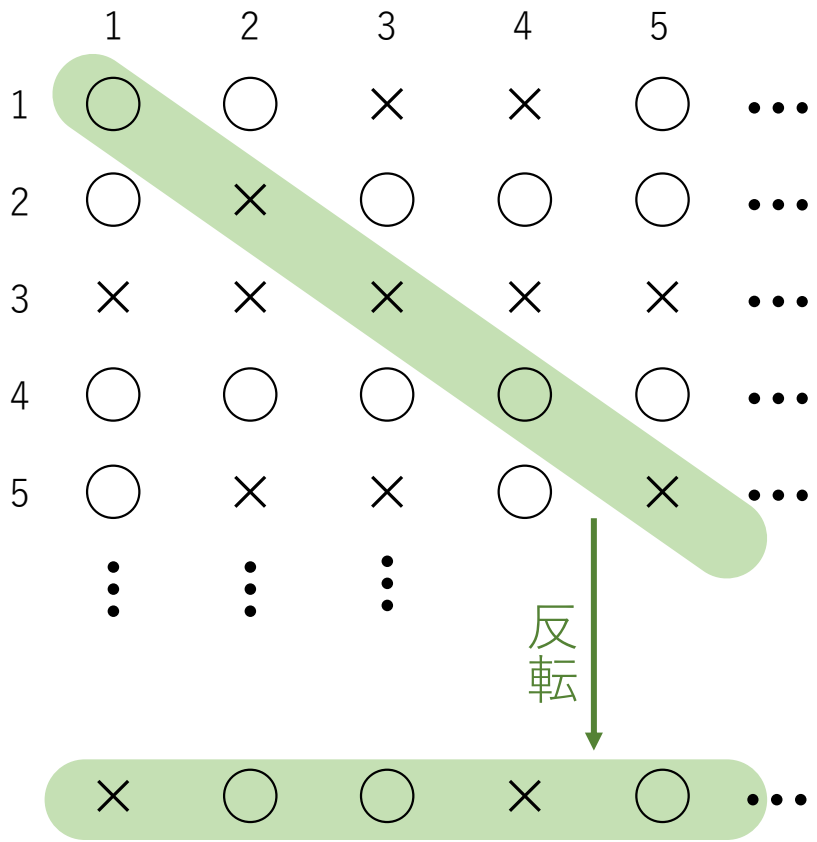
計算の理論

令和5年5月9日

河村彰星（京大）

※本日のみ担当します



番号１２３…を
つけて無限に

左図のように
○と×を横に並べたものを**○×行**と呼び
○×行を縦に並べたものを**○×表**と呼ぶ

どちらが正しいでしょうか？

~~うまく○×表を作ると
あらゆる○×行が現れるようにできる~~

如何なる○×表に対しても
それに現れない○×行が存在する

∵　左図のように　対角線上の内容と
喰い違うように定義すればよい

問題と機械

問題とは

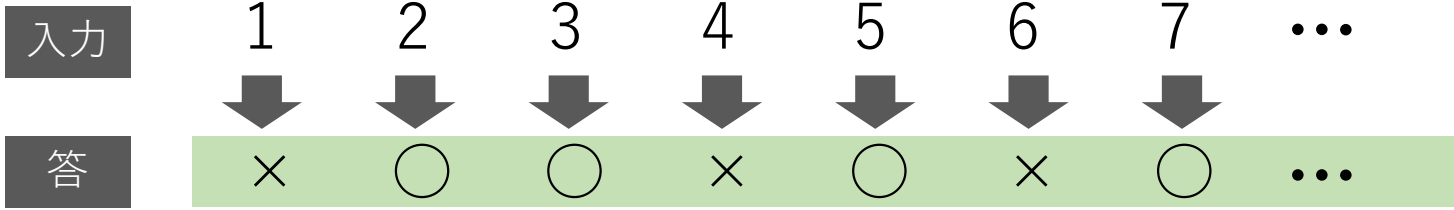
ここでは 各入力に対して　答が○か×か定めたもの（入力は文字列で表される） と考えることにする

例えば

問題 PRIME

0 1 … 9 からなる文字列

与えられた正整数（十進法で書く）が素数か答えよ



問題 ABA

a と b からなる

与えられた文字列に「aba」が現れるか答えよ

問題とは

ここでは 各入力に対して　答が○か×か定めたもの（入力は文字列で表される） と考えることにする

例えば

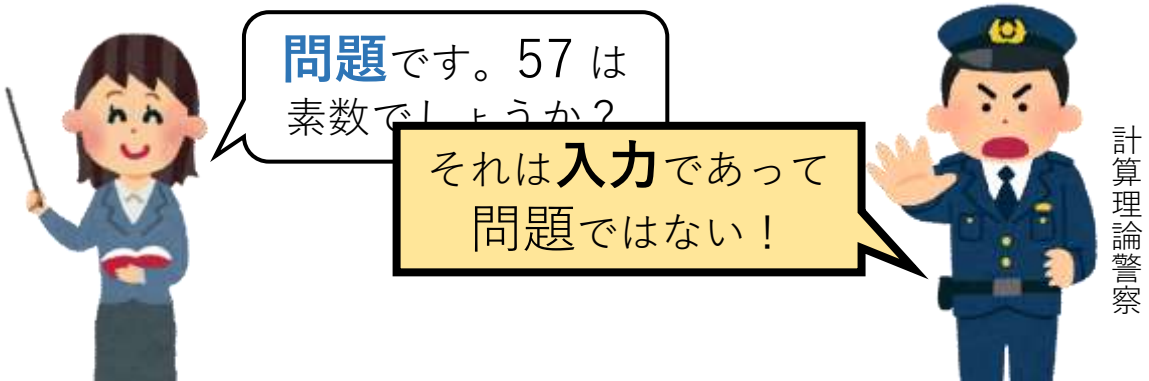
問題 PRIME

0 1 … 9 からなる文字列

与えられた正整数（十進法で書く）が素数か答えよ

入力	1	2	3	4	5	6	7	…
答	×	○	○	×	○	×	○	…

問題とは
コレ全体のこと！



問題 SR

入力　書換え規則の（有限）集合 R と文字列 $w \in \Sigma^*$

R の各規則は $u \to v$ という形（$u, v \in \Sigma^*$）

文字列の一部を u から v に書換えることができるという意味

つまり $xuy \Rightarrow_R xvy$ とできる

答　R による書換えを次々と w に施して ε（空文字列）にできるか

$w \Rightarrow_R^* \varepsilon$ と書くことにする

例 1

入力

$$R \begin{cases} \text{aa} \to \text{bbb} \\ \text{aba} \to \text{a} \\ \text{bb} \to \varepsilon \end{cases}$$

