チューリングマシン: Turing Machine

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

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

- PDA では、$\{\mathsf{a}^n\mathsf{b}^n\mathsf{c}^n | n \in N\}$ を受理できない
  - スタックの制約から: Based on the restriction of stack
  - 二つのスタックならば可能 $\rightarrow$ 自由に読み書きできるリストと同等: Two stacks enable to handle. However, they are equivalent to a list that can be read and written freely
- 自由に読み書きできる「メモリ」をモデル化: How to model a "memory" that can be read and written freely

Church-Turing thesis

- 計算できる関数とは、: A function is computable if
  - その関数を計算する Turing マシンが存在: there is a Turing machine that computes the function
  - アルゴリズムが存在: there is an algorithm
- 例外は知られていない: No exception is known

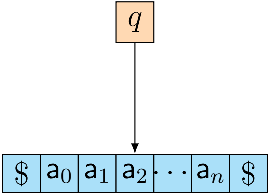
Alan Turing (1912 – 1954)

- イギリスの数学者: British mathematician
  - 第 2 次世界大戦中に、暗号解読に従事: Engaged in code breaking during World War II
  - Manchester Mark I などの開発に従事: Engaged in the development of Manchester Mark I and others
- Turing Test: 人工知能と人を見分ける: Distinguish between artificial intelligence and humans
- 数理生物学や化学反応にも関心: Interested in mathematical biology and chemical reactions
  - Turing pattern, etc.
- 「イミテーション・ゲーム」: “The Imitation Game”

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

Turing machine

- 読み書きできる左右に無限長のテープ: Infinite tape on left and right that can be read and written
- $ は、その外側には何も書いていないことを表す記号: $ represents that nothing is written outside
- テープヘッドは左右に動くことができる: The tape head can move left and right



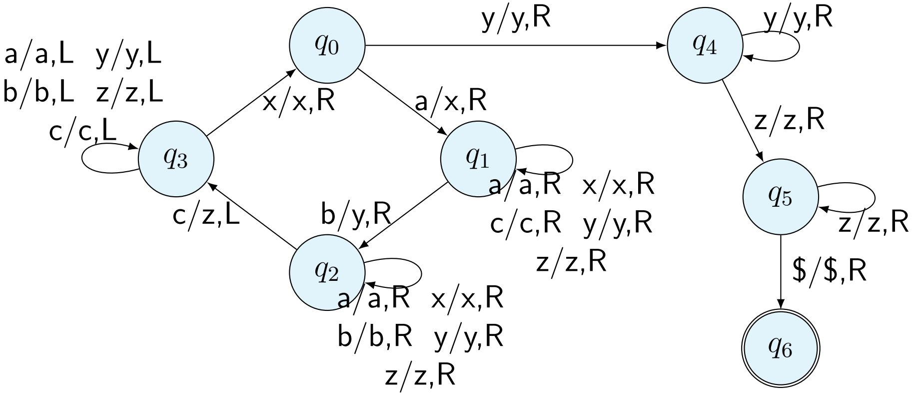
両方に無限に長いテープ

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

- Q: **内部状態の有限集合**: Finite set of internal states
- Γ: **テープ上のアルファベット**: Alphabet on the tape
- $\Sigma \subset \Gamma$: **入力アルファベット**: Input alphabet
- $\delta : Q \times \Gamma \to Q \times \Gamma \times \{\mathsf{L}, \mathsf{R}\}$
  - **文字を読んだ場所に文字を上書き**: Overwrite a character at the current location
  - $\{\mathsf{L}, \mathsf{R}\}$: **テープヘッドの左右への移動**: Move the tape head left or right
- $q_0 \in Q$: **初期状態**: Initial state
- $\$ \in \Gamma \setminus \Sigma$: **その外側には何も書いていないことを表す記号**: $\$$ represents that nothing is written outside
- $F \subseteq Q$: **受理状態の集合**: Set of accepting states

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

$$
\begin{aligned}
Q &= \{q_0, q_1, q_2, q_3, q_4, q_5, q_6\} \\
F &= \{q_6\} \\
\Gamma &= \{\mathsf{a}, \mathsf{b}, \mathsf{c}, \mathsf{x}, \mathsf{y}, \mathsf{z}, \$\} \\
\Sigma &= \{\mathsf{a}, \mathsf{b}, \mathsf{c}\}
\end{aligned}
$$


q_0
q_1
q_2
q_3
q_4
q_5
q_6
y/y,R
a/x,R
x/x,R
b/y,R
c/z,L
a/a,L y/y,L
b/b,L z/z,L
c/c,L
a/a,R x/x,R
c/c,R y/y,R
z/z,R
a/a,R x/x,R
b/b,R y/y,R
z/z,R
y/y,R
z/z,R
z/z,R
$/$,R

