決定不能問題の話

y. (@waidotto)

http://iso.2022.jp/

2017 年 9 月 16 日 @ 第 10 回関西すうがく徒のつどい

導入

突然ですが問題です:

素数判定問題

正整数がひとつ与えられたとき，それが素数であるかどうかを判定するにはどうすればよいか.

導入

突然ですが問題です:

素数判定問題

正整数がひとつ与えられたとき，それが素数であるかどうかを判定するにはどうすればよいか．

解答例

与えられた整数を n とおく．$n = 1$ なら素数ではないし，$n = 2$ なら素数である．$n > 2$ なら，$2, 3, \ldots, n-1$ で順番に割ってみて，どれかひとつでも割り切れるものがあれば素数ではない．どれでも割り切れなければ素数である．

観察

先程の問題は「20170916 は素数か？」といった問題とは趣が異なっている

- 具体的な個々の正整数が素数かどうかを問うているのではなく，「すべての正整数について，素数かどうかを判定できるひとつの “方法”(アルゴリズム) を与えよ」という問題
- この立場で見れば，20170916 は無数にある「入力」のひとつにすぎない

入力	1	2	3	4	5	6	···	20170916	···
答え	No	Yes	Yes	No	Yes	No	···	No	···

この表の全体が「問題」

決定問題の定義

定義 (決定問題)

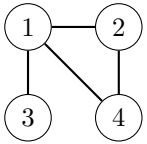
与えられた文字列に対して，YES または NO で答える問題を決定問題(decision problem) という．

- 今回は入力を文字列に限る
- 文字の集合 Σ (有限集合) をあらかじめ決めておく (Σ をアルファベット(alphabet) という)
- 文字列の全体 (Σ の元の有限列の全体) を Σ^* で表す

決定問題の定義

定義 (決定問題)

与えられた文字列に対して，YES または NO で答える問題を決定問題(decision problem) という．

- 今回は入力を文字列に限る
- 文字の集合 Σ (有限集合) をあらかじめ決めておく (Σ をアルファベット(alphabet) という)
- 文字列の全体 (Σ の元の有限列の全体) を Σ^* で表す
- 自然数やグラフなど，計算の対象となるものは文字列で表すことができるので，これは本質的な制約ではない
  - すべての自然数は 10 進法で表記できる ($\Sigma = \{0,1,2,3,4,5,6,7,8,9\}$)
  - グラフ 

は {{1,2,3,4},{{1,2},{1,3},{1,4},{2,4}}} と表せる

決定問題を解くということ

- 「決定問題を解く」ということは，全ての入力に対して正しい答えを返すようなアルゴリズムを見つけることに他ならない
- そのようなアルゴリズムが存在しない決定問題を決定不能問題(undecidable problem) という
  - これが今回のテーマ

注意

「$e+\pi$ は超越数か？」という問題は (未解決だが) 決定可能である

- これは決定問題としては「入力が $e+\pi$ の一通りしかない」問題
- $e+\pi$ は代数的数であるか超越数であるかのどちらかなので，「YES を返すプログラム」と「NO を返すプログラム」のいずれかは正しい答えを返す
入力	$e+\pi$
答え	YES

    か

入力	$e+\pi$
答え	NO

    のどちらかが必ず正しい
- どちらが正しいかは現時点ではわからないが，正しい答えを返すアルゴリズムが「存在する」ことは確か

注意

「$e+\pi$ は超越数か？」という問題は (未解決だが) 決定可能である

- これは決定問題としては「入力が $e+\pi$ の一通りしかない」問題
- $e+\pi$ は代数的数であるか超越数であるかのどちらかなので，「YES を返すプログラム」と「NO を返すプログラム」のいずれかは正しい答えを返す
  - $\begin{array}{c|c} \text{入力} & e+\pi \\ \hline \text{答え} & \text{YES} \end{array}$ か $\begin{array}{c|c} \text{入力} & e+\pi \\ \hline \text{答え} & \text{NO} \end{array}$ のどちらかが必ず正しい
- どちらが正しいかは現時点ではわからないが，正しい答えを返すアルゴリズムが「存在する」ことは確か

同様に，入力が有限通りしかない問題は必ず決定可能なので考察する意味がない

- 決定問題を考えるときは「入力が無限通りある」ことが重要！

定義したはいいものの……

決定不能問題を定義はしたものの，その存在には触れなかった

疑問 (その 1)