$w = \text{aababab}$

答　○（受理）

（aababab $\Rightarrow_R$ aabab $\Rightarrow_R$ aab $\Rightarrow_R$ bbbb $\Rightarrow_R$ bb $\Rightarrow_R \varepsilon$ とできるので）

例 2

入力

$$R \begin{cases} \text{aa} \to \text{bab} \\ \text{aba} \to \text{a} \\ \text{bb} \to \varepsilon \end{cases}$$

$w = \text{aababab}$

答　×（拒否）

（a を消せる規則がないので）

定義

有限個の文字からなる集合 Σ を考える
Σ に属する文字を有限個並べて得られる文字列の全体を Σ^* と書く
Σ^* の部分集合を**言語**あるいは**問題**という

この講義では　与えられた文字列が言語に属するか問う問題のみを考える

例　文字集合 $\Sigma = \{0, 1, 2, 3, 4, 5, 6, 7, 8, 9\}$ を考える（今後は一々断らず適当に定める）

Σ 上の文字列（36 とか 7 とか ε（空文字列） とか 119 とか）の全体を Σ^* で表す

文字列のうち十進法で素数を表すもの全体が
言語 PRIME $= \{2, 3, 5, 7, 11, 13, \ldots\} \subseteq \Sigma^*$ である

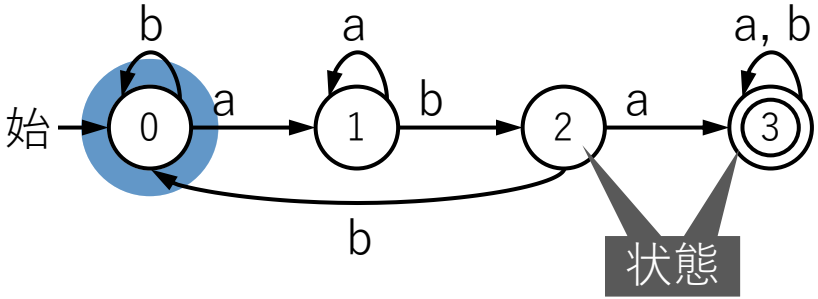
この講義では「文字列を入力すると　それが PRIME に属するか
答えてくれる計算機械」などについて考えたい

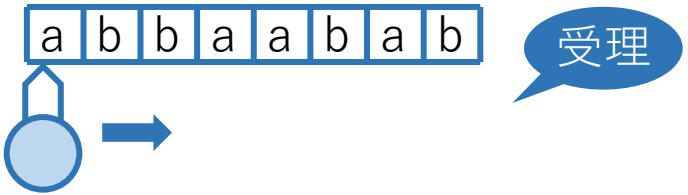
この「入力された文字列が言語 PRIME に属するか判断せよ」という問題
のことも PRIME と呼ぶ（言語と問題は同じものと考える）ことにする

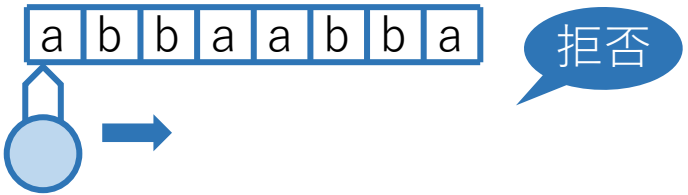
問題 ABA

与えられた文字列に「aba」が現れるか答えよ

この問題を解く有限状態機械







状態は有限個
矢印も有限本

➡ 有限の記述で書き表せる

次にどの状態に行くか表す遷移規則 δ

$\delta(0,\mathrm{a}) = 1$　　$\delta(0,\mathrm{b}) = 0$

$\delta(1,\mathrm{a}) = 1$　　$\delta(1,\mathrm{b}) = 2$

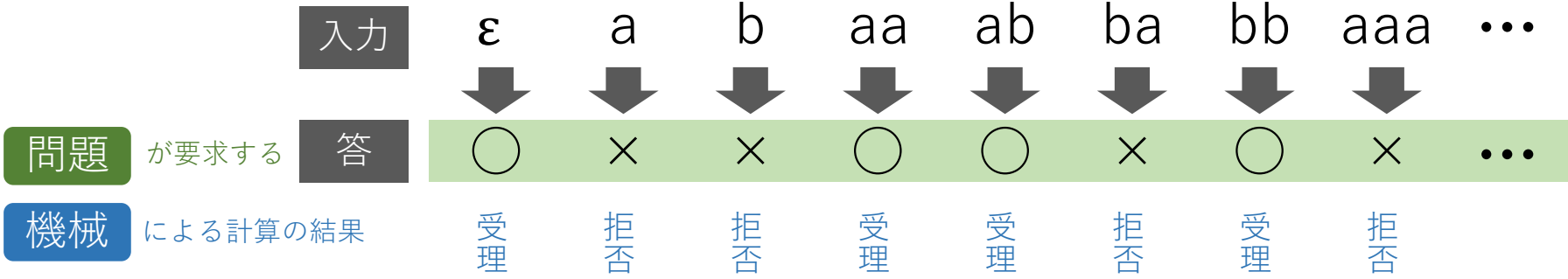
$\delta(2,\mathrm{a}) = 3$　　$\delta(2,\mathrm{b}) = 0$

