↑スライドはここに置いてあります

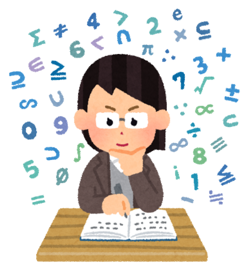
数学入門公開講座

計算量理論入門
―「複雑さ」をとらえる―

河村彰星
令和3年8月

計算量理論　(Computational) Complexity Theory

「どれほど計算し難いか」という尺度で　複雑さを測る



第一日
問題と機械

計算して問題を解くことの難しさを論じたいので……

まず　**問題**とは？

ここでは **各入力に対して　答が〇か×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　与えられた正整数（十進法で書く）が素数か答えよ

0 1 … 9 からなる文字列

入力	1	2	3	4	5	6	7	…
答	×	〇	〇	×	〇	×	〇	…

問題 ABA　与えられた文字列に「aba」が現れるか答えよ

a と b からなる

計算して問題を解くことの難しさを論じたいので……

まず　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　与えられた正整数（十進法で書く）が素数か答えよ

0 1 ⋯ 9 からなる文字列

入力	1	2	3	4	5	6	7	…
答	×	○	○	×	○	×	○	…

問題とは
コレ全体のこと！

問題です。57 は
素数でしょうか？

それは**入力**であって
問題ではない！

計算理論警察

計算して問題を解くことの難しさを論じたいので……

まず　問題とは？

文字列 u の後に v を付けた文字列を uv
文字列 $\underbrace{uu\cdots u}_{m\text{個}}$ を u^m　などと表すことにする

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**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 $\overset{\text{空文字列}}{\varepsilon}$ とか 119 とか）の全体を Σ^* で表す

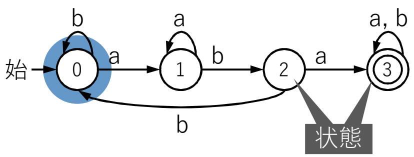
文字列のうち十進法で素数を表すもの全体が
言語 PRIME $= \{2,3,5,7,11,13,\ldots\} \subseteq \Sigma^*$ である

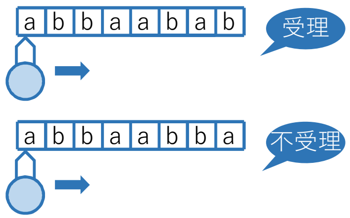
この講義では「文字列を入力すると　それが PRIME に属するか
答えてくれる計算機械」などについて考えたい

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

現在の状態

読んだ文字	0	1	2	3
a	1	1	3	3
b	0	2	0	3

（次にどの状態に行くか記した表）

定義

有限状態機械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
- **遷移規則**と呼ばれる函数 $\delta: Q \times \Sigma \to Q$

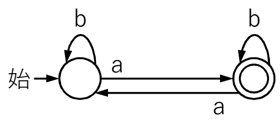
初め機械は始状態 $q_{始}$ にあり　与えられた文字列 $x \in \Sigma^*$ の左端から
次のこと（遷移）を繰返す
　状態 q で文字 σ を読むと
　状態を $\delta(q,\sigma)$ に変