「決定不能問題」は存在するか？

定義したはいいものの……

決定不能問題を定義はしたものの，その存在には触れなかった

疑問 (その 1)

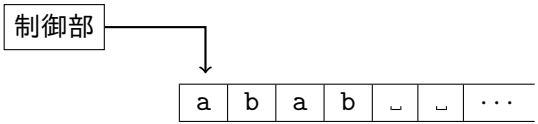
「決定不能問題」は存在するか？

- 決定問題のうち，それを解くアルゴリズムが存在しないようなものを探せばよい，のだが……
- この問題に答えるためには，「アルゴリズム」の数学的な定義が不可欠

Turing 機械

Turing 機械(Turing machine) は，1936 年に A. Turing により提案された計算モデルのひとつ

- 無限長のテープ(tape) があり，ひとつのマスにはひとつの文字が書き込まれている
- 機械のヘッド(head) はテープのひとつのマスを見ており，左右に動くことができる
- 機械は一定の規則に基づいて，ヘッドが見ている文字を書き換えたりヘッドを左右に動かしたりして計算を進める



Turing 機械の正式な定義

定義

Turing 機械とは，7 個組 $(Q, \Sigma, \Gamma, \delta, q_0, q_{\text{accept}}, q_{\text{reject}})$ である．ここで，Q, Σ, Γ は全て有限集合であり，

- Q は状態(state) の集合
- Σ は入力アルファベットで空白記号を含まない: $\textvisiblespace \notin \Sigma$
- Γ はテープアルファベットで $\textvisiblespace \in \Gamma \supseteq \Sigma$
- $\delta\colon Q \times \Gamma \to Q \times \Gamma \times \{\mathrm{L}, \mathrm{R}\}$ は遷移関数(transition function)
- $q_0 \in Q$ は開始状態
- $q_{\text{accept}} \in Q$ は受理状態
- $q_{\text{reject}} \in Q$ は拒否状態で $q_{\text{reject}} \neq q_{\text{accept}}$

Turing 機械の計算方法

Turing 機械 M への入力文字列を $w = w_1 w_2 \cdots w_n \in \Sigma^*$ としたとき，計算は次のように進む:

1. テープの左端から n マス目まで $w_1, w_2, \ldots, w_n$ が書き込まれ，$n+1$ マス目以降は全て空白文字␣が書き込まれる．
2. M の内部状態を初期状態 q_0 とする．
3. M の現在の状態が q で，現在ヘッドが見ている文字が a で，$\delta(q, a) = (q', b, \mathrm{R})$ ならば M の次の状態を q' にし，ヘッドが見ているマスを b に書き換え，ヘッドをひとつ右に動かす (L なら左に動かす)．
4. 状態が q_{accept} か q_{reject} になるまで繰り返し，q_{accept} になったら直ちに計算を終了して受理する．q_{reject} になったら拒否する．

- 状態が q_{accept} か q_{reject} になるときは，最後の状態によって $M(w) =$ 受理 または $M(w) =$ 拒否
- いつまで経っても状態が $q_{\mathsf{accept}}, q_{\mathsf{reject}}$ にならない (無限ループしている) 場合には，計算結果は未定義

Turing 機械の例

入力アルファベットを $\Sigma = \{\mathsf{a}, \mathsf{b}\}$ として，以下の問題を解く Turing 機械を作る:

問題

入力: 文字列 $w \in \Sigma^*$

質問: w は $\underbrace{\mathsf{a}\cdots\mathsf{a}}_{n}\underbrace{\mathsf{b}\cdots\mathsf{b}}_{n}(n \geq 0)$ という形をしているか？

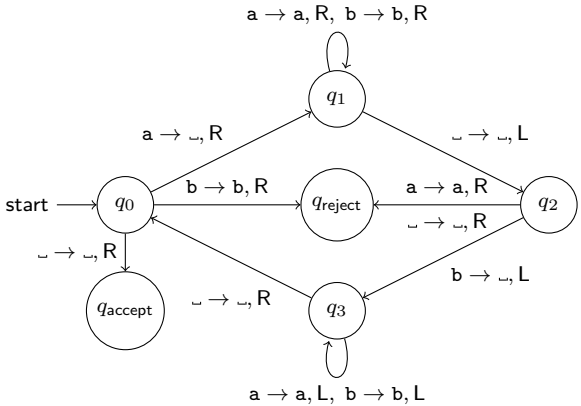
例

$Q = \{q_0, q_1, q_2, q_3, q_{\text{accept}}, q_{\text{reject}}\}, \Gamma = \{\mathsf{a}, \mathsf{b}, \sqcup\}$ とし，δ を以下のように定めればよい:

