

チューリングマシン

離散数学・オートマトン

2024 年後期

佐賀大学理工学部 只木進一

- 1 序論: Introduction
- 2 Turing マシン: Turing Machine
- 3 句構造文法: Phase Structure Grammar
- 4 列挙: Enumeration
- 5 停止問題・決定問題: Halting and Decision Problem

更に強力なオートマトンが必要?: Do we need more powerful automata?

- PDA では、 $\{a^n b^n c^n \mid n \in N\}$ を受理できない
 - スタックの制約から
 - 二つのスタックならば可能 → 自由に読み書きできるリストと同等
- 自由に読み書きできる「メモリ」をモデル化したい
- Church-Turing のテーゼ (thesis)
 - 計算できる関数とは、その関数を計算する Turing マシンが存在する関数である。
 - 計算できる、つまりアルゴリズムがあることと、Turing マシンが同等
 - 例外は知られていない

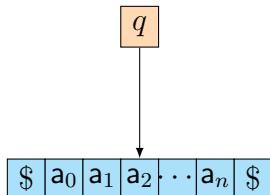
Alan Turing (1912 – 1954)

- もともとは数学者
- 第2次世界大戦中に、暗号解読に従事
- Manchester Mark Iなどの開発に従事
- Turing Test: 人工知能と人を見分ける
- 数理生物学や化学反応にも関心
 - Turing pattern など
- 「イミテーション・ゲーム」

<https://www.britannica.com/biography/Alan-Turing>

Turing マシン

- 読み書きできる左右に無限長のテープ
- \$ は、テープの空白 (何も書いていない) を表す特別な記号
 - テープへは無限に書くことができる
 - \$ は、その外側には何も書いていないことを表す記号
- テープヘッドは左右に動くことができる



両方に無限に長いテープ

$$M = \langle Q, \Gamma, \Sigma, \delta, q_0, \$, F \rangle$$

- Q : 内部状態の有限集合
- Γ : テープ上のアルファベット
- $\Sigma \subseteq \Gamma$: 入力アルファベット
- $\delta : Q \times \Gamma \rightarrow Q \times \Gamma \times \{L, R\}$
 - テープへ書き込み可能: 文字を読んだ場所に文字を上書きする
 - $\{L, R\}$ は、テープヘッドの左右への移動
- $q_0 \in Q$: 初期状態
- $\$ \in \Gamma \setminus \Sigma$: テープ上の空白記号: スペース記号とは異なる
- $F \subseteq Q$: 受理状態の集合

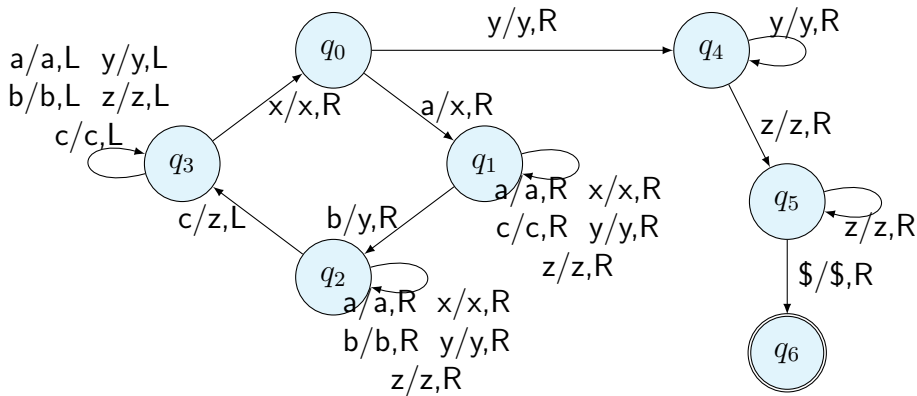
例 2.1: $L = \{a^n b^n c^n \mid n \in N\}$

