

## 代表的な計算モデル

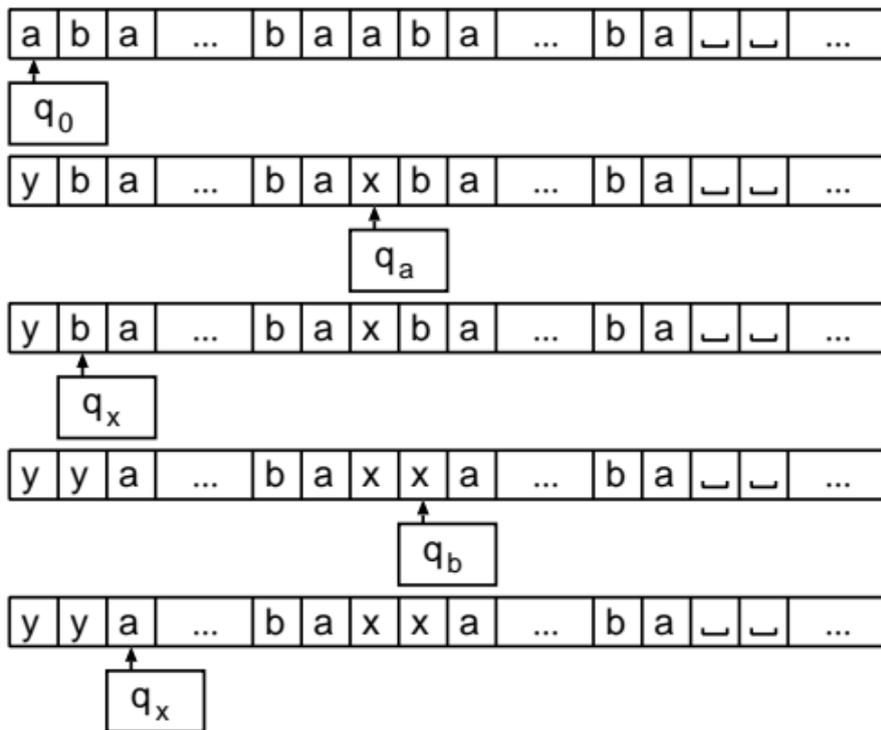
- 有限オートマトン（有限状態機械）
- プッシュダウンオートマトン
- チューリングマシン

## チューリングマシン

- 有限個の内部状態を持つ
- 入力データはテープ上に一区画一文字ずつ書き込まれて与えられる
- データを読み書きするヘッドがテープ上を動く
- 遷移関数は次の形：  
内部状態とヘッドが今いる場所の文字とによって、その場所の文字を書き換え、次の内部状態に移り、ヘッドを左か右かに動かす
- 受理状態または拒否状態に達したら停止するが、停止しないこともある

# (非決定性) チューリングマシンによる

言語  $A = \{ww \mid w \in \Sigma^*\}$  の認識



## チューリングマシンによる言語の認識

チューリングマシン  $T$  が言語  $A$  を認識する



$$A = \left\{ w \in \Sigma^* \mid \begin{array}{l} \text{入力 } w \text{ に対し、} \\ \text{受理状態で停止する} \\ \text{遷移が存在} \end{array} \right\}$$



$w \in A \iff$  入力  $w$  に対し、  
受理状態で停止する遷移が存在

## チューリングマシンによる言語の判定

チューリングマシン  $T$  が言語  $A$  を判定する



$T$  は  $A$  を認識し、  
かつ、全ての入力に対し必ず停止する



$w \in A \iff$  入力  $w$  に対し、  
受理状態で停止する遷移が存在  
かつ

$w \notin A \iff$  入力  $w$  に対し、  
すべての遷移が拒否状態で停止

## Church-Turing の提唱

「全てのアルゴリズム（計算手順）は、  
チューリングマシンで実装できる」

（アルゴリズムと呼べるのは  
チューリングマシンで実装できるものだけ）

… 「アルゴリズム」の定式化

## 何故「チューリングマシン」なのか？

- およそ計算機で実行したいことは模倣可能  
(非有界のメモリにランダムアクセスできる  
計算機モデル)
- 多少モデルを変更しても強さが同じ  
(モデルの頑強性)
  - ★ テープが両方に無限に伸びているか
  - ★ ヘッドが動かないことがあっても良いか
  - ★ 複数テープチューリングマシン
  - ★ 決定性 / 非決定性                      などなど

プログラム内蔵方式 ( von Neumann 型 )

… プログラムもデータとして保持

→ 一つの機械で様々な計算を柔軟に実現

同様の働きをするチューリングマシンが構成できる

… 普遍チューリングマシン

( 万能チューリングマシン )

( universal Turing machine )