$$
\begin{array}{lll}
\delta(q_0, \mathsf{a}) = (q_1, \sqcup, \mathsf{R}), & \delta(q_0, \mathsf{b}) = (q_{\text{reject}}, \mathsf{b}, \mathsf{R}), & \delta(q_0, \sqcup) = (q_{\text{accept}}, \sqcup, \mathsf{R}), \quad \text{(左端の a を消す)} \\
\delta(q_1, \mathsf{a}) = (q_1, \mathsf{a}, \mathsf{R}), & \delta(q_1, \mathsf{b}) = (q_1, \mathsf{b}, \mathsf{R}), & \delta(q_1, \sqcup) = (q_2, \sqcup, \mathsf{L}), \quad \text{(右端へ行く)} \\
\delta(q_2, \mathsf{a}) = (q_{\text{reject}}, \mathsf{a}, \mathsf{R}), & \delta(q_2, \mathsf{b}) = (q_3, \sqcup, \mathsf{L}), & \delta(q_2, \sqcup) = (q_{\text{reject}}, \sqcup, \mathsf{R}), \quad \text{(右端の b を消す)} \\
\delta(q_3, \mathsf{a}) = (q_3, \mathsf{a}, \mathsf{L}), & \delta(q_3, \mathsf{b}) = (q_3, \mathsf{b}, \mathsf{L}), & \delta(q_3, \sqcup) = (q_0, \sqcup, \mathsf{R}). \quad \text{(左端へ行く)}
\end{array}
$$

Turing 機械の例

状態遷移図を描くと次のようになる:



(動作はシミュレータで実演します)

Turing 機械の例

- 計算の進行を計算状況(configuration) という文字列を並べた計算履歴(computation history) として表すことができる
- 例えば入力文字列が $\texttt{aabb} \in \Sigma^*$ のとき,

$$
\begin{array}{ll}
q_0\texttt{aabb} & q_3\texttt{␣ab␣␣} \\
\texttt{␣}q_1\texttt{abb} & \texttt{␣}q_0\texttt{ab␣␣} \\
\texttt{␣a}q_1\texttt{bb} & \texttt{␣␣}q_1\texttt{b␣␣} \\
\texttt{␣ab}q_1\texttt{b} & \texttt{␣␣b}q_1\texttt{␣␣} \\
\texttt{␣abb}q_1\texttt{␣} & \texttt{␣␣}q_2\texttt{b␣␣} \\
\texttt{␣ab}q_2\texttt{b␣} & \texttt{␣}q_3\texttt{␣␣␣␣} \\
\texttt{␣a}q_3\texttt{b␣␣} & \texttt{␣␣}q_0\texttt{␣␣␣} \\
\texttt{␣}q_3\texttt{ab␣␣} & \texttt{␣␣␣}q_{\text{accept}}\texttt{␣␣}.
\end{array}
$$

Turing 機械の計算能力

- 状態 Q やテープアルファベット Γ の数を増やしていくことで，Turing 機械は様々な計算をすることができるようになる
- 慣れてくると，Turing 機械には次のような計算ができるはずだ，ということが信じられるようになる
  - 四則演算
  - Turing 機械 M の記述 $\langle M \rangle$ (遷移関数を文字列で表現したもの) をもとに，M の動作をシミュレートする

Church-Turing の提唱

Church-Turing の提唱

$f\colon \Sigma^* \to \{\mathrm{YES}, \mathrm{NO}\}$ が計算可能 (決定可能) $\iff$ f を計算する Turing 機械が存在する

- 右辺は数学的にはっきりした主張だが，左辺はそうではない
- 左辺を右辺で定義しよう，ということ

Church-Turing の提唱の傍証:

- 他のやり方で定義された様々な計算モデルと計算能力が等価であることが証明されている
  - ラムダ計算，再帰関数，レジスター機械，etc.

疑問その 1 への回答

疑問 (その 1，再掲)

「決定不能問題」は存在するか？

疑問その 1 への回答

疑問 (その 1，再掲)

「決定不能問題」は存在するか？

回答

存在する．

証明.