$\delta(3,\mathrm{a}) = 3$　　$\delta(3,\mathrm{b}) = 3$

定義

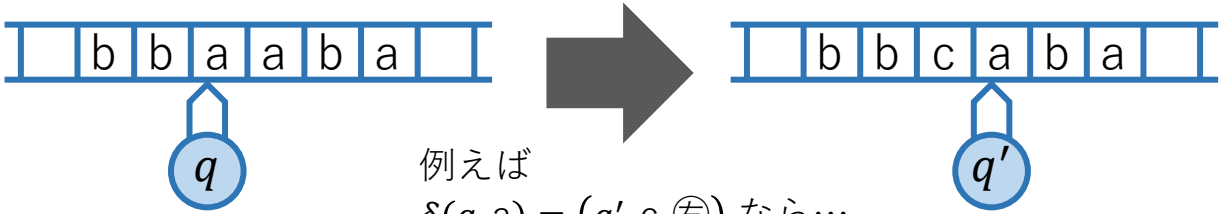
機械 M が言語 A を**判定**するとは　任意の入力 x に対し
- $x \in A$ のとき M は x を受理し　かつ
- $x \notin A$ のとき M は x を拒否する

すべての入力で
正しく判断！



左から右に読み進めるだけでは ABA のように簡単な問題しか解けないので

読むばかりでなく
書く機能と
左右に動く機能
を与える（次頁）



機械が文字列を受理／拒否するとは

（前スライドまでの理解で良いが一応細かく定義すると）

定義

チューリング機械（以下単に機械と呼ぶ）は次のものにより指定される

- 有限個の**状態**の集合 Q　但し次のものが定まっている
  - 始状態 $q_{始} \in Q$
  - 受理状態の集合 $Q_{受理} \subseteq Q$
- 有限個の**文字**の集合 Σ　これと空白文字 ␣ を含む集合 $\Gamma \supseteq \Sigma \cup \{␣\}$
- **遷移規則** $\delta: Q \times \Gamma \to (Q \times \Gamma \times \{㊧, ㊨\}) \cup \{\textbf{止}\}$

初め機械は始状態 $q_{始}$ にあり　テープ上に与えられた文字列 $x \in \Sigma^*$ の左端から始めて　次のこと（遷移）を繰返す

　状態 $q \in Q$ で文字 σ を読むと　$\delta(q, \sigma)$ が

- **止** ならば停止する
- (q', σ', d) なら　状態を q' にし σ' を書込み d の向きに一歩進む

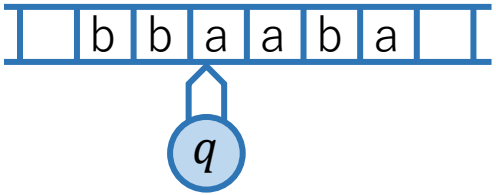
$Q_{受理}$ に属する状態で停止したら機械は x を**受理**したという

［属しない］　［**拒否**］

※停止せず永遠に動き続ける場合は　受理も拒否もしていないと考える
（そうなってしまう x があるような機械は　言語を判定できていないとする）

- 停止
  - 受理
  - 拒否
- 不停止

更に厳密に言うと……

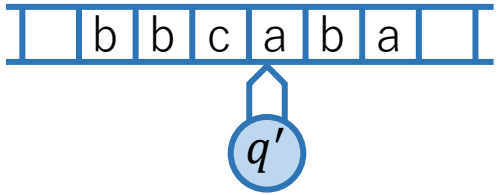


この状況を　bb$q$aaba
のように表すことにする

M

状況の遷移

例えば $\delta(q, \mathrm{a}) = (q', \mathrm{c}, ㊨)$ なら…



次の状況　bbcq'aba

Γ^* の文字列の途中に Q の元がちょうど 1 回だけ現れる文字列を**状況**と呼び（但し先頭や末尾に ␣ を加えた文字列も同じ状況とみなす）状況の遷移 $\xrightarrow{M}$ を次で定義する

状態 $q \in Q$ と文字 $\sigma \in \Gamma$ に対し

- もし $\delta(q, \sigma) = (q', \sigma', ㊧)$ ならば　任意の $u, v \in \Gamma^*$ と $\tau \in \Gamma$ に対し $u\tau q\sigma v \xrightarrow{M} uq'\tau\sigma' v$
- もし $\delta(q, \sigma) = (q', \sigma', ㊨)$ ならば　任意の $u, v \in \Gamma^*$ に対し $uq\sigma v \xrightarrow{M} u\sigma' q' v$

文字列 $x \in \Sigma^*$ を機械 M が**受理**するとは

状況の有限列 $(s_0, \ldots, s_f)$ であって次を満すものが存在することをいう

- $s_0 = q_{始} x$
- $s_0 \xrightarrow{M} s_1 \xrightarrow{M} s_2 \xrightarrow{M} \cdots \xrightarrow{M} s_f$
- s_f には $\delta(q, \sigma) =$ **止** かつ $q \in Q_{受理}$ なる $q\sigma$ が現れる

チャーチとチューリングの定立

機械的な計算手順で判定できる ＝ （チューリング機械で）判定可能

ボンヤリした概念

数学的にハッキリ定義した概念

チューリング機械ですべての「計算」が実現できているなんて本当？

幾つかの理由から　今では広く受入れられています

- 現実の計算に出て来そうな色々な手順は確かにチューリング機械で書けるようだ（経験上）
- 他の様々なやり方で定義された計算可能性の概念とも一致

チューリングの論文（次頁）

A. M. Turing. On computable numbers, with an application to the Entscheidungsproblem. *Proceedings of the London Mathematical Society* 2, **42**: 230–265, 1936.

