない記号であると仮定され，これらを文字 (*letters*)，あるいは，記号 (*symbols*) と呼ぶ．

例：
$\Sigma = \{0, 1\}$ は文字 0 と 1 を含むアルファベットである．
$\Sigma = \{a, b, c\}$ 文字 a，b および c を含むアルファベットである．

コンパイル等においては，例えば **begin** や **end** を含むアルファベットを考えることもある．ただし，**begin** や **end** は分解できないものと仮定される．

アルファベット

アルファベット (*alphabet*) と呼ばれる非空の有限集合を Σ と表す．Σ 中の要素は，これ以上分解できない記号であると仮定され，これらを文字 (*letters*)，あるいは，記号 (*symbols*) と呼ぶ．

例：
$\Sigma = \{0, 1\}$ は文字 0 と 1 を含むアルファベットである．
$\Sigma = \{a, b, c\}$ 文字 a，b および c を含むアルファベットである．

コンパイル等においては，例えば **begin** や **end** を含むアルファベットを考えることもある．ただし，**begin** や **end** は分解できないものと仮定される．

文字列

定義 5

アルファベット Σ 上の**文字列 *(string)*** とは，Σ 中の文字からなる有限長の系列である．典型的な文字列は $s = a_1 a_2 \cdots a_k$ と書かれる．ここで，$i = 1, \ldots, k$ について $a_i \in \Sigma$ である．

文字列

定義 5

アルファベット Σ 上の文字列 *(string)* とは，Σ 中の文字からなる有限長の系列である．典型的な文字列は $s = a_1 a_2 \cdots a_k$ と書かれる．ここで，$i = 1, \ldots, k$ について $a_i \in \Sigma$ である．

$k = 0$ の場合は，空 *(empty)* 文字列となり，これを λ で表す．k は s の長さ *(length)* と呼ばれ，$|s|$ で表すものとする．つまり，$|\lambda| = 0$ である．Σ 上のすべての文字列の集合を Σ^* で表す．また，$\Sigma^+ = \Sigma^* \setminus \{\lambda\}$ とする．

連接

$s, w \in \Sigma^*$ とする．連接 (*concatenation*) (あるいは，語の直積) と呼ばれる二項演算を定義する．s と w の連接とは，文字列 sw のことである．

連接

$s, w \in \Sigma^*$ とする．連接 (*concatenation*) (あるいは，語の直積) と呼ばれる二項演算を定義する．s と w の連接とは，文字列 sw のことである．

例：$\Sigma = \{0, 1\}$，$s = 000111$ および $w = 0011$ とする．
このとき，$sw = 0001110011$ である．

連接の性質

命題 1

Σ を任意のアルファベットとする．

(1) 連接は結合的である．すなわち，すべての $x, y, z \in \Sigma^*$ について，$x(yz) = (xy)z$.

(2) 空文字列 λ は，Σ^* における両側単位元である．すなわち，すべての $x \in \Sigma^*$ について，

$$x\lambda = \lambda x = x\,.$$

(3) Σ^* には自明でない単位元が存在しない．すなわち，すべての $x, y, z \in \Sigma^*$ について，

i) $zx = zy$ ならば $x = y$ かつ，
ii) $xz = yz$ ならば $x = y$.

(4) すべての $x, y \in \Sigma^*$ について，$|xy| = |x| + |y|$ である．

文字列集合への拡張

X, Y を文字列の集合とする．このとき，X と Y の直積 (*product*) を次の通り定義する．

$$XY = \{xy \mid x \in X \text{ かつ } y \in Y\}.$$

$X \subseteq \Sigma^*$ について，$X^0 = \{\lambda\}$ とし，すべての $i \geqslant 0$ について，$X^{i+1} = X^i X$ と定める．X の クリーニ閉包 (*Kleene closure*) を，$X^* = \bigcup_{i \in \mathbb{N}} X^i$ と定義する．また，X の 半群閉包 (*semigroup closure*) を，$X^+ = \bigcup_{i \in \mathbb{N}^+} X^i$ とする．

最後に文字列と文字列集合の転置・反転 (*transpose*) を定義する．

文字列集合への拡張

X, Y を文字列の集合とする．このとき，X と Y の直積 (*product*) を次の通り定義する．

$$XY = \{xy \mid x \in X \text{ かつ } y \in Y\}.$$

$X \subseteq \Sigma^*$ について，$X^0 = \{\lambda\}$ とし，すべての $i \geqslant 0$ について，$X^{i+1} = X^i X$ と定める．X の クリーニ閉包 (*Kleene closure*) を，$X^* = \bigcup_{i \in \mathbb{N}} X^i$ と定義する．また，X の 半群閉包 (*semigroup closure*) を，$X^+ = \bigcup_{i \in \mathbb{N}^+} X^i$ とする．