決定問題全体の集合 $\{\mathrm{YES}, \mathrm{NO}\}^{\Sigma^*}$ の濃度は $\mathbb{R}$ の濃度と等しく，非可算無限である．一方で，Turing 機械の遷移関数は可算通りしかないので，Church-Turing の提唱より計算可能な関数の全体も高々可算である．したがって決定不能問題が (非可算個) 存在する． □

ちょっと待った！

よくよく考えると，我々が「文章で具体的に記述可能な」決定問題も可算通りしかない

- 次の自然な疑問が生まれる: 「我々が文章で具体的に記述できる決定問題は全て決定可能なのではないか？」

ちょっと待った！

よくよく考えると，我々が「文章で具体的に記述可能な」決定問題も可算通りしかない

- 次の自然な疑問が生まれる: 「我々が文章で具体的に記述できる決定問題は全て決定可能なのではないか？」

疑問 (その 2)

“具体的な” 決定不能問題は存在するか？

ちょっと待った！

よくよく考えると，我々が「文章で具体的に記述可能な」決定問題も可算通りしかない

- 次の自然な疑問が生まれる: 「我々が文章で具体的に記述できる決定問題は全て決定可能なのではないか？」

疑問 (その 2)

“具体的な” 決定不能問題は存在するか？

回答

存在する.

停止問題

問題 (停止問題 (halting problem; $HALT$))

入力: Turing 機械 M と入力文字列 w の組 (を 1 つの文字列で表したもの)$\langle M, w \rangle$
質問: M は入力 w を受理するか？

定理

$HALT$ は決定不能である.

$HALT$ の決定不能性の証明

証明.

仮に $HALT$ を解く Turing 機械 H が存在したとすると，次のような Turing 機械 D を作ることができる:

$$D(\langle M \rangle) = \begin{cases} \text{拒否} & (H(\langle M, \langle M \rangle \rangle) = \text{受理}) \\ \text{受理} & (H(\langle M, \langle M \rangle \rangle) = \text{拒否}) \end{cases}$$

このとき，

$$D(\langle D \rangle) = \text{受理} \iff H(\langle D, \langle D \rangle \rangle) = \text{拒否} \iff D(\langle D \rangle) = \text{拒否 or 停止しない}$$

となり矛盾. □

対角線論法

- 先程の証明は実は対角線論法になっている
- 全ての Turing 機械を一列に並べてみると……

Table: $H(\langle M_i, \langle M_j \rangle\rangle)$ の出力

	$\langle M_1 \rangle$	$\langle M_2 \rangle$	$\langle M_3 \rangle$	$\langle M_4 \rangle$	$\cdots$	$\langle D \rangle$	$\cdots$
M_1	受理	拒否	受理	拒否	$\cdots$	受理	$\cdots$
M_2	受理	受理	受理	受理	$\cdots$	受理	$\cdots$
M_3	拒否	拒否	拒否	拒否	$\cdots$	拒否	$\cdots$
M_4	受理	受理	拒否	拒否	$\cdots$	受理	$\cdots$
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\ddots$	$\vdots$	
D	拒否	拒否	受理	受理	$\cdots$	?	$\cdots$
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$		$\vdots$	$\ddots$

決定不能問題はあった，でも……

- $HALT$ はとても人工的で作為的な印象を受ける
- (計算機科学の人を除いて) 停止問題そのものに大して興味をそそられない

決定不能問題はあった，でも……

- $HALT$ はとても人工的で作為的な印象を受ける
- (計算機科学の人を除いて) 停止問題そのものに大して興味をそそられない

疑問 (その 3)

より “自然な” 決定不能問題は存在するか？

決定不能問題はあった，でも……

- $HALT$ はとても人工的で作為的な印象を受ける
- (計算機科学の人を除いて) 停止問題そのものに大して興味をそそられない

疑問 (その 3)

より “自然な” 決定不能問題は存在するか？

回答

存在する.