1. [Type (*a*)]. This argument is only an elaboration of the ideas of § 1.

Computing is normally done by writing certain symbols on paper. We may suppose this paper is divided into squares like a child's arithmetic book. In elementary arithmetic the two-dimensional character of the paper is sometimes used. But such a use is always avoidable, and I think that it will be agreed that the two-dimensional character of paper is no essential of computation. I assume then that the computation is carried out on one-dimensional paper, *i.e.* on a tape divided into squares. I shall also suppose that the number of symbols which may be printed is finite. If we were to allow an infinity of symbols, then there would be symbols differing to an arbitrarily small extent†. The effect of this restriction of the number of symbols is not very serious. It is always possible to use sequences of symbols in the place of single symbols. Thus an Arabic numeral such as

紙に文字を書きながら計算する様子を考えよう

文字は有限個と考えてよかろう

無限個使っても区別できなきゃ意味ないので

† If we regard a symbol as literally printed on a square we may suppose that the square is $0 \leqslant x \leqslant 1$, $0 \leqslant y \leqslant 1$. The symbol is defined as a set of points in this square, viz. the set occupied by printer's ink. If these sets are restricted to be measurable, we can define the "distance" between two symbols as the cost of transforming one symbol into the other if the cost of moving unit area of printer's ink unit distance is unity, and there is an infinite supply of ink at $x = 2$, $y = 0$. With this topology the symbols form a conditionally compact space.

17 or 999999999999999 is normally treated as a single symbol. Similarly in any European language words are treated as single symbols (Chinese, however, attempts to have an enumerable infinity of symbols). The differences from our point of view between the single and compound symbols is that the compound symbols, if they are too lengthy, cannot be observed at one glance. This is in accordance with experience. We cannot tell at a glance whether 9999999999999999 and 999999999999999 are the same.

The behaviour of the computer at any moment is determined by the symbols which he is observing, and his "state of mind" at that moment. We may suppose that there is a bound B to the number of symbols or squares which the computer can observe at one moment. If he wishes to observe more, he must use successive observations. We will also suppose that the number of states of mind which need be taken into account is finite. The reasons for this are of the same character as those which restrict the number of symbols. If we admitted an infinity of states of mind, some of them will be "arbitrarily close" and will be confused. Again, the restriction is not one which seriously affects computation, since the use of more complicated states of mind can be avoided by writing more symbols on the tape.

大きな数値を記号と考えてもやはり瞬時には区別できない

現在の文字と現在の状態から次の動きを決める

状態も有限個と考えてよかろう

一度に読み書きできる文字数も

解けない問題（判定不能な言語）

機械は文字列で表せる

万能機械

プログラマ

チューリング機械は（有限の）文字列で記述できる

遷移規則 $\delta(q,\sigma)=(q',\sigma',d)$ が有限個あるだけ

その文字列（算譜（プログラム））を機械と呼ぶと思ってもよい

ε
0
1
00
01
10
11
000
001
010
011
100
101
110
111
0000
0001
0010
0011
0100
0101

すべての機械はこのどこかに現れる

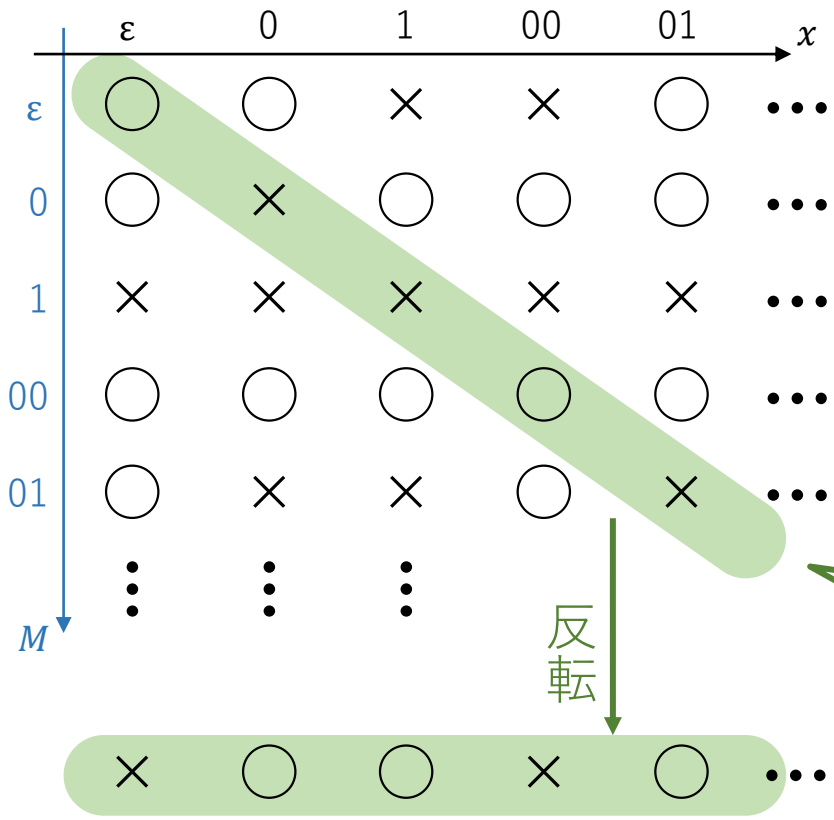
このお蔭で　各機械を実際に建造せずとも
算譜を読込むことで同じ機能を実現できる

なぜなら　先程やったように
機械の遷移規則を見ながらそれを実行することは
単純な機械的作業である（ので
チャーチとチューリングの定立により　同じことを機械でできる）