動作: Motion of the Turing Machine

$$
\begin{aligned}
q_0\text{aabbcc} &\vdash \text{x}q_1\text{abbcc} \vdash \text{xa}q_1\text{bbcc} \\
&\vdash \text{xay}q_2\text{bcc} \vdash \text{xayb}q_2\text{cc} \\
&\vdash \text{xay}q_3\text{bzc} \vdash \text{xa}q_3\text{ybzc} \\
&\vdash \text{x}q_3\text{aybzc} \vdash q_3\text{xaybzc} \\
&\vdash \text{x}q_0\text{aybzc} \\
&\vdash \cdots \vdash \text{x}q_3\text{xyyzz} \\
&\vdash \text{xx}q_0\text{yyzz} \vdash \text{xxy}q_4\text{yzz} \\
&\vdash \text{xxyy}q_4\text{zz} \vdash \text{xxyyz}q_5\text{z} \\
&\vdash \text{xxyyzz}q_5 \vdash \text{xxyyzz}q_6
\end{aligned}
$$

ここまでで、a、b、c を一つづつ x、y、z に置き換え

全ての、a、b、c を x、y、z に置き換え

動作失敗: Failure

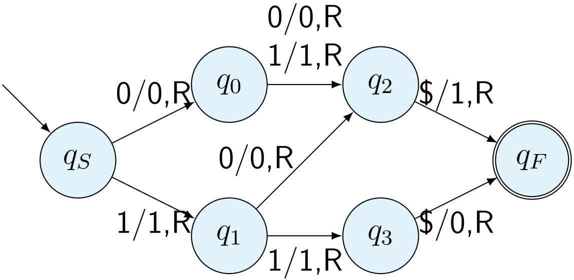
$$
\begin{aligned}
q_0\mathsf{aabbc} &\vdash \mathsf{x}q_1\mathsf{abbc} \vdash \mathsf{xa}q_1\mathsf{bbc} \\
&\vdash \mathsf{xay}q_2\mathsf{bc} \vdash \mathsf{xayb}q_2\mathsf{c} \\
&\vdash \mathsf{xay}q_3\mathsf{bz} \vdash \mathsf{xa}q_3\mathsf{ybz} \\
&\vdash \mathsf{x}q_3\mathsf{aybz} \vdash q_3\mathsf{xaybz} \\
&\vdash \mathsf{x}q_0\mathsf{aybz} \vdash \mathsf{xx}q_1\mathsf{ybz} \\
&\vdash \mathsf{xxy}q_1\mathsf{bz} \vdash \mathsf{xxyy}q_2\mathsf{z} \\
&\vdash \mathsf{xxyyz}q_2 \vdash \mathsf{xxyyz}q_2
\end{aligned}
$$

入力の受理を示す関数: Function to indicate acceptance of input

- **入力を受理したら、テープ終端に受理を表す特殊な文字を書き込む**: Write a special character at the end of the tape to indicate acceptance
- **入力を受理しなかったら、テープ終端に失敗を表す特殊な文字を書き込む**: Write a special character at the end of the tape to indicate failure
- **入力を判定し、真偽を返す関数に対応した** Turing **マシン**: Turing machine calculating a function that returns True or False for input

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

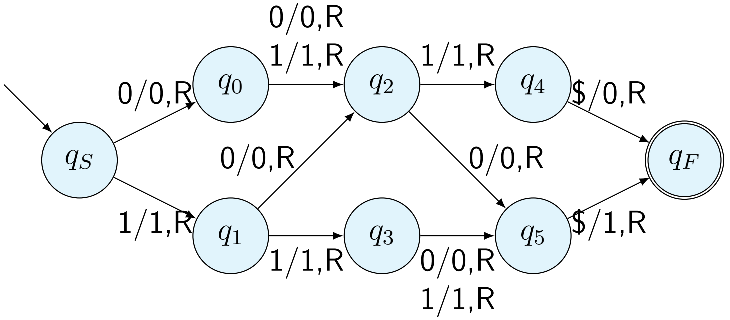
TM が「計算」できること: Turing machine can “compute”