Post の対応問題

定義

文字列の組 (u, v) を縦に並べた $\left[\frac{u}{v}\right]$ をドミノとよぶ．ドミノの有限列

$$\left[\frac{u_1}{v_1}\right]\left[\frac{u_2}{v_2}\right]\cdots\left[\frac{u_n}{v_n}\right]$$

がマッチであるとは，上下の文字列を繋げた文字列がそれぞれ等しい
$(u_1u_2\cdots u_n = v_1v_2\cdots v_n)$ ことをいう．

問題 (Post の対応問題 (Post correspondence problem; PCP))

入力: ドミノの有限集合 P
質問: P はマッチを持つか？ (ただし，同じドミノを何回使ってもよいとする)

Post の対応問題: 入力例

例

$$P_1 = \left\{ \left[\frac{\mathtt{wah}}{\mathtt{wa}}\right], \left[\frac{\mathtt{ey}}{\mathtt{h}}\right], \left[\frac{\mathtt{h}}{\mathtt{ey}}\right], \left[\frac{\mathtt{d}}{\mathtt{eyd}}\right] \right\}$$

は次のマッチを持つ:

$$\left[\frac{\mathtt{wah}}{\mathtt{wa}}\right]\left[\frac{\mathtt{ey}}{\mathtt{h}}\right]\left[\frac{\mathtt{h}}{\mathtt{ey}}\right]\left[\frac{\mathtt{ey}}{\mathtt{h}}\right]\left[\frac{\mathtt{d}}{\mathtt{eyd}}\right].$$

一方,

$$P_2 = \left\{ \left[\frac{\mathtt{alg}}{\mathtt{al}}\right], \left[\frac{\mathtt{dal}}{\mathtt{g}}\right], \left[\frac{\mathtt{gd}}{\mathtt{d}}\right] \right\}$$

は (上の文字列の方が下の文字列より常に長いので) マッチを持たない.

PCP は決定不能

定理

PCP は決定不能である.

PCP は決定不能

定理

PCP は決定不能である.

証明のアイディア:

- 「もし PCP を解くアルゴリズムが存在したら，それを利用して $HALT$ を解くアルゴリズムが作れてしまう」ことを示す
- そのために，$HALT$ の入力 $\langle M, w\rangle$ が与えられたら，

$$M(w) = \text{受理} \iff P_{\langle M,w\rangle} \text{ はマッチを持つ}$$

となるような $P_{\langle M,w\rangle}$ を作る

PCP は決定不能

定理

PCP は決定不能である.

証明のアイディア:

- 「もし PCP を解くアルゴリズムが存在したら，それを利用して $HALT$ を解くアルゴリズムが作れてしまう」ことを示す
- そのために，$HALT$ の入力 $\langle M, w\rangle$ が与えられたら，

$$M(w) = \text{受理} \iff P_{\langle M,w\rangle} \text{ はマッチを持つ}$$

となるような $P_{\langle M,w\rangle}$ を作る

- つまり，“Turing 機械の動作をドミノで模倣する”！
- このようなテクニックを**Turing** 還元(Turing reduction) または**Turing** 帰着とよぶ

PCP の決定不能性の証明

技術的な理由により，2 段階にわけて示す

問題 (modified PCP; $MPCP$)

入力: ドミノの有限集合 P と $d \in P$
質問: P は左端が d であるようなマッチを持つか？

- PCP が解けたとすると $MPCP$ が解け，最終的に $HALT$ も解けてしまうことを示す:

$$\underbrace{PCP \longrightarrow MPCP}_{\text{まずこっちを示す}} \longrightarrow HALT.$$

PCP の決定不能性の証明

PCP が解けたら $MPCP$ が解けることの証明.

$MPCP$ の入力を $P = \left\{ d = \left[\frac{u_1}{v_1}\right], \left[\frac{u_2}{v_2}\right], \ldots, \left[\frac{u_n}{v_n}\right] \right\}$ とし，$*, \Diamond$ を P に現れない文字とする．
一般に文字列 $w = w_1 \cdots w_l$ に対し，

$$
\begin{aligned}
*w &:= *w_1 * w_2 * \cdots * w_l \\
w* &:= w_1 * w_2 * \cdots * w_l * \\
w &:= *w_1 * w_2 * \cdots * w_l *
\end{aligned}
$$

と定義し，PCP の入力を $P' := \left\{ \left[\frac{*u_1}{*v_1*}\right], \left[\frac{*u_1}{v_1*}\right], \left[\frac{*u_2}{v_2*}\right], \ldots, \left[\frac{*u_n}{v_n*}\right], \left[\frac{*\Diamond}{\Diamond}\right] \right\}$ と定める．このとき明らかに

$$P \text{ が左端が } d \text{ であるマッチを持つ} \iff P' \text{ がマッチを持つ.}$$

□

PCP の決定不能性の証明

- 次に，$MPCP$ が解けたら $HALT$ が解けることを示す:

$$PCP \underset{\text{済}}{\longrightarrow} \underbrace{MPCP \longrightarrow HALT}_{\text{今度はこっちを示す}}.$$