しかし　機械が有限の文字列で表せることから

判定可能でない言語

を構成することもできてしまう



対角線論法

定理

どんな○×表に対しても
それに現れない○×行が存在する

∵　左図のように　対角線上の内容と
喰い違うように定義すればよい

各行を機械の動作と考えると……

機械 M が入力 x に対し停止するか否かを
第 (M,x) 成分に○×で記した表にこの定理を適用

与えられた文字列 x に対し
これの第 x 成分が○か×か問う問題

を判定する機械は存在しない

∵　もし存在したら　それをもとに
「第 x 成分が×なら停止し
○なら停止しない」機械が作れてしまう

今の議論より　次の問題 HALT が判定不能と示された

問題 HALT

入力　二つの文字列の組 (M, x)

答　機械 M は入力 x で停止するか

判定不能

対角線論法で作った問題

入力　文字列 x

答　機械 x は入力 x で停止するか

判定不能

これは　次のような機械 H が存在しないという意味

■ 機械 M が入力 x で停止するならば　H は入力 (M, x) を受理（して停止）

■ 機械 M が入力 x で停止しないなら　H は入力 (M, x) を拒否（して停止）

次のように「半判定」する機械 U なら存在する

■ 機械 M が入力 x で停止するならば　U は入力 (M, x) で停止する

■ 機械 M が入力 x で停止しないなら　U は入力 (M, x) で停止しない

再

問題 SR

入力　書換え規則の（有限）集合 R と文字列 $w \in \Sigma^*$

R の各規則は $u \to v$ という形（$u, v \in \Sigma^*$）

文字列の一部を u から v に書換えることができるという意味

つまり $xuy \Rightarrow_R xvy$ とできる

答　R による書換えを次々と w に施して ε（空文字列）にできるか

$w \Rightarrow_R^* \varepsilon$ と書くことにする

例 1

入力

$$R \begin{cases} \text{aa} \to \text{bbb} \\ \text{aba} \to \text{a} \\ \text{bb} \to \varepsilon \end{cases}$$

$w = \text{aababab}$

答　○（受理）

（aababab $\Rightarrow_R$ aabab $\Rightarrow_R$ aab $\Rightarrow_R$ bbbb $\Rightarrow_R$ bb $\Rightarrow_R \varepsilon$ とできるので）

例 2

入力

$$R \begin{cases} \text{aa} \to \text{bab} \\ \text{aba} \to \text{a} \\ \text{bb} \to \varepsilon \end{cases}$$

$w = \text{aababab}$

答　×（拒否）

（a を消せる規則がないので）

問題 SR

入力 書換え規則の（有限）集合 R と文字列 $w \in \Sigma^*$

R の各規則は $u \to v$ という形（$u, v \in \Sigma^*$）

文字列の一部を u から v に書換えることができるという意味

つまり $xuy \Rightarrow_R xvy$ とできる

答 R による書換えを次々と w に施して ε（空文字列）にできるか

$w \Rightarrow_R^* \varepsilon$ と書くことにする

例 1

入力

$$R \begin{cases} \text{aa} \to \text{bbb} \\ \text{aba} \to \text{a} \\ \text{bb} \to \varepsilon \end{cases}$$

$w = \text{aababab}$

答 ○（受理）

（aababab $\Rightarrow_R$ aabab $\Rightarrow_R$ aab $\Rightarrow_R$ bbbb $\Rightarrow_R$ bb $\Rightarrow_R \varepsilon$ とできるので）

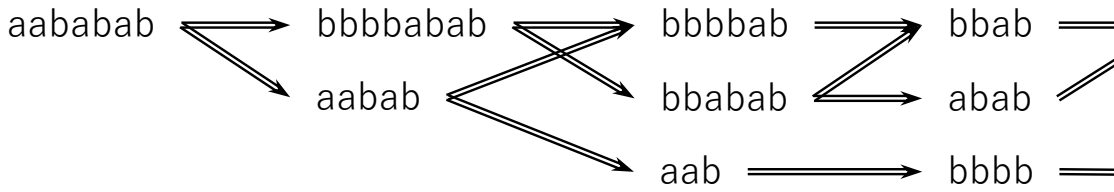
定理

SR は半判定可能

書換え 1 回で作れる文字列をすべて列挙（ε があれば受理）

書換え 2 回で作れる文字列をすべて列挙（ε があれば受理）

……と順に調べてゆけばよい



（この方法では $w \Rightarrow_R^* \varepsilon$ でないときは計算が停止しない）

実は SR は判定可能でないことが後で判る

その前にまず　二つの似た問題のどちらが
より判定可能でありそうか比べる論法（**帰着**）について考えよう

問題 SR

入力　書換え規則の集合 R と文字列 w

答　R による書換えを次々と w に施して空文字列にできるか

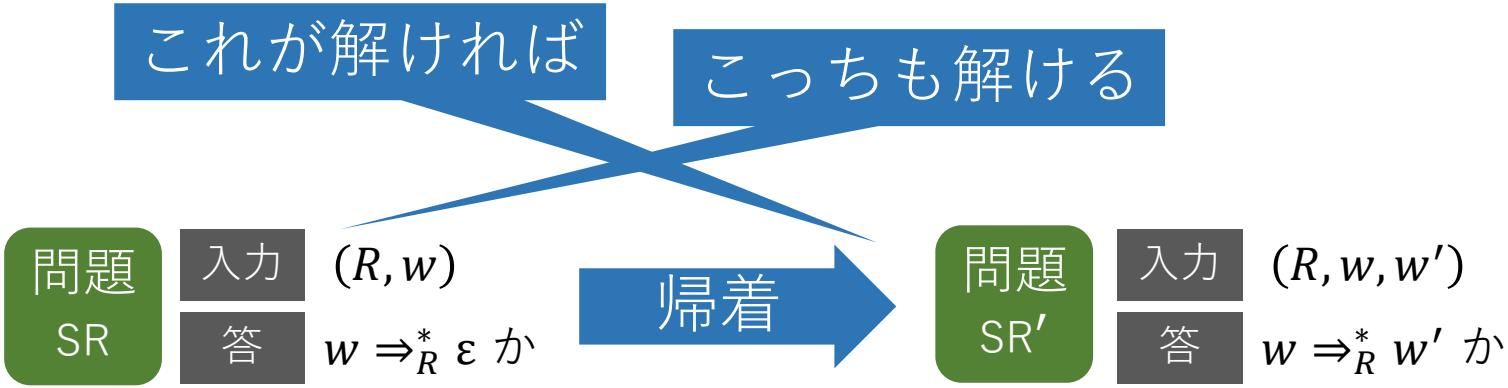
問題 SR′

入力　書換え規則の集合 R と文字列 w, w'