## 普遍チューリングマシン

全てのチューリングマシンの動作を模倣する

- 入力： $(\langle M \rangle, w)$ 
  - ★  $\langle M \rangle$ ：機械  $M$  の符号化（プログラムに相当）
  - ★  $w$ ： $M$  に与える入力データ
  
- 出力：機械  $M$  が入力  $w$  を受理するかどうか

## 普遍チューリングマシン

普遍チューリングマシンとは、  
言語

$$A_{\text{TM}} = \left\{ (\langle M \rangle, w) \mid \begin{array}{l} \langle M \rangle : \text{TM } M \text{ の符号化} \\ M \text{ が入力 } w \text{ を受理} \end{array} \right\}$$

を認識するチューリングマシン

普遍チューリングマシンが存在

$\iff A_{\text{TM}}$  がチューリングマシンで認識可能

## 定理

### 言語

$$A_{\text{TM}} = \left\{ (\langle M \rangle, w) \mid \begin{array}{l} \langle M \rangle : \text{TM } M \text{ の符号化} \\ M \text{ が入力 } w \text{ を受理} \end{array} \right\}$$

は認識可能だが、判定可能ではない。

証明は一種の対角線論法による  
(Russell のパラドックス風)

## 対角線論法の例：Russell のパラドックス

$X := \{A \mid A \notin A\}$  とせよ

$X \in X$  であるか？

- $X \in X$  と仮定すると、定義より  $X \notin X$
- $X \notin X$  と仮定すると、定義より  $X \in X$

→ どちらにしても**矛盾** !!

## $A_{TM}$ の判定不可能性

$A_{TM}$  を判定する TM  $U$  があったとする。

入力  $\langle M \rangle$  に対し、

- $M$  が  $\langle M \rangle$  を受理するなら拒否
- $M$  が  $\langle M \rangle$  を拒否するなら受理

となる TM  $D$  が ( $U$  を使って) 作れる。

これに、入力  $\langle D \rangle$  を喰わせよ。

対角線論法が出てきたついでに...

対角線論法の例：冪集合の濃度

集合  $X$  の冪集合 (power set)

$$\mathcal{P}(X) = \{S \mid S \subset X\}$$

について、

$$\#X \not\leq \#\mathcal{P}(X)$$

## 集合の濃度

集合  $A, B$  に対し、

$$A \prec B \iff \exists \iota : A \longrightarrow B : \text{単射}$$

$$\iff \exists \pi : B \longrightarrow A : \text{全射}$$

( $\iff$  には選択公理が必要)

$$A \sim B \iff \exists \varphi : A \longrightarrow B : \text{全単射}$$

$$\iff A \prec B \text{ かつ } B \prec A$$

(Bernstein の定理)

$\sim$  は “集合全体の集まり” の上の “同値関係”

## 集合の濃度

A の属する “同値類” : A の濃度 (cardinality)  
( $\#A, |A|, \text{card}(A)$  等と書く)

- $\aleph_0 = \aleph := \#\mathbb{N}$  : 可算濃度 (可付番濃度)  
(countable, enumerable)
- $\aleph = \mathfrak{c} := \#\mathbb{R}$  : 連続体濃度 (continuum)

濃度の比較 :  $\#A \leq \#B \iff A \prec B$

- $\#A \leq \#A$
  - $\#A \leq \#B, \#B \leq \#A \implies \#A = \#B$
  - $\#A \leq \#B, \#B \leq \#C \implies \#A \leq \#C$
- ( $\leq$  は濃度の間 “順序関係” である)

## 対角線論法の例：冪集合の濃度

集合  $X$  の冪集合 (power set)

$$\mathcal{P}(X) = \{S \mid S \subset X\}$$

について、

$$\#X \prec \# \mathcal{P}(X)$$

応用：

$$\begin{aligned} \# \mathbb{N} = \# \mathbb{Q} = \aleph_0 & \text{ (可算濃度) だが、} \\ \# \mathbb{R} = \# \mathcal{P}(\mathbb{N}) = \aleph & \succ \aleph_0 \text{ (連続体濃度)} \end{aligned}$$

注： $\aleph$  は  $\aleph_0$  の次の大きさ、とは言えない  
(連続体仮説)

## 定理

チューリングマシンで認識可能でない言語が存在する。

- チューリングマシン全体の集合
- 言語全体の集合

の濃度とを比較せよ

さて、本講義最後の話題は、

## 計算量

について

問題の難しさを如何に計るか？

## Church-Turing の提唱 (再掲)

「全てのアルゴリズム (計算手順) は、  
チューリングマシンで実装できる」

(アルゴリズムと呼べるのは  
チューリングマシンで実装できるものだけ)

… 「アルゴリズム」の定式化