- $HALT$ の入力を $\langle M, w\rangle$ $(M = (Q, \Sigma, \Gamma, \delta, q_0, q_{\text{accept}}, q_{\text{reject}}), w = w_1 \cdots w_n)$ とする
- $MPCP$ の入力 $P = P_{\langle M,w\rangle}$ を，マッチが $M(w)$ の受理計算履歴となるように作っていく

PCP の決定不能性の証明

1. まず，計算の開始状況を表すドミノ $d = \left[\frac{\#}{\#q_0 w_1 \cdots w_n \#}\right]$ を P に追加する (#は Γ には含まれない区切り記号)
   - 以後，上段を揃えようとしたら勝手に下段に次の計算状況が現れるように P を作っていく
2. 各 $a, b \in \Gamma, q, r \in Q\,(r \neq q_{\mathsf{reject}})$ に対し $\delta(q, a) = (r, b, \mathsf{R})$ のとき $\left[\frac{qa}{br}\right]$ を P に追加する
3. 各 $a, b, c \in \Gamma, q, r \in Q\,(r \neq q_{\mathsf{reject}})$ に対し $\delta(q, a) = (r, b, \mathsf{L})$ のとき $\left[\frac{cqa}{rcb}\right]$ を P に追加する
4. 各 $a \in \Gamma$ に対して $\left[\frac{a}{a}\right]$ を追加する
5. $\left[\frac{\#}{\#}\right], \left[\frac{\#}{\sqcup\#}\right]$ を P に追加する

PCP の決定不能性の証明

6. 各 $a \in \Gamma$ に対して $\left[\frac{aq_{\mathsf{accept}}}{q_{\mathsf{accept}}}\right], \left[\frac{q_{\mathsf{accept}}a}{q_{\mathsf{accept}}}\right]$ を P に追加する
7. $\left[\frac{q_{\mathsf{accept}}\#\#}{\#}\right]$ を P に追加する

例

はじめの Turing 機械の例で，入力が ab のとき

PCP の決定不能性の証明

6. 各 $a \in \Gamma$ に対して $\left[\frac{aq_{\text{accept}}}{q_{\text{accept}}}\right], \left[\frac{q_{\text{accept}}a}{q_{\text{accept}}}\right]$ を P に追加する
7. $\left[\frac{q_{\text{accept}}\#\#}{\#}\right]$ を P に追加する

例

はじめの Turing 機械の例で，入力が ab のとき

$$P = \left\{ d = \left[\frac{\#}{\#q_0\mathtt{ab}\#}\right], \left[\frac{q_0\mathtt{a}}{\sqcup q_1}\right], \left[\frac{q_0\sqcup}{\sqcup q_{\text{accept}}}\right], \left[\frac{q_1\mathtt{a}}{\mathtt{a}q_1}\right], \left[\frac{q_1\mathtt{b}}{\mathtt{b}q_1}\right], \left[\frac{q_3\sqcup}{\sqcup q_0}\right], \left[\frac{\mathtt{a}q_1\sqcup}{q_2\mathtt{a}\sqcup}\right], \left[\frac{\mathtt{b}q_1\sqcup}{q_2\mathtt{b}\sqcup}\right], \left[\frac{\sqcup q_1\sqcup}{q_2\sqcup\sqcup}\right], \right.$$
$$\left[\frac{\mathtt{a}q_2\mathtt{b}}{q_3\mathtt{a}\sqcup}\right], \left[\frac{\mathtt{b}q_2\mathtt{b}}{q_3\mathtt{b}\sqcup}\right], \left[\frac{\sqcup q_2\mathtt{b}}{q_3\sqcup\sqcup}\right], \left[\frac{\mathtt{a}q_3\mathtt{a}}{q_3\mathtt{a}\mathtt{a}}\right], \left[\frac{\mathtt{b}q_3\mathtt{a}}{q_3\mathtt{b}\mathtt{a}}\right], \left[\frac{\sqcup q_3\mathtt{a}}{q_3\sqcup\mathtt{a}}\right], \left[\frac{\mathtt{a}q_3\mathtt{b}}{q_3\mathtt{a}\mathtt{b}}\right], \left[\frac{\mathtt{b}q_3\mathtt{b}}{q_3\mathtt{b}\mathtt{b}}\right], \left[\frac{\sqcup q_3\mathtt{b}}{q_3\sqcup\mathtt{b}}\right], \left[\frac{\mathtt{a}}{\mathtt{a}}\right], \left[\frac{\mathtt{b}}{\mathtt{b}}\right], \left[\frac{\sqcup}{\sqcup}\right],$$
$$\left.\left[\frac{\#}{\#}\right], \left[\frac{\#}{\sqcup\#}\right], \left[\frac{\mathtt{a}q_{\text{accept}}}{q_{\text{accept}}}\right], \left[\frac{\mathtt{b}q_{\text{accept}}}{q_{\text{accept}}}\right], \left[\frac{\sqcup q_{\text{accept}}}{q_{\text{accept}}}\right], \left[\frac{q_{\text{accept}}\mathtt{a}}{q_{\text{accept}}}\right], \left[\frac{q_{\text{accept}}\mathtt{b}}{q_{\text{accept}}}\right], \left[\frac{q_{\text{accept}}\sqcup}{q_{\text{accept}}}\right], \left[\frac{q_{\text{accept}}\#\#}{\#}\right] \right\}.$$

