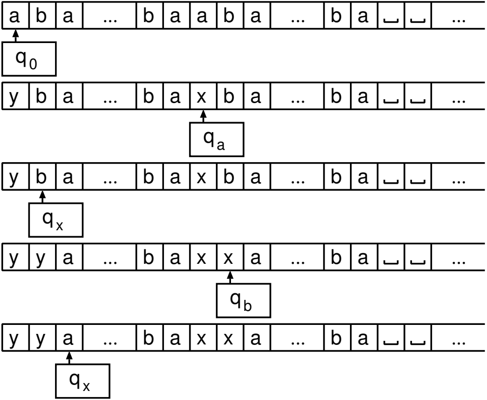
代表的な計算モデル

- 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づ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 $A = \{ww | w \in \Sigma^*\}$ 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

$$\Updownarrow$$

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

$$\Updownarrow$$

$$w \in A \iff \begin{array}{l} \text{入力 } w \text{ に対し、} \\ \text{受理状態で停止する遷移が存在} \end{array}$$

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**

$$\Updownarrow$$

T **は** A **を認識し、**
かつ、全ての入力に対し必ず停止する

$$\Updownarrow$$

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

かつ

$w \notin A \Longleftrightarrow$ **入力** w **に対し、拒否状態で停止する遷移が存在**

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）

$\cdots$ 「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
  （無限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）

$\cdots$ プログラムもデータとして保持

$\longrightarrow$ 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 $\mathbb{U}$ が構成できる

$\cdots$ 万能チューリングマシン (universal Turing machine)

プログラム内蔵方式（von Neumann 型）

$\cdots$ プログラムもデータとして保持

$\longrightarrow$ 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 $\mathbb{U}$ が構成できる

$\cdots$ 万能チューリングマシン (universal Turing machine)

プログラム内蔵方式（von Neumann 型）

$\cdots$ **プログラムもデータとして保持**

$\longrightarrow$ **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 $\mathbb{U}$ が構成できる

$\cdots$ **万能チューリングマシン (universal Turing machine)**

万能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：** $(\langle M \rangle, w)$
  $\star$ $\langle M \rangle$**：機械** M **の符号化（プログラムに相当）**
  $\star$ 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**

定理

言語

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

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）

定理

言語

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

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

$X := \{A | A \notin A\}$ 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 $X \in X$

$\longrightarrow$ どちらにしても矛盾!!

対角線論法の例：Russell のパラドックス

$X := \{A | A \notin A\}$ 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 $X \in X$

$\longrightarrow$ どちらにしても矛盾!!

対角線論法の例：Russell のパラドックス

$X := \{A | A \notin A\}$ 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 $X \in X$

$\longrightarrow$ どちらにしても矛盾!!

対角線論法の例：Russell のパラドックス

$X := \{A | A \notin A\}$ 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 $X \in X$

$\longrightarrow$ どちらにしても矛盾!!

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 $X \in X$

$\longrightarrow$ どちらにしても**矛盾**!!

$A_{\mathtt{TM}}$ の判定不可能性

$A_{\mathtt{TM}}$ を判定する TM U があったとする。

入力 $\langle \mathrm{M} \rangle$ に対し、

- **M が $\langle \mathrm{M} \rangle$ を受理するなら拒否**
- **M が $\langle \mathrm{M} \rangle$ 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 $\langle \mathrm{D} \rangle$ を喰わせよ。

$A_{\mathtt{TM}}$ の判定不可能性

$A_{\mathtt{TM}}$ 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。

$A_{\mathtt{TM}}$ の判定不可能性

$A_{\mathtt{TM}}$ 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。