答　R による書換えを次々と w に施して w' にできるか

どちらが
より難しい？

問題の難しさの比較（**帰着**）



入力

(R, w)

が SR に属するか知りたい

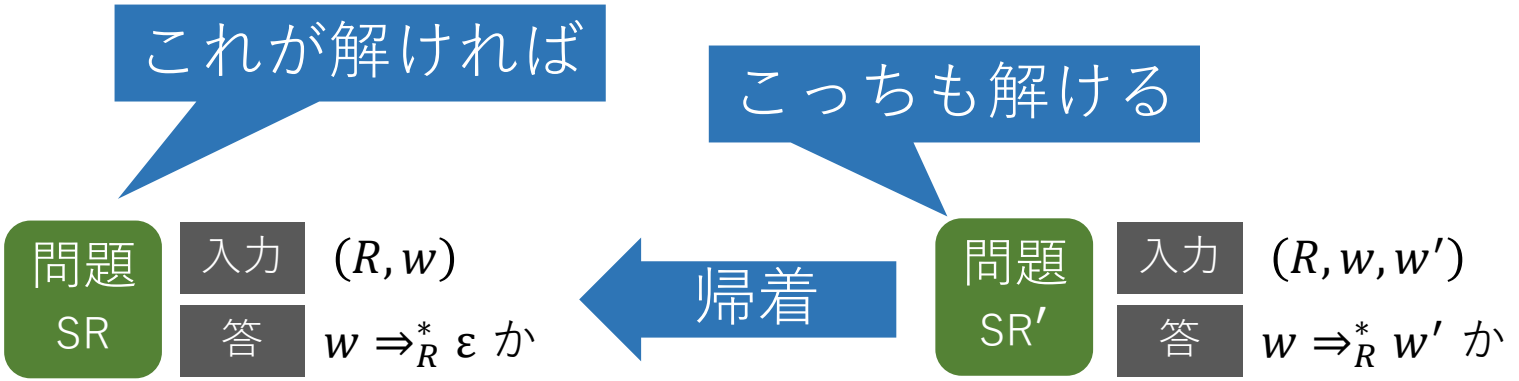
そのためには

(R, w, ε)

が SR′ に属するか調べれば良い

$$(R, w) \in \mathrm{SR} \quad \Leftrightarrow \quad (R, w, \varepsilon) \in \mathrm{SR}'$$

問題の難しさの比較（**帰着**）



入力

$(R \cup \{\triangleright w' \triangleleft \to \varepsilon\}, \triangleright w \triangleleft)$ ◀ ┈┈┈┈┈┈┈┈ (R, w, w')

（▸ と ◂ は新たな文字）

が SR′ に属するか知りたい

$$(R \cup \{\triangleright w' \triangleleft \to \varepsilon\}, \triangleright w \triangleleft) \in \mathrm{SR} \quad \Leftrightarrow \quad (R, w, w') \in \mathrm{SR}'$$

二つの問題は（判定可能かどうかに関しては）
「同じ難しさ」であることが判った！

問題の難しさの比較（**帰着**）

入力　書換え規則の集合 R と文字列 w

答　R による書換えを次々と w に施して空文字列にできるか

（前頁）

入力　書換え規則の集合 R と文字列 w, w'

答　R による書換えを次々と w に施して w' にできるか

問1　この帰着（SR″ も同じ難しさであること）を示して下さい

入力　書換え規則の集合 R と文字列 w, w''

答　R による書換えを次々と w に施して
w'' が現れる文字列にできるか

定理

SR は判定不能

これが判定不能と判っているので

これも

これも判定不能

∵ 右図の帰着による

(M,x) ⋯▸ $(R,\triangleright q_{始} x \triangleleft,$ 停$)$

但し R は　M の各状態 $q \in Q$ と文字 $\sigma \in \Gamma$ について次の規則を加えて作る

- もし $\delta(q,\sigma) = (q',\sigma',$ 左$)$ ならば　任意の $\tau \in \Gamma$ に対し規則 $\tau q\sigma \to q'\tau\sigma'$
- もし $\delta(q,\sigma) = (q',\sigma',$ 右$)$ ならば　規則 $q\sigma \to \sigma' q'$
- もし $\delta(q,\sigma) =$ 止 ならば　規則 $q\sigma \to$ 停

また　規則 $\triangleright \to \triangleright\sqcup$ および規則 $\triangleleft \to \sqcup\triangleleft$ も R に含める　すると

$(M,x) \in$ HALT $\Leftrightarrow$ 状況 $q_{始}x$ から遷移 M を辿ってゆくと
$\delta(q,\sigma) =$ 止 かつ $q \in Q_{受理}$ なる $q\sigma$ が現れる状況に到達

$\Leftrightarrow$ 文字列 $\triangleright q_{始} x \triangleleft$ に R の規則を施して
停を含む文字列に辿り着ける
（すなわち $(R,\triangleright q_{始} x \triangleleft,$ 停$) \in$ SR″）

