計算論A / 2015年度

チューリングマシンと決定可能性
(今回は教科書の範囲外です)

角川　裕次

もくじ

＊＊＊ 本日の重要概念 ＊＊＊

チューリングマシン, Church の提唱
決定可能性

φ(・ω・)メモメモ

本日の内容 (つづき)

基本的な概念を紹介します

細かな証明はしません

後期の計算論Bの受講を強くおすすめします
- 計算の複雑さ(NP完全の概念など)について学べます
- 計算機科学の非常に重要な概念が学べます

1. チューリングマシン (Turing Machine, 以下TMと略記)

1テープTM

チューリングマシン(TM)とは

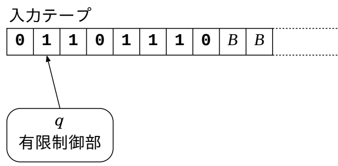
Alan Turing が提案した理論的な計算モデル
- 1936年; 電子計算機が登場以前のこと

「可能な計算」のモデル

1テープTM, 概要

有限状態数の制御部と1本のテープで構成

- テープは右方向に無限の長さ
- ヘッドは左右に動ける
- テープのヘッド位置の記号を書き換え可能



1テープTMの動作開始と終了

TM の動作開始

- テープに入力記号列が書き込まれた状態
- 入力の右端より先は全て空白文字B

TM の動作終了

- 受理状態または棄却状態に入った時

TM は永久に停止しない場合もある

- この場合は入力の受理とはならない

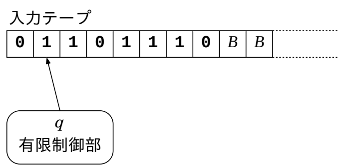
1テープTMの定義

TM は7個組 $(Q, \Sigma, \Gamma, \delta, q_0, q_{\mathrm{accept}}, q_{\mathrm{reject}})$

- Q ： 有限制御部の状態集合(有限)
- Σ ： 入力アルファベット (空白文字Bは含まない)
- Γ ： テープアルファベット ($\Sigma \subseteq \Gamma$)
- $\delta : Q \times \Gamma \to Q \times \Gamma \times \{L, R\}$ ： 遷移関数
- $q_0 \in Q$ ： 初期状態
- $q_{\mathrm{accept}} \in Q$ ： 受理状態
- $q_{\mathrm{reject}} \in Q$ ： 棄却状態（ただし$q_{\mathrm{reject}} \neq q_{\mathrm{accept}}$）

1テープTMの遷移関数$\delta : Q \times \Gamma \to Q \times \Gamma \times \{L, R\}$

状態($\in Q$)とヘッド位置の記号($\in \Gamma$)で動作が決定



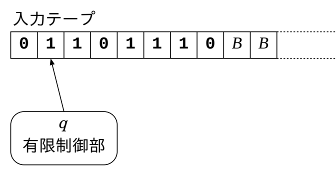
1テープTMの遷移関数$\delta : Q \times \Gamma \to Q \times \Gamma \times \{L, R\}$

次の状態($\in Q$)を決める

ヘッド位置の記号の書き換え($\in \Gamma$)を決める

- 空白Bを他の記号に書き換えても良い

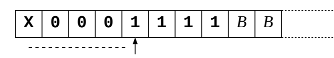
ヘッド位置を左(L)または右(R)に動かすかを決める



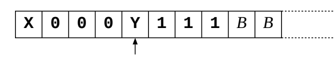
例1： $L_1 = \{0^n 1^n \mid n \geq 1\}$ の認識 (1/3)

入力の左端より動作開始し、以下を繰り返す ：

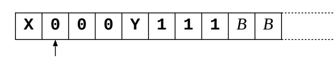
最初の0をXに置き換えて
1を見つけるまで0とYを読み飛ばし右へ移動



最初の1をYに書き換える



Xを見つけるまでYと0を読み飛ばし左へ移動し
見つけたXのすぐ右隣へ移動



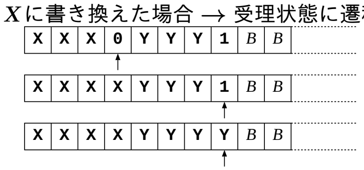
例1： $L_1 = \{0^n1^n \mid n \geq 1\}$ の認識 (2/3)