PCP の決定不能性の証明

- これにて証明終了！
  - $\langle M, w \rangle$ から P を作る手順は Turing 機械で実行可能
- 証明のポイントと注意:
  - q_{reject} を含むドミノを入れなかったことに気を付ける
    $\rightarrow$ 拒否計算履歴は作れない！
  - 今の証明では文字に $\Gamma \cup \{\#\}$ を使ったが，実際には $\{0, 1\}$ で十分
    $\rightarrow$ 実際，$\Gamma \cup \{\#\} = \{a_1, \ldots, a_n\}$ としたとき，a_i を $1\underbrace{0 \cdots 0}_{i}1$ で置き換えればよい

その他の問題

問題 (Matrix Mortality Problem; $\mathrm{MORT}_{\mathbb{Z}}(n)$)

入力: 整数成分の $n \times n$ 行列の有限集合 $F \subseteq \mathrm{M}_n(\mathbb{Z})$
質問: F が生成する乗法半群 $\langle F \rangle$ は零行列を含むか？ (F の元の積で零行列が作れるか？)

その他の問題

問題 (Matrix Mortality Problem; $\mathrm{MORT}_{\mathbb{Z}}(n)$)

入力: 整数成分の $n \times n$ 行列の有限集合 $F \subseteq \mathrm{M}_n(\mathbb{Z})$
質問: F が生成する乗法半群 $\langle F \rangle$ は零行列を含むか？ (F の元の積で零行列が作れるか？)

定理

$\mathrm{MORT}_{\mathbb{Z}}(n)$ は $n \geq 3$ で決定不能．(Peterson, 1970)

証明の概略

- $\mathrm{MORT}_{\mathbb{Z}}(3)$ が解けたとすると，PCP も解けてしまうことを示す．
- ドミノの結合を行列の積で再現する！ (ただし，使える文字は $\{2,3\}$ のみとする)

$$W(23,2)W(223,32) = \begin{pmatrix} 100 & 0 & 0 \\ 0 & 10 & 0 \\ 23 & 2 & 1 \end{pmatrix} \begin{pmatrix} 1000 & 0 & 0 \\ 0 & 100 & 0 \\ 223 & 32 & 1 \end{pmatrix} = \begin{pmatrix} 100000 & 0 & 0 \\ 0 & 1000 & 0 \\ 23223 & 232 & 1 \end{pmatrix}$$
$$= W(23223,232).$$

あとは

$$S = \begin{pmatrix} 1 & 0 & 1 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix}, T = \begin{pmatrix} 1 & -1 & 0 \\ -1 & 1 & 0 \\ 0 & 0 & 0 \end{pmatrix}$$

とおけば，$u = v$ のときのみ $SW(u,v)T = 0$ となる

ちなみに

Matrix Mortality Problem は $n = 2$ のときの決定可能性は未解決

- 挑戦しよう！

その他の問題

問題 (Matrix Identity Problem)

入力: 整数成分の $n \times n$ 行列の有限集合 $F \subseteq \mathrm{M}_n(\mathbb{Z})$
質問: F が生成する乗法半群 $\langle F \rangle$ は単位行列を含むか？ (F の元の積で単位行列が作れるか？)

その他の問題

問題 (Matrix Identity Problem)