$$Q = \{q_0, q_1, q_2, q_3, q_4, q_5, q_6\}$$

$$F = \{q_6\}$$

$$\Gamma = \{a, b, c, x, y, z, \$\}$$

$$\Sigma = \{a, b, c\}$$



動作: 状態の右側の文字を読むことに注意

ここまでで、a、b、
c を一つづつ x、y、
z に置き換え

$$q_0 a a b b c c \vdash x q_1 a b b c c \vdash x a q_1 b b c c$$

$$\vdash x a y q_2 b c c \vdash x a y b q_2 c c$$

$$\vdash x a y q_3 b z c \vdash x a q_3 y b z c$$

$$\vdash x q_3 a y b z c \vdash q_3 x a y b z c$$

$$\vdash x q_0 a y b z c$$

$$\vdash \dots \vdash x q_3 x y y z z$$

$$\vdash x x q_0 y y z z \vdash x x y q_4 y z z$$

$$\vdash x x y y q_4 z z \vdash x x y y z q_5 z$$

$$\vdash x x y y z z q_5 \vdash x x y y z z q_6$$

全ての、a、b、c を
x、y、z に置き換え

動作失敗

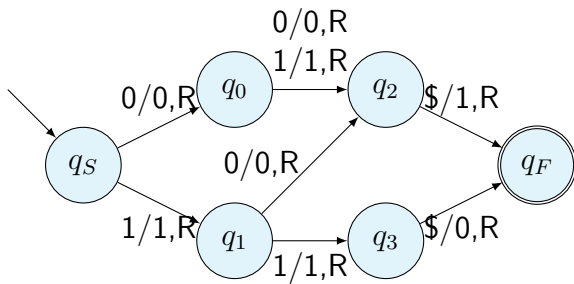
$$\begin{aligned}
 q_0 a a b b c &\vdash x q_1 a b b c \vdash x a q_1 b b c \\
 &\vdash x a y q_2 b c \vdash x a y b q_2 c \\
 &\vdash x a y q_3 b z \vdash x a q_3 y b z \\
 &\vdash x q_3 a y b z \vdash q_3 x a y b z \\
 &\vdash x q_0 a y b z \vdash x x q_1 y b z \\
 &\vdash x x y q_1 b z \vdash x x y y q_2 z \\
 &\vdash x x y y z q_2 \vdash x x y y z q_2
 \end{aligned}$$

入力の受理と関数

- 例 2.1 を少し拡張
- q_6 に到達したら
 - 右端の \$ を 1 に置き換える
 - 全ての文字を \$ で置き換え、左端に 1 と書く
 - 動作失敗したら右端の \$ を 0 で置き換える
 - 全ての文字を \$ で置き換え、左端に 0 と書く
- 正しい入力は判定し、0 または 1 を返す関数に対応した Turing マシン

Turing Machine と DAND ゲート $\overline{a \wedge b}$

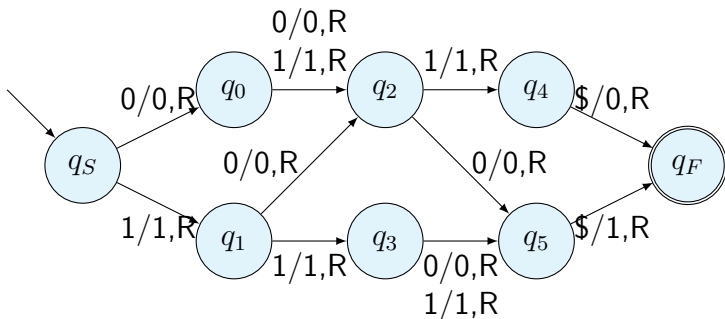
TM が「計算」できること



動作

$$q_S 00 \vdash 0 q_0 0 \vdash 00 q_2 \vdash 001 q_F$$
$$q_S 01 \vdash 0 q_0 1 \vdash 01 q_2 \vdash 011 q_F$$
$$q_S 10 \vdash 1 q_1 0 \vdash 10 q_2 \vdash 101 q_F$$
$$q_S 11 \vdash 1 q_1 1 \vdash 11 q_3 \vdash 110 q_F$$