入力が $\mathbf{0^*1^*}$ 以外の形の場合

- いつかは動作できなくなる
- 棄却状態に遷移して停止

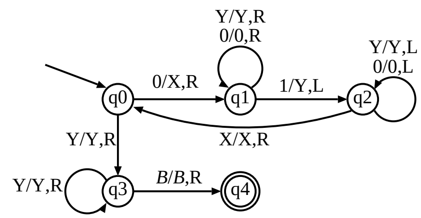
入力が $\mathbf{0^n1^n}$ の場合

- 最後の1をYに書き換えた回で
  0を全てXに書き換えた場合 $\longrightarrow$ 受理状態に遷移し停止



例1： $L_1 = \{0^n1^n \mid n \geq 1\}$ の認識 (3/3)

遷移図



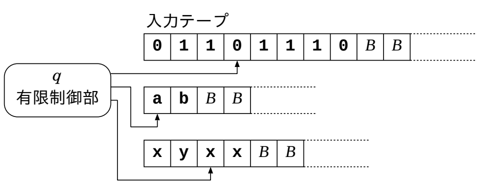
- 見方(例): 0/X,R
  — 0を読んだらそれをXに書き換えて右(R)に移動
- 図に対応する遷移がない時は棄却状態(q_5)へ遷移し停止

複数テープTM

複数テープTM：複数のテープを持つTM

1本は入力テープ

残りは作業用テープ (初期状態では内容は全てB)



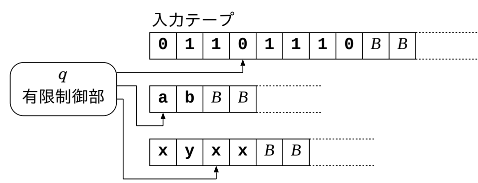
利点

- 動作記述が簡潔になる
- 認識にかかる時間を短縮できる

kテープTMの遷移関数 δ

$\delta : Q \times \Gamma^k \to Q \times \Gamma^k \times \{L, R, S\}^k$

- k個の各ヘッドは記号を読む
- k個の各ヘッドの下の記号を書き換える
- k個の各ヘッドの移動を決める (L:左, R:右, S:移動せず)



1テープTMとkテープTMは同じ能力

kテープTMの能力 $\geq$ 1テープTMの能力

- 自明

kテープTMの能力 $\leq$ 1テープTMの能力

- 1テープTMでkテープTMの動作を模倣できる
- (次スライド参照)

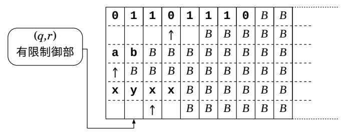
1テープTMのクラスとkテープTMのクラスは受理する言語のクラスが同じ

- (ただし受理に要する時間は等しいとは限らない)

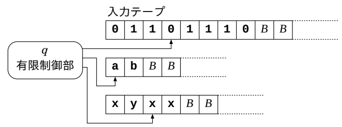
1テープTMによるkテープTMの模倣：アイディア (1/2)

1本のマルチトラックテープでk本のテープを模倣

マルチトラックテープにk個の各ヘッドの位置も記録



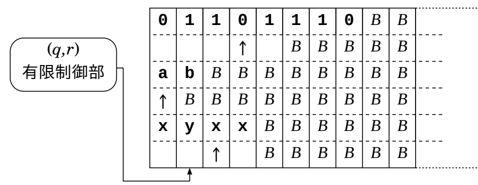
元



1テープTMによるkテープTMの模倣：アイディア (2/2)

テープの1マスを$2k$個の記号の組で構成

- 例：左端の1マスは1つの記号 $(0, -, a, \uparrow, x, -)$



1テープTMは複数のステップを用いて
kテープTMの1ステップを模倣

- テープ内容とヘッド位置を更新

非決定性TM

非決定性TM

非決定的な遷移関数を持つTM

$\delta : Q \times \Gamma \to \mathcal{P}(Q \times \Gamma \times \{L, R\})$

- 1テープTMでの場合 (kテープTMでも同様に定義可能)
- 注：$\mathcal{P}(S)$ は集合Sのべき集合

非決定性TMが受理
$\Leftrightarrow$ 非決定的な計算のうち少なくともひとつが受理

非決定性TMと決定性TMの能力の比較

非決定性TMの能力 $\geq$ 決定性TMの能力

- 自明

非決定性TMの能力 $\leq$ 決定性TMの能力

- 決定性TMで非決定性TMの動作を模倣できる
- (次スライド参照)

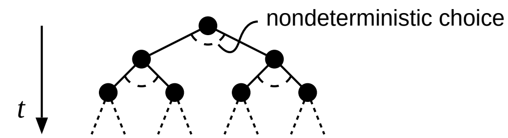
非決定性TMのクラスと決定性TMのクラスの能力は等価

- 認識できる言語のクラスの観点で等価という意味
- 注意：受理に要する時間は等しいとは限らない
  - ✔ 計算時間は重要な関心事
  - ✔ 計算論Bでより詳しく：クラスP, クラスNP, NP完全性 etc.