動作: Operation

$$
\begin{aligned}
q_S00 &\vdash 0q_00 \vdash 00q_2 \vdash 001q_F \\
q_S01 &\vdash 0q_01 \vdash 01q_2 \vdash 011q_F \\
q_S10 &\vdash 1q_10 \vdash 10q_2 \vdash 101q_F \\
q_S11 &\vdash 1q_11 \vdash 11q_3 \vdash 110q_F
\end{aligned}
$$

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



動作: Operation

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

句構造文法: Phase Structure Grammars

- Turing マシンに対応する文法: Grammar corresponding to Turing machine
- 生成規則: Productions

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

- 文脈依存: *context-dependent*
  - 左辺が N を必ず含む N または Σ の列: LHS is a sequence of N or Σ that must contain N

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

$$
\begin{aligned}
N &= \{S, A, B, C, D\} \\
\Sigma &= \{\mathsf{a}, \mathsf{b}, \mathsf{c}\}
\end{aligned}
$$

$$
\begin{array}{llll}
S \to \mathsf{a}AD, & A \to \mathsf{a}A\mathsf{b}B, & A \to C, & \\
B\mathsf{b} \to \mathsf{b}B, & C\mathsf{b} \to \mathsf{b}C, & BD \to D\mathsf{c}, & CD \to \mathsf{bc}
\end{array}
$$

導出例: Derivation example

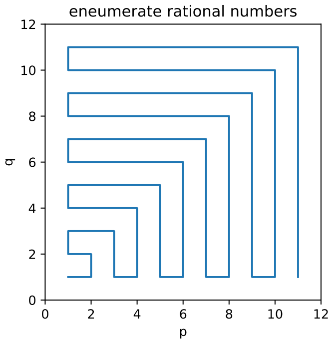
$$S \Rightarrow \mathsf{a}AD \Rightarrow \mathsf{a}CD \Rightarrow \mathsf{abc}$$

$$S \Rightarrow \mathsf{a}AD \Rightarrow \mathsf{aa}A\mathsf{b}BD \Rightarrow \mathsf{aa}C\mathsf{b}BD \Rightarrow \mathsf{aa}C\mathsf{b}D\mathsf{c}$$
$$\Rightarrow \mathsf{aab}CD\mathsf{c} \Rightarrow \mathsf{aabbcc}$$

$$S \Rightarrow \mathsf{a}AD \Rightarrow \mathsf{aa}A\mathsf{b}BD \Rightarrow \mathsf{aaa}A\mathsf{b}B\mathsf{b}BD$$
$$\Rightarrow \mathsf{aaa}C\mathsf{b}B\mathsf{b}BD \Rightarrow \mathsf{aaab}CB\mathsf{b}BD$$
$$\Rightarrow \mathsf{aaab}C\mathsf{b}BBD \Rightarrow \mathsf{aaabb}CBBD$$
$$\Rightarrow \mathsf{aaabb}CBD\mathsf{c} \Rightarrow \mathsf{aaabb}CD\mathsf{cc} \Rightarrow \mathsf{aaabbbccc}$$

正の有理数を列挙する: Enumerate positive rational numbers

- 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: Different natural numbers can be associated with all rational numbers



無理数は列挙できない: Irrational numbers cannot be enumerated

- $x \in [0, 1)$ **が列挙できると仮定**: Assume that $x \in [0, 1)$ can be enumerated

$$
\begin{aligned}
1 &\leftrightarrow 0.a_{11}a_{12}a_{13}a_{14}\cdots \\
2 &\leftrightarrow 0.a_{21}a_{22}a_{23}a_{24}\cdots \\
3 &\leftrightarrow 0.a_{31}a_{32}a_{33}a_{34}\cdots \\
4 &\leftrightarrow 0.a_{41}a_{42}a_{43}a_{44}\cdots
\end{aligned}
$$

- $y = 0.b_1 b_2 b_3 b_4 \cdots (b_i \neq a_{ii})$ は列挙したリストに含まれない: y is not included in the enumerated list
  - 列挙できるならば、上記リストに含まれている: If irrational numbers can be enumerated, y is included in the list
- 列挙できるという仮定と矛盾: Contradiction with the assumption that irrational numbers can be enumerated
- 対角線論法 (diagonal method)