最後に文字列と文字列集合の転置・反転 (*transpose*) を定義する．

定義 6

Σ を任意のアルファベットとする．Σ^* 中の文字列と，文字列集合 $X \subseteq \Sigma^*$ に対する転置 ***(transpose)*** 演算を次の通り定義する：

$$\begin{aligned} \lambda^T &= \lambda, \text{ and} \\ (xa)^T &= a(x^T) \text{ for all } x \in \Sigma^* \text{ and } a \in \Sigma \\ X^T &= \{x^T \mid x \in X\}. \end{aligned}$$

言語 I

定義 7

Σ を任意のアルファベットとする．任意の部分集合 $L \subseteq \Sigma^*$ は言語 ***(language)*** と呼ばれる．

言語 I

定義 7

Σ を任意のアルファベットとする．任意の部分集合 $L \subseteq \Sigma^*$ は言語 ***(language)*** と呼ばれる．

空集合，および，$L = \{\lambda\}$ もまた言語であることに注意しよう．

言語 I

定義 7

Σ を任意のアルファベットとする．任意の部分集合 $L \subseteq \Sigma^*$ は言語 ***(language)*** と呼ばれる．

空集合，および，$L = \{\lambda\}$ もまた言語であることに注意しよう．

次に，どのくらいの数の言語が存在するかを考える．m を Σ の要素数とする．長さ 0 の文字列 (つまり λ) はただひとつ，また，長さ 1 の文字列 (つまり，すべての $a \in \Sigma$ について a) は m 存在する．さらに，長さ 2 の文字列は m^2 に存在し，一般には，長さ n の文字列は m^n 存在する．よって，Σ^* の要素数は，可算無限 ***(countably infinite)*** となる．

言語 II

カントールの定理より，$\mathrm{card}(M) < \mathrm{card}(\wp(M))$ であることがわかる．よって，非可算 ***(uncountably)*** 個の言語が存在すると結論できる (実数の総数と同じ).
言語の生成と識別はアルゴリズム論的に行なわれるべきであるから，可算個の言語のみがアルゴリズムによって生成・識別されることが直ちにわかる.

回文 I

回文 (*palindrome*) は，左から読んでも右から読んでも同じ文字列である．例えば，

AKASAKA

回文 I

回文 (*palindrome*) は，左から読んでも右から読んでも同じ文字列である．例えば，

AKASAKA

トビコミコビト
にわとりとことりとわに
テングノハハノグンテ

回文 I

回文 (*palindrome*) は，左から読んでも右から読んでも同じ文字列である．例えば，

AKASAKA

トビコミコビト
にわとりとことりとわに
テングノハハノグンテ

ここで，アルファベット $\{a, b\}$(話を簡単にするため) 上のすべての回文から成る言語がどのように記述できるかを考えよう．

回文 II

それでは始めよう．もちろん，λ，a および b は回文である．すべての回文は，同じ文字で始まり，そして，終らなければならない．また，最初と最後の文字を取り除いても，やはり回文が得られる．こうした観察は，L_{pal} の定義における次の基底と帰納ステップをもたらす．

回文 II

それでは始めよう．もちろん，λ，a および b は回文である．すべての回文は，同じ文字で始まり，そして，終らなければならない．また，最初と最後の文字を取り除いても，やはり回文が得られる．こうした観察は，L_{pal} の定義における次の基底と帰納ステップをもたらす．

基底： λ, a および b は回文である．

回文 II

それでは始めよう．もちろん，λ，a および b は回文である．すべての回文は，同じ文字で始まり，そして，終らなければならない．また，最初と最後の文字を取り除いても，やはり回文が得られる．こうした観察は，L_{pal} の定義における次の基底と帰納ステップをもたらす．

基底：λ, a および b は回文である．

帰納ステップ：もし $w \in \{a, b\}^*$ が回文であるならば，awa と bwb もまた回文である．また，基底と帰納ステップに従うもの以外の $w \in \{a, b\}^*$ は，回文ではない．