Turing Machine と DAND ゲート $\overline{(a \wedge b)} \wedge c$



動作

$q_S000 \vdash 0q_000 \vdash 00q_20 \vdash 000q_5 \vdash 0001q_F$
 $q_S001 \vdash 0q_001 \vdash 00q_21 \vdash 001q_4 \vdash 0010q_F$
 $q_S010 \vdash 0q_010 \vdash 01q_21 \vdash 010q_5 \vdash 0101q_F$
 $q_S011 \vdash 0q_011 \vdash 01q_21 \vdash 011q_4 \vdash 0110q_F$
 $q_S100 \vdash 1q_100 \vdash 10q_20 \vdash 100q_5 \vdash 1001q_F$
 $q_S101 \vdash 1q_101 \vdash 10q_21 \vdash 101q_4 \vdash 1010q_F$
 $q_S110 \vdash 1q_110 \vdash 11q_30 \vdash 110q_5 \vdash 1101q_F$
 $q_S111 \vdash 1q_111 \vdash 11q_31 \vdash 111q_5 \vdash 1111q_F$

句構造文法: Phase Structure Grammar

- Turing マシンに対応する文法
- 生成規則

$$P : (N \cup \Sigma)^* N (N \cup \Sigma)^* \rightarrow (N \cup \Sigma)^*$$

- 文脈依存であることに注意
 - 左辺が N を必ず含む N または Σ の列

例 3.1: $L = \{a^n b^n c^n \mid n \in N\}$

$$N = \{S, A, B, C, D\}$$

$$\Sigma = \{a, b, c\}$$

$$\begin{array}{lll} S \rightarrow aAD, & A \rightarrow aAbB, & A \rightarrow C, \\ Bb \rightarrow bB, & Cb \rightarrow bC, & BD \rightarrow Dc, \quad CD \rightarrow bc \end{array}$$

導出例

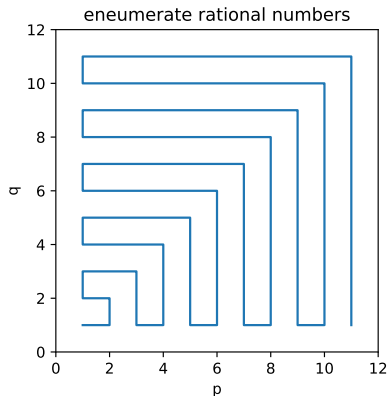
$$S \Rightarrow aAD \Rightarrow aCD \Rightarrow abc$$

$$\begin{aligned} S \Rightarrow aAD &\Rightarrow aaAbBD \Rightarrow aaC'bBD \Rightarrow aaC'bDc \\ &\Rightarrow aabCDc \Rightarrow aabbcc \end{aligned}$$

$$\begin{aligned} S \Rightarrow aAD &\Rightarrow aaAbBD \Rightarrow aaaAbBbBD \\ &\Rightarrow aaaC'bBbBD \Rightarrow aaabCBbBD \\ &\Rightarrow aaabC'bBBD \Rightarrow aaabbCBBD \\ &\Rightarrow aaabbCBDC \Rightarrow aaabbCDcc \Rightarrow aaabbbccc \end{aligned}$$

無限を数える: 正の有理数を列挙する

- 有理数と自然数は同じ数だけ存在する
- 全ての有理数に異なる自然数を対応づけることができる



無限を数える: 無理数は列挙できない

- $x \in [0, 1)$ が列挙できると仮定

$$1 \leftrightarrow 0.a_{11}a_{12}a_{13}a_{14}\cdots$$

$$2 \leftrightarrow 0.a_{21}a_{22}a_{23}a_{24}\cdots$$

$$3 \leftrightarrow 0.a_{31}a_{32}a_{33}a_{34}\cdots$$

$$4 \leftrightarrow 0.a_{41}a_{42}a_{43}a_{44}\cdots$$

- $y = 0.b_1b_2b_3b_4\cdots (b_i \neq a_{ii})$ は列挙したリストに含まれない
 - 列挙できるならば、上記リストに含まれている
- 列挙できるという仮定と矛盾
- 対角線論法 (diagonal method)