入力: 整数成分の $n \times n$ 行列の有限集合 $F \subseteq \mathrm{M}_n(\mathbb{Z})$
質問: F が生成する乗法半群 $\langle F \rangle$ は単位行列を含むか？ (F の元の積で単位行列が作れるか？)

事実

Matrix Identity Problem は $n \geq 4$ で決定不能 (Bell & Potapov, 2009)，$n \leq 2$ で決定可能. (Potapov & Semukhin, 2016)

その他の問題

問題 (Matrix Identity Problem)

入力: 整数成分の $n \times n$ 行列の有限集合 $F \subseteq \mathrm{M}_n(\mathbb{Z})$
質問: F が生成する乗法半群 $\langle F \rangle$ は単位行列を含むか？ (F の元の積で単位行列が作れるか？)

事実

Matrix Identity Problem は $n \geq 4$ で決定不能 (Bell & Potapov, 2009)，$n \leq 2$ で決定可能. (Potapov & Semukhin, 2016)

- $n = 3$ は未解決なので挑戦しよう！

その他の問題 (紹介のみ)

問題 (Hilbert の第 10 問題)

入力: 整数係数多項式 $f \in \bigcup_{n>0} \mathbb{Z}[x_1, \ldots, x_n]$
質問: f は整数根を持つか？

$\rightarrow$ 決定不能 (Davis, Putnam, Robinson, Matiyasevich)

問題

入力: 算術の閉論理式 φ ($0, 1, +, \times, <, =, \wedge, \vee, \neg, \rightarrow, \forall, \exists, x, y, z, \ldots$ で作られるもの)
質問: φ は $\mathbb{N}$ で正しい ($\mathbb{N} \models \varphi$) か？

$\rightarrow$ 決定不能 (Gödel の第一不完全性定理，または上の Hilbert の第 10 問題による)

その他の問題 (紹介のみ)

問題

入力: 群の有限表示 $G = \langle g_1, \ldots, g_n \mid (g_1, \ldots, g_n \text{ の関係式}) \rangle$
質問: G は自明な群 ($G \cong \{1\}$) か？

$\rightarrow$ 決定不能 (有限群か，アーベル群か，などもすべて決定不能)

問題

入力: 2 つの (抽象) 単体複体 M, N
質問: M と N は同相か？

$\rightarrow$ 決定不能

その他の問題 (紹介のみ)

事実

次のようなゲームが存在する:

- 先手 A と後手 B が交互に自然数を出し合うゲーム
- 3 手 (x_1, x_2, x_3) で終わる
- 盤面から勝者を計算する関数 $W\colon \mathbb{N}^3 \to \{A, B\}$ は計算可能
- B に必勝法がある
- B の必勝戦略 $x_2 = g(x_1)$ は計算不能！

問題 (Polyomino tiling)

入力: ポリオミノ P (有限個の単位正方形を辺で貼り合わせた図形)
質問: P の無限個のコピーで平面を埋め尽くせるか？

$\to$ 未解決！

まとめ

- 決定問題というのがある
- 決定不能問題は存在する (濃度の比較から)
- “具体的な” 決定不能問題が存在する (停止問題 $HALT$)
- “自然な” 決定不能問題が存在する (Post の対応問題 PCP)
- 世の中は決定不能問題に溢れている！
- 挑戦しよう:
  - Matrix Mortality Problem ($n = 2$)
  - Matrix Identity Problem ($n = 3$)
  - Polyomino tiling
  - etc.

参考文献

- M. Sipser (太田和夫・田中圭介 監訳, 阿部正幸・植田広樹・藤岡淳・渡辺治 訳), 計算理論の基礎 [原著第 2 版] 2. 計算可能性の理論, 共立出版, 2008.
- B. Poonen, *Undecidable problems: a sampler*, Preprint, arXiv:1204.0299v2, 2012.
- 河村彰星, はじめての計算可能性, 数学基礎論サマースクール 2017, http://toshio-suzuki-logic.jp/meeting/summer2017.html.
- M. S. Peterson, Unsolvability in 3×3 Matrices, *Studies in Applied Mathematics*, Vol. 49, 1970. pp. 105–107.
- P. C. Bell, I. Potapov, *The Identity Correspondence Problem and its Applications*, Preprint, arXiv:0902.1975v3, 2009.
- I. Potapov, P. Semukhin, *Decidability of the Membership Problem for* 2×2 *integer matrices*, Preprint, arXiv:1604.02303v1, 2016.