Gödel numbering

- Turing マシンを列挙する: Enumerate Turing machines
  - アルファベット、状態を整数と対応付け: Mapping alphabet and states to integers
  - 遷移関数は、整数から整数への写像: Transition function is a mapping from integers to integers
- $M = \langle Q, \{0,1\}, \Gamma, \Sigma, \delta q_1, \$, F \rangle$
  - $Q = \{q_1, q_2, \cdots\}$
  - $\Gamma = \{x_1, x_2, \cdots\}$
  - $D = \{\mathsf{L}, \mathsf{R}\} = \{D_1, D_2\}$
- $\delta(q_i, x_j) = (q_k, x_\ell, D_m)$ に対して

$$0^i 1 0^j 1 0^k 1 0^\ell 1 0^m$$

万能 Turing マシン: Universal Turing Machine

- 任意の Turing マシンの動作を模倣する万能 Turing マシンが存在できる: There is a universal Turing machine that mimics the operation of any Turing machine
  - Turing マシンは符号化できる: Turing machine can be encoded
  - 万能 Turing マシンは、符号化された Turing マシンとその入力を受け取り、その動作を模倣する: Universal Turing machine takes an encoded Turing machine and its input and mimics its operation

Turing マシンの停止問題: Halting Problem

- **必ず停止するか**?: Does the Turing machine always stop?
- **答えはあるのに、計算で答えを求められない問題の存在**: Problems for which the answer exists but cannot be obtained by calculation

決定問題: decision problem

- **述語**: **答えが** true/false **のいずれかである関数**: Predicate is a function that is either true or false
- **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**?: Does there exist a set of natural numbers (x, y, z) that satisfy $x^2 + y^2 = z^2$?
  - (x, y, z) **は列挙可能**: (x, y, z) can be enumerated
  - $x^2 + y^2 = z^2$ **に代入し、等号が成立する場合に、**true **を返して、停止**: Substitute into $x^2 + y^2 = z^2$ and return true and stop if the equality holds
  - Example $(x, y, z) = (3, 4, 5)$

停止問題: Halting Problem

- Turing マシン M に対して、入力 w を与えると停止するか?: Does a Turing machine M stop for given input w?

$$f : (M, w) \to \{\mathsf{true}, \mathsf{false}\}$$

- Turing マシンの停止問題は決定不能: The halting problem of Turing machines is undecidable
  - true/false を決定できない: Cannot determine true/false
  - false と停止しないことは違うことに注意: Note that false and not stopping are different

- Turing マシンとその入力は列挙できる: pairs of Turing machines and their inputs are enumerable

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

- 停止問題を解く Turing マシン $\tilde{M}$ が存在すると仮定: Assume that there is a Turing machine $\tilde{M}$ that solves the halting problem
  - (M_d, x)
    - 停止 (stop):$a_{ii} = 0$
    - 停止しない (not stop):$a_{ii} = 1$
  - $\tilde{M}$ が停止を判断できる: $\tilde{M}$ can determine whether M_d stops or not

- M_d 自体が、列挙した M_i のいずれか: M_d itself is one of the enumerated M_i
- M_d を M_j とする: Let $M_d = M_j$
    - $a_{jj} = 1$ ならば、M_j は停止するが、M_d は停止しないことになる: If $a_{jj} = 1$, M_j stops, but M_d does not stop
    - $a_{jj} = 0$ ならば、M_j は停止せず、M_d は停止することになる: If $a_{jj} = 0$, M_j does not stop, but M_d stops
- 矛盾するため、$\tilde{M}$ は存在できない: Contradiction! $\tilde{M}$ cannot exist

停止問題が決定不能とは: Halting problem is undecidable

- 停止問題: Halting problem
  - $f(w)$ の値を決定する: Determine the value of $f(w)$
- 停止問題が決定不能: Halting problem is undecidable
  - 述語 $f(w)$ の真偽を判定できない場合がある: There are cases where the truth of the predicate $f(w)$ cannot be determined
- 正しく設定された述語 $f(w)$ は、真か偽のいずれかである。: A properly set predicate $f(w)$ is either true or false.
  - しかし、判定できない場合がある: However, there are cases where it cannot be determined
- 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: Even in a rigorous logical system like mathematics, there may be propositions that cannot be proven by calculation