このように　既に判定不能と判っている問題を帰着させることで
次々と多くの問題の判定不能が示される

問題 PCP

上下に文字列の書かれた札が何種類か与えられる
これらを横に有限枚並べて　（各種類の札が幾らでもある）
上段と下段の文字列を一致させることができるか？


Emil Post

例

b	a	ca	abc
ca	ab	a	c

➡ 受理

a	b	ca	a	abc
ab	ca	a	ab	c

とできるので

acc	ca	abc
cb	a	ab

➡ 拒否

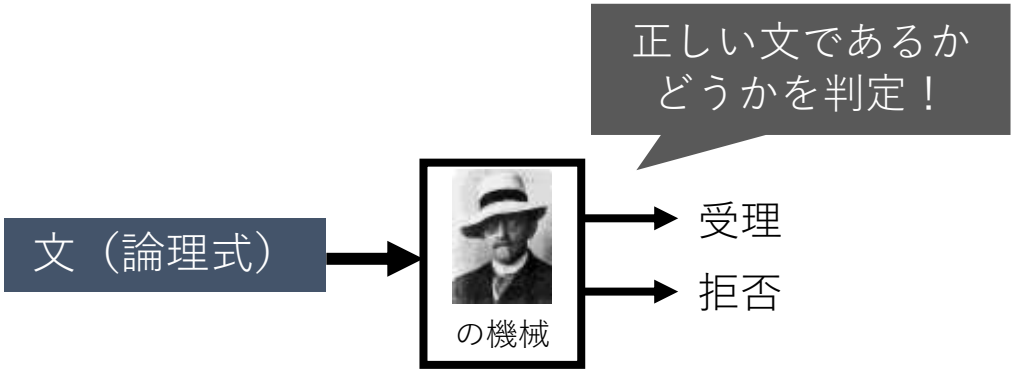
問2 問題 PCP の判定不能性を（これまでに判定不能と判っている問題どれかからの帰着により）示して下さい

問3 他に何らかの判定不能な問題を（調べて）選び　その判定不能性が如何なる帰着によって示されるか概要をわかりやすく説明して下さい

ヒルベルトの判定問題

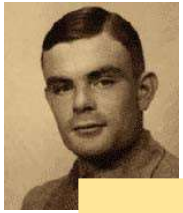
一階述語論理で書かれた文が与えられたとき
それが正しい（＝証明可能）か否かを判定する
ような機械的な手順はあるか？ (1928)

A. M. Turing. On computable numbers, with an application to the Entscheidungsproblem. *Proceedings of the London Mathematical Society* 2, **42**: 230–265, 1936.



ヒルベルトの判定問題

一階述語論理で書かれた文が与えられたとき
それが正しい（＝証明可能）か否かを判定する
ような機械的な手順はあるか？ (1928)



A. M. Turing. On computable numbers, with an application to the Entscheidungsproblem. *Proceedings of the London Mathematical Soci* **42**: 230–265, 1936.

もしあるとすると先程の HALT が判定できてしまうので、無い

計算量

既にこれまでの講義に出てきた
「多項式時間アルゴリズム」などの概念も
厳密にはチューリング機械で定義できる

As soon as an Analytic Engine exists, it will necessarily guide the future course of the science. Whenever any result is sought by its aid, the question will arise—By what course of calculation can these results be arrived at by the machine in the shortest time?

— *Charles Babbage (1864)*

解析機関は、実現すれば必ず科学の発展を方向づけるものとなろう。この機械を用いて結果を得ようとするとき重要となる問題があるーーそれは、**いかなる手順で計算すれば最も短時間で結果に辿り着くか**である。

解析機関（Analytic Engine）

「解析機関」『ウィキペディア日本語版』https://ja.wikipedia.org/w/index.php?title=%E8%A7%A3%E6%9E%90%E6%A9%9F%E9%96%A2&oldid=82653674）

計算量の考え方（原則） 1950～70年代

- チューリング機械での計算にかかる
  時間（遷移の回数）や空間（訪れる欄の数）を考える
  現実にかかる時間や空間をよく表している
- それが入力の長さに応じてどう変るか函数として表す
  「長さ n の入力なら必ず時間（や空間）が $T(n)$ 以内で済む」
  ような函数 T が計算量の尺度
- その函数の増大の速さに着目
  特に n の多項式以内か否かが重要

これまでの講義にも出てきた
「多項式時間アルゴリズム」
の正確な定義

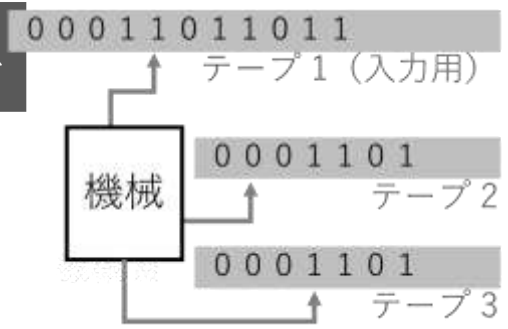
定義

機械 M が**多項式時間**であるとは　或る多項式 p が存在し
どの長さ n の入力に対しても M の計算が
時間 $p(n)$ 以内に終る（遷移 $p(n)$ 回以下で停止する）ことをいう
言語 A を判定する多項式時間の機械 M が存在するとき
A は**多項式時間判定可能**であるという

チャーチとチューリングの定立（続）

「現実的な手間で判定できる」 ＝ （チューリング機械で）多項式時間判定可能

テープの形や
読取り部の動き方など



チューリング機械の定義の細部は
計算法が多項式時間であるかないかに
影響を与えない

（「$O(n^3)$ であるかないか」など細かい量には
それなりに影響がある）

時間＝「基本的な操作の回数」と大雑把に考えてよい
- 四則演算やビットの操作など
- 機械語での命令数

定義

機械 M が**多項式空間**であるとは　或る多項式 p が存在し
どの長さ n の入力に対しても M の計算が
空間 $p(n)$ 以内である（テープ上で初めの欄の左右 $p(n)$ 個の範囲に留まる）ことをいう
言語 A を判定する多項式空間の機械 M が存在するとき
A は**多項式空間判定可能**であるという