Gödel ナンバリング

- Turing マシンを列挙する
 - アルファベット、状態を整数と対応付け
 - 遷移関数は、整数から整数への写像
 - Turing マシンも整数に対応させることができる
- $M = \langle Q, \{0, 1\}, \Gamma, \Sigma, \delta, q_1, \$, F \rangle$
 - $Q = \{q_1, q_2, \dots\}$
 - $\Gamma = \{x_1, x_2, \dots\}$
 - $D = \{L, R\} = \{D_1, D_2\}$
- $\delta(q_i, x_j) = (q_k, x_\ell, D_m)$ に対して

$$0^i 10^j 10^k 10^\ell 10^m$$

万能 Turing マシン

- Turing マシンの動作を模倣する**万能 Turing マシン**が存在できる
 - Turing マシンは符号化できる
 - Turing マシンとその入力を受け取り、その動作を模倣する

Turing マシンの停止問題

- Turing マシンを整理
 - テープ上の入力に対して、結果をテープに残す。
 - テープに残ったものが関数の値。
 - テープ上の入力に対する関数と考える
- 必ず停止するか?
- これから問題とするのは
 - 答えはあるのに、計算で答えを求められない問題の存在
 - 答えを得るためのアルゴリズムがない問題の存在

決定問題: decision problem

- 答えが true/false のいずれかである関数・問題: 述語
- 例: $x^2 + y^2 = z^2$ を満たす自然数の組 (x, y, z) は存在するか?
 - (x, y, z) は列挙可能であるため、順に生成できる
 - $x^2 + y^2 = z^2$ に代入し、等号が成立する場合に、true を返して、停止
 - 例えば、 $(x, y, z) = (3, 4, 5)$ を見つけて停止

決定可能:decidable

- ある Turing マシン M が、決定問題 P の具体例 w に対して、答えを出して停止する
 - 前のシートの例は、停止する
- $x^3 + y^3 = z^3$ を満たす自然数の組 $\{x, y, z\}$ を見つける問題では、前のシートの方法では停止しない
 - Fermat の最終定理: $x^n + y^n = z^n$ ($n > 2$) を満たす整数の組 $\{x, y, z\}$ は存在しない

停止問題: Halting Problem

- 任意の Turing マシン M に対して、入力 w を与えると停止するか
 - これ自体が述語 $f(M, w)$
- Turing マシンの停止問題は決定不能
 - true/false を決定できない
 - false と停止しないことは違うことに注意

- Turing マシンとその入力は列挙できることに注意

$$a_{ij} = \begin{cases} 1 & M_i \text{ は、入力 } x_j \text{ に対して停止} \\ 0 & M_i \text{ は、入力 } x_j \text{ に対して停止しない} \end{cases}$$

- 停止問題を解く Turing マシン \tilde{M} が存在すると仮定
 - 任意の x_i に対して、 $a_{ii} = 0$ のとき、かつそのときだけ停止する Turing マシン M_d に対して、 \tilde{M} が停止を判断できる
 - M_d 自体が、列挙した M_i のいずれか
- M_d を M_j とする
 - $a_{jj} = 1$ ならば、 M_j は停止するが、 M_d は停止しないことになる
 - $a_{jj} = 0$ ならば、 M_j は停止せず、 M_d は停止することになる
- 矛盾するため、 \tilde{M} は存在できない

停止問題が決定不能とは

- 停止問題
 - $f(w)$ の値を決定する
- 停止問題が決定不能
 - 述語 $f(w)$ の真偽を判定できない場合がある
- 正しく設定された述語 $f(w)$ は、真か偽のいずれかである。
 - しかし、判定できない場合がある
- 数学のような厳密な論理体系にあっても、計算によって証明できない命題が存在し得る