決定性TMによる非決定性TMの動作の模倣: アイディア

非決定性TMの計算を木構造で考える

- ●は計算状況 (根は初期計算状況)



決定性TMは各ステップを模倣して計算木をたどる

- 幅優先探索で行なう
- (深さ優先探索だと停止しない計算を永遠にたどるかも)

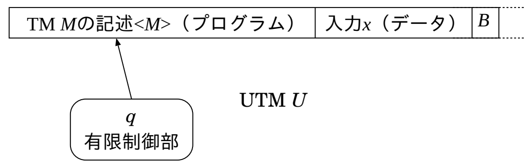
非決定性TMが受理 $\Leftrightarrow$ 決定性TMが受理

万能TM

万能TM (Universal TM, UTM) (1/2) 26

入力テープ: TM $\boldsymbol{M}$ の動作記述 $\langle \boldsymbol{M} \rangle$ と $\boldsymbol{M}$ への入力 $\boldsymbol{x}$

- $\langle \boldsymbol{M} \rangle$ は TM $\boldsymbol{M}$ を記号列で表現したもの



$\boldsymbol{U}$ の動作： TM $\boldsymbol{M}$ に $\boldsymbol{x}$ を与えた時の動作を模倣

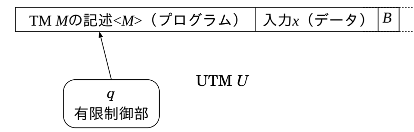
万能TM $\boldsymbol{U}$ は一般的な TM のモデルの範囲内

- 遷移関数で一種の仮想機械を実現

万能TM (Universal TM, UTM) (2/2) 27

万能TMはプログラム格納式計算機の一種

- 有限制御部 ： CPU (仮想機械)
- $\langle \boldsymbol{M} \rangle$ ： プログラム (バイトコード)
- テープ ： 作業用メモリ



注： 以前に紹介したTMは特定動作限定のもの

- 言語ごとに専用の機械(=専用の遷移関数)を用意していた

2. Churchの提唱

決定性1テープTMと変形TMの等価性 (1/2) 29

決定性1テープTMと変形TMは等価な能力

- 多テープTM
- 非決定性TM
- その他のさまざまな変形

TMの本質的特徴

- 量に関して制限のないメモリ
- メモリの使い方に制限なし (読み書き/ランダムアクセス)

決定性1テープTMと変形TMの等価性 (2/2) 30

TMの本質特徴を有する計算モデルは
いずれもTMと同等な能力を有している

- 変形TMの計算モデルに限らない
- 様々な計算モデルに対して等価性が示されている
  - ✔ 一般帰納的関数, Markovアルゴリズム,
    Postのtagシステム, etc.
  - ✔ 新しい計算モデルを考案してみた
    → 調べてみたら TM と等価だった,
    という研究の積み重ねの結果の結論

アルゴリズムとは

直観的概念： ある問題を解決する有限の命令の列

チューリングアルゴリズムとは

- TMでの基本操作を用いて書かれたアルゴリズム

様々な計算モデルでのアルゴリズム ：
チューリングアルゴリズムで記述可能

- 逆も可
- 様々な計算モデルとTMが能力的に等価なため

Churchの提唱 (あるいはChurch-Turingの提唱)

$\boxed{\text{アルゴリズムの直観的概念}}$

等価

$\boxed{\text{チューリングアルゴリズム}}$

TMで記述可能なアルゴリズムを
アルゴリズムの厳密な定義にしようという提唱

3. 決定問題

決定問題とは

決定問題 (decision problem)

Aに対するPの決定問題 (P, A) ：
各 $a \in A$ に対し $P(a)$ が真か偽かを決定する
アルゴリズムを作る問題

- P： 性質
- A： あるクラス

例

- P： グラフは連結
- A： グラフの(無限)集合

決定アルゴリズム

決定問題 (P, A) に対するPの決定アルゴリズム

- 各 a $(\in A)$ が性質Pを持つか否かを
  真偽で決定するアルゴリズム

決定アルゴリズム D

- 入力： a を表現する文字列 $d(a)$
- 出力： 真($P(a)$が真の時) ／ 偽($P(a)$が偽の時)
- すなわち $P(a) \Leftrightarrow D(d(a))$