多項式 q が存在して
入力長 n のとき $2^{q(n)}$ 時間以内

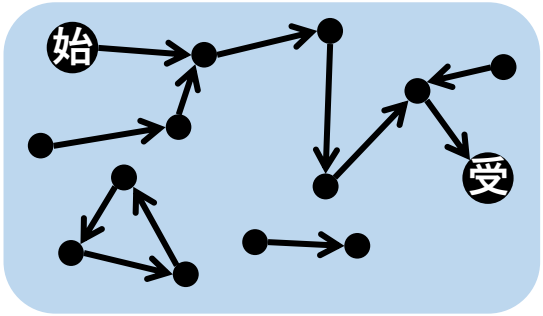
定理

多項式時間判定可能 $\overset{①}{\Longrightarrow}$ 多項式空間判定可能 $\overset{②}{\Longrightarrow}$ 指数時間判定可能

① 空間を 1 使う（新たな欄に移動する）にも
時間 1 かかるので

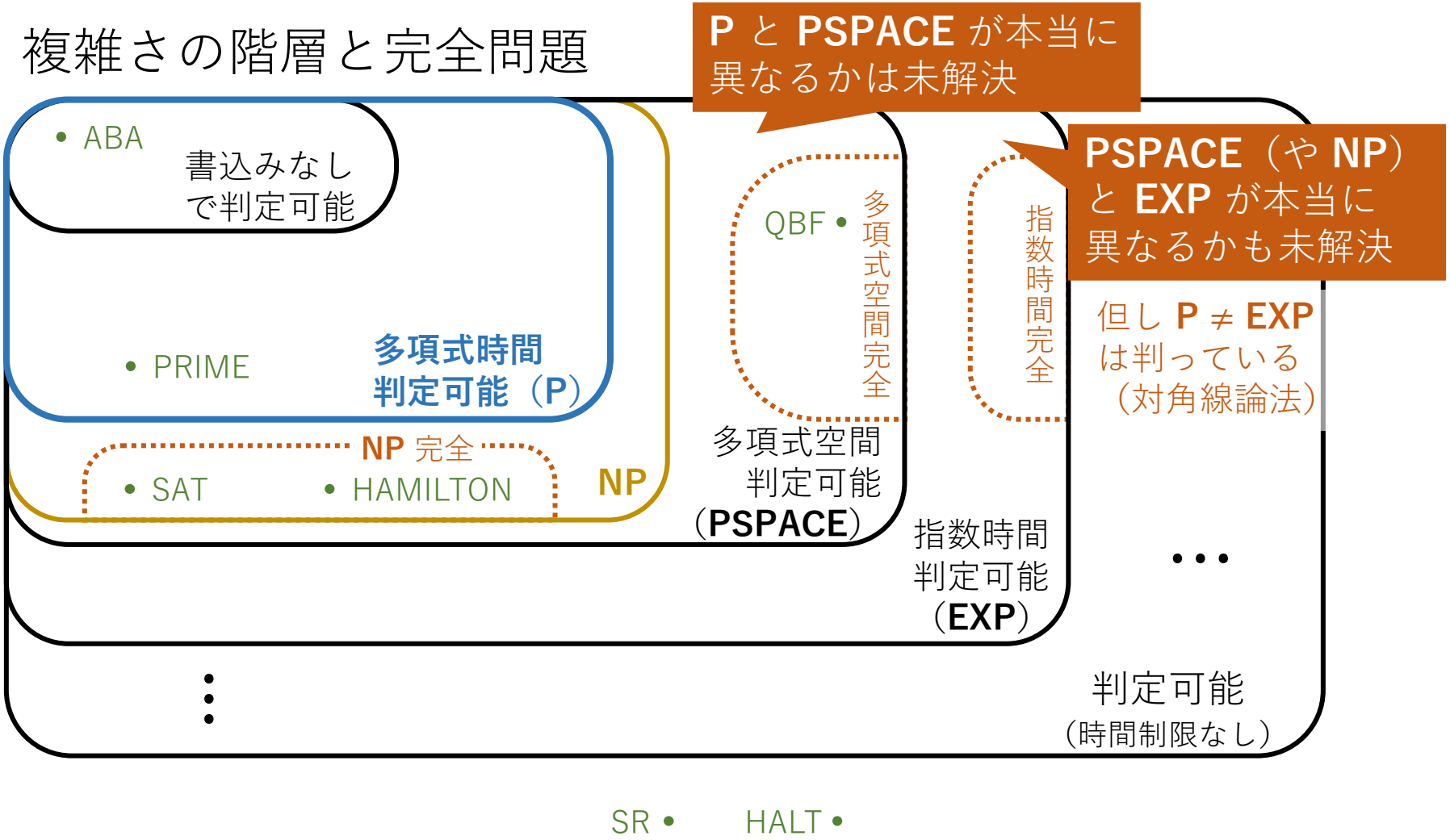
② 多項式空間機械 M に対して多項式 q が存在し
長さ n の入力に対する M の状況として
あり得るものは $2^{q(n)}$ 個以内

受理するのであれば時間 $\leq 2^{q(n)}$ で受理する



状況間の遷移関係

複雑さの階層と完全問題



レポート課題例

提出方法・期限等については
今井先生の指示に従って下さい

スライド中の 問1 問2 問3 のうち１つ以上を提出

自分で考えてもよいし、文献やウェブページなどを参考にしてもよいが、
自分の理解に基づいて説明し、**参考とした文献等については適切な形で明記**せよ。

今日の講義（や「計算の理論」全般）に興味をもった人は

M・シプサ著、太田・田中監訳『計算理論の基礎 原著第3版』共立出版（令和5年）

岩間一雄『アルゴリズム理論入門』朝倉書店（平成26年）

情報科学科ガイダンス

12号館**1212**教室

本日**18:45**～