回文 II

それでは始めよう．もちろん，λ，a および b は回文である．すべての回文は，同じ文字で始まり，そして，終らなければならない．また，最初と最後の文字を取り除いても，やはり回文が得られる．こうした観察は，L_{pal} の定義における次の基底と帰納ステップをもたらす．

基底：λ, a および b は回文である．

帰納ステップ：もし $w \in \{a, b\}^*$ が回文であるならば，awa と bwb もまた回文である．また，基底と帰納ステップに従うもの以外の $w \in \{a, b\}^*$ は，回文ではない．

しかし，ここで止めておく．すべての回文から成る言語を定義する際，転置演算子 T を用いることができる．すなわち，
$\tilde{L}_{pal} = \{w \in \{a, b\}^* \mid w = w^T\}$.

回文 III

いまの定義において，敢えて異なる表記を用いた．なぜなら，$L_{pal} = \tilde{L}_{pal}$ であるか否かが，まだわからないからである．この等式を得るには，証明が必要である．

回文 III

いまの定義において，敢えて異なる表記を用いた．なぜなら，$L_{pal} = \tilde{L}_{pal}$ であるか否かが，まだわからないからである．この等式を得るには，証明が必要である．

定理 2

$L_{pal} = \tilde{L}_{pal}$.

回文 III

いまの定義において，敢えて異なる表記を用いた．なぜなら，$L_{pal} = \tilde{L}_{pal}$ であるか否かが，まだわからないからである．この等式を得るには，証明が必要である．

定理 2

$L_{pal} = \tilde{L}_{pal}$.

証明： 集合 X, Y が等しいことは，多くは $X \subseteq Y$ および $Y \subseteq X$ を示すことで証明される．それではまず，$L_{pal} \subseteq \tilde{L}_{pal}$ を示そう．

回文 IV

基底で定義される文字列，λ，a および b から始めよう．転置演算の定義より，$\lambda^T = \lambda$ であるから，$\lambda \in \tilde{L}_{pal}$ となる．次に a を考える．転置演算の定義を適用するために，命題 1の性質 (2)，すなわち，$a = \lambda a$ を用いる．すると，

$$a^T = (\lambda a)^T = a\lambda^T = a\lambda = a\,.$$

を得る．b に関する証明は同様なので省略する．

回文 IV

基底で定義される文字列，λ，a および b から始めよう．転置演算の定義より，$\lambda^T = \lambda$ であるから，$\lambda \in \tilde{L}_{pal}$ となる．次に a を考える．転置演算の定義を適用するために，命題 1の性質 (2)，すなわち，$a = \lambda a$ を用いる．すると，

$$a^T = (\lambda a)^T = a\lambda^T = a\lambda = a\,.$$

を得る．b に関する証明は同様なので省略する．

いま，帰納法の仮定として，$|w| \leqslant n$ なるすべての文字列 w について，$w \in L_{pal}$ ならば $w \in \tilde{L}_{pal}$ であるとする．L_{pal} の定義に従い，帰納ステップは n から $n+2$ へと行なわれる．$w \in L_{pal}$ を，$|w| = n+2$ なる任意の文字列とする．すると，$w = avb$ または $w = bvb$ となる．ただし，$v \in \{a,b\}^*$ および $|v| = n$ である．L_{pal} の定義と，帰納法の仮定より，v は回文であるから，$v = v^T$ であることがわかる．

回文 V

次の主張は，転置演算の特別な性質を与えるものである．

主張 1. 任意のアルファベット Σ 上の文字列を $w = w_1 \dots w_n \in \Sigma^*$ とする．ただし，$n \in \mathbb{N}^+$ で，すべての $i \in \{1, \dots, n\}$ について，$w_i \in \Sigma$ である．このとき，$w^T = w_n \dots w_1$ である．

回文 V

次の主張は，転置演算の特別な性質を与えるものである．

主張 1. 任意のアルファベット Σ 上の文字列を $w = w_1 \dots w_n \in \Sigma^*$ とする．ただし，$n \in \mathbb{N}^+$ で，すべての $i \in \{1, \dots, n\}$ について，$w_i \in \Sigma$ である．このとき，$w^T = w_n \dots w_1$ である．

主張 2. すべての $n \in \mathbb{N}$ について，もし $p = p_1 x p_{n+2}$ ならば，すべての $p_1, p_{n+2} \in \{a, b\}$ および $x \in \{a, b\}^*$ について，$p^T = p_{n+2} x^T p_1$ である．ただし，$|x| = n$ である．