決定可能性 (1/2)

問題 (P, A) は決定可能
⇔ 決定アルゴリズムが存在するとき

- 決定アルゴリズムの存在を示せば良い

問題 (P, A) は決定不能
⇔ 決定アルゴリズムが存在しないとき

- 決定アルゴリズムが存在しないことを証明できるとき

決定可能性 (2/2)

重要な事実: 決定不能な問題が存在する

- 計算できない問題が存在する

TMの能力が弱いからではない

- 考案された様々な計算モデルはTMと同等(先述の通り)

関連する話題 (各自で調べてみて下さい)

- ゲーデルの不完全性定理
- 自己言及のパラドックス
  (e.g., 「クレタ人は嘘つき」だとクレタ人が言った)

決定不能問題

決定不能問題(1) : TMの空入力停止問題 (ε-Halt, A)

A : TMのクラス

ε-Halt(T) :

- TM $T \in A$ の入力を空入力とする
  (入力テープには空白Bだけ)
- Tを動作させたとき
  - Tがいつか停止するなら真
  - Tが停止しなければ偽

TMの空入力停止問題の解釈の例

任意に与えられたプログラムに対して
「無限ループになってるよ」
と確実に無限ループを検出するコンパイラは作れない

注意

- いくつかのプログラムに対しては可能
  - ✔ 例: あらかじめ用意しておいたパターンとマッチング
- すべてのプログラムに対しては不可能ということ

決定不能問題(2) : Postの対応問題 (POST, P_S) (1/2)

P_S : Σ上の語の対の有限集合の集合

- 対応問題の個別具体問題の集合

個別具体問題の例 :

$$\{(ab, abcd), (c, e), (def, fg), (gh, h)\}$$

- このようなもののすべての集合が P_S

決定不能問題(2)：Postの対応問題 (POST, P_S) (2/2)

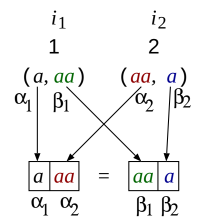
POST(I)：

- $I \in P_S$
- $I = \{(\alpha_1, \beta_1), (\alpha_2, \beta_2), \cdots, (\alpha_n, \beta_n)\}$ に対し
  ある添字列 $i_1 i_2 \cdots i_k \in \{1, \cdots n\}^+$ が存在して
  $$\alpha_{i_1}\alpha_{i_2}\cdots\alpha_{i_k} = \beta_{i_1}\beta_{i_2}\cdots\beta_{i_k}$$
  となればPOST(I)は真
- 存在しなければPOST(I)は偽

Postの対応問題の例 (その1)

$I = \{(a, aa), (aa, a)\}$ の場合

- 添字列は存在：$i_1 i_2 = 12$　— POST(I)は真
- $\boxed{a}\boxed{aa} = \boxed{aa}\boxed{a}$



- $i_1 i_2 = 21$も解 (これら解の繰り返しも解)

Postの対応問題の例 (その2)

$I = \{(a, aa)\}$ の場合

- 添字列は存在しない　— POST(I)は偽
- つなげばつなぐほど長さが食い違って行く
  $\boxed{a} \neq \boxed{aa}$
  $\boxed{a}\boxed{a} \neq \boxed{aa}\boxed{aa}$
  $\boxed{a}\boxed{a}\boxed{a} \neq \boxed{aa}\boxed{aa}\boxed{aa}, \cdots$

TMの停止問題

決定問題の言語の形による表現

決定問題(P, A)は言語の形でも表現可能

$$L_{(P,A)} = \{a \in \Sigma^* \mid a \in A \wedge P(a)\}$$

問題(P, A)が決定可能
⇔ あるTMにより言語$L_{(P,A)}$が判定可能

TMによる言語の判定とは
　任意の入力$a \in A$に対し受理または棄却で停止

- 停止保証：無限ループには入らない

決定不能問題(3)：TMの停止問題 (halting problem)

TMの停止問題
$L_{\mathrm{TM}} = \{\langle M, w\rangle \mid M$はTMであり, Mはwを受理$\}$

- 任意に与えられたTM M
- 任意に与えられた語 w

言語 L_{TM} は判定不可能

- Mがwを受理するか否かを判定するアルゴリズムはない

言語 L_{TM} は認識可能

一方、言語 L_{TM} を認識するTM T は存在

TMによる言語Lの認識とは

- 各入力 $a \in L$ に対し受理して停止
- 各入力 $a \notin L$ に対しては棄却または無限ループ
- 停止保証がない

TM T による L_{TM} の認識: 単にMの動作を模倣

- Tへの入力は$\langle M, w\rangle$
- 模倣したMがwを受理/棄却すればTは受理/棄却し停止
- Mが無限ループの時はTも無限ループ

まとめ: TMによる言語の判定可能性と認識可能性

TMによる言語$L_{(P,A)}$の判定 (チューリング判定)

- 任意の入力 $a \in A$ に対し必ず受理または棄却で停止
- 無限ループには入らない
- 停止の保証がある

TMによる言語$L_{(P,A)}$の認識 (チューリング認識)

- 各入力 $a \in L_{(P,A)}$ に対し受理して停止
- 各入力 $a \notin L_{(P,A)}$ に対しては棄却または無限ループ
- 停止の保証がない

TMの停止問題の決定不可能性 (1/3)

$L_{\mathrm{TM}} = \{\langle M, w\rangle \mid M\text{はTMであり}, M\text{は}w\text{を受理}\}$

- $\langle M, w\rangle$ は M と w の組を文字列で表したもの

背理法で説明

L_{TM} を判定するTM H の存在を仮定して矛盾を示す

$$H(\langle M, w\rangle) = \begin{cases} \text{受理} & M\text{が}w\text{を受理する時} \\ \text{棄却} & M\text{が}w\text{を受理しない時} \\ & (\text{棄却又は無限ループの時}) \end{cases}$$

TMの停止問題の決定不可能性 (2/3)

H を用いて TM D を構成

- D への入力: 任意のTMの文字列表現 $\langle M\rangle$
- D の動作: 内部で H を模倣

1. H への入力: $\langle M, \langle M\rangle\rangle$ — w の代わりに $\langle M\rangle$
2. H の動作を模倣 — H は必ず停止
3. 出力: $D(\langle M\rangle) = \begin{cases} \text{受理} & H\text{が棄却の時} \\ \text{棄却} & H\text{が受理の時} \end{cases}$

すなわち:

$$D(\langle M\rangle) = \begin{cases} \text{受理} & M\text{が}\langle M\rangle\text{を受理しない時} \\ & (\text{棄却又は無限ループの時}) \\ \text{棄却} & M\text{が}\langle M\rangle\text{を受理する時} \end{cases}$$

TMの停止問題の決定不可能性 (3/3)

D にそれ自身の文字列表現 $\langle D\rangle$ を与え動作させる

$$D(\langle D\rangle) = \begin{cases} \text{受理} & D\text{が}\langle D\rangle\text{を受理しない時} \\ & (\text{棄却又は無限ループの時}) \\ \text{棄却} & D\text{が}\langle D\rangle\text{を受理する時} \end{cases}$$

矛盾: 受理しない時に受理し, 受理する時に棄却

結論: L_{TM} を判定するTM H は存在し得ない

【説明おわり】

TMでは認識不能な言語の存在

TMでは認識不能な言語の存在：アイディア (1/2)

TM 全ての集合の濃度は可算

- 可算 ＝ 自然数の集合と同じ濃度(あるいは有限濃度)
- 任意のTM: 二進文字列で記述できる
- 任意の二進文字列: TMの記述とみなせる
  ⇒ TM と自然数は1対1対応
  (各TM に自然数の番号をつけることができる)

言語全ての集合の濃度は非可算

- 言語と自然数の1対1対応は存在しない
  (各言語に自然数の番号をつけることはできない)

TMでは認識不能な言語の存在：アイディア (2/2)

TMの集合と言語の集合の間には1対1対応が存在しない

- TMの数よりも言語の数の方が圧倒的に多い
- 対応するTMがない言語が存在

結論: いかなるTMによっても認識されない言語が存在

ここでのまとめ

帰納的言語

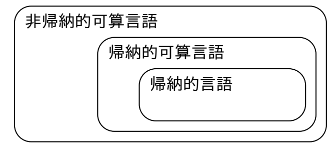
- 判定する TM が存在し停止保証あり (決定可能)

帰納的可算言語

- 認識する TM が存在するが停止保証なし (決定不可能)

非帰納的可算言語

- 認識する TM が存在しない (決定不可能)



詳しくは計算論Ｂで学ぼう!

計算論Aの講義は これでおわり

(･ω･)ﾉｼ

今日のミニレポート (CLE)

(講義で紹介したものテキストに載っているもの以外の)
決定不能問題にどのようなものがあるかを調べ、
どのような問題か簡潔に説明せよ