回文 V

次の主張は，転置演算の特別な性質を与えるものである．

主張 1. 任意のアルファベット Σ 上の文字列を $w = w_1 \dots w_n \in \Sigma^*$ とする．ただし，$n \in \mathbb{N}^+$ で，すべての $i \in \{1, \dots, n\}$ について，$w_i \in \Sigma$ である．このとき，$w^T = w_n \dots w_1$ である．

主張 2. すべての $n \in \mathbb{N}$ について，もし $p = p_1 x p_{n+2}$ ならば，すべての $p_1, p_{n+2} \in \{a, b\}$ および $x \in \{a, b\}^*$ について，$p^T = p_{n+2} x^T p_1$ である．ただし，$|x| = n$ である．

主張 1 は，主張 2 を示す際に必要となる．これらの証明については演習問題とする．

主張 2 を用いると次を得る．

$$w^T = (avb)^T = av^T b \underbrace{=}_{\text{by IH}} avb = w\,.$$

$w = bvb$ の場合も同様なので省略する．

回文 VI

最後に，$\tilde{L}_{pal} \subseteq L_{pal}$ を示す．
帰納法の基底として，$\lambda = \lambda^T$, すなわち，$\lambda \in \tilde{L}_{pal}$ であることがわかっている．また，L_{pal} の定義における基底から，$\lambda \in L_{pal}$ であることもわかる．
こうして，次の帰納法の仮定を得る．長さ n のすべての文字列 w について：$w = w^T$ ならば $w \in L_{pal}$ である．

回文 VI

最後に，$\tilde{L}_{pal} \subseteq L_{pal}$ を示す．
帰納法の基底として，$\lambda = \lambda^T$, すなわち，$\lambda \in \tilde{L}_{pal}$ であることがわかっている．また，L_{pal} の定義における基底から，$\lambda \in L_{pal}$ であることもわかる．
こうして，次の帰納法の仮定を得る．長さ n のすべての文字列 w について：$w = w^T$ ならば $w \in L_{pal}$ である．

帰納ステップは n から $n+1$ へと行なわれる．すなわち，次を示す：$|w| = n+1$ かつ $w = w^T$ ならば，$w \in L_{pal}$ である．

回文 VI

最後に，$\tilde{L}_{pal} \subseteq L_{pal}$ を示す．
帰納法の基底として，$\lambda = \lambda^T$, すなわち，$\lambda \in \tilde{L}_{pal}$ であることがわかっている．また，L_{pal} の定義における基底から，$\lambda \in L_{pal}$ であることもわかる．
こうして，次の帰納法の仮定を得る．長さ n のすべての文字列 w について：$w = w^T$ ならば $w \in L_{pal}$ である．

帰納ステップは n から $n+1$ へと行なわれる．すなわち，次を示す：$|w| = n+1$ かつ $w = w^T$ ならば，$w \in L_{pal}$ である．

$n = 1$ の場合は直ちに a および b が得られ，また，$a, b \in L_{pal}$ であるから，以下では $n > 1$ を仮定する．

回文 VII

では，$|w| = n + 1$ かつ $w = w^T$ なる任意の文字列 $w \in \{a, b\}^*$，例えば，$w = a_1 \dots a_{n+1}$ (ただし，$a_i \in \Sigma$) を考える．仮定より，次を得る．

$$a_1 \dots a_{n+1} = a_{n+1} \dots a_1 .$$

回文 VII

では，$|w| = n + 1$ かつ $w = w^T$ なる任意の文字列 $w \in \{a, b\}^*$，例えば，$w = a_1 \dots a_{n+1}$ (ただし，$a_i \in \Sigma$) を考える．仮定より，次を得る．

$$a_1 \dots a_{n+1} = a_{n+1} \dots a_1 .$$

命題 1の性質 (3) を適用すると，直ちに $a_1 = a_{n+1}$ が得られる．ここで，$a_1 = a$ および $a_1 = b$ の場合を区別する必要がある．両者とも同様に扱うことができるので，ここでは $a_1 = a$ の場合のみを考える．すると，$w = avа$ と結論できる．ただし $v \in \{a, b\}^*$ かつ $|v| = n - 1$ である．次に，先に示した転置演算の性質を適用すると，$v = v^T$，すなわち，$v \in L_{pal}$ が得られる．最後に，L_{pal} の定義における “帰納” の部分より，$w \in L_{pal}$ であることが直ちにわかる．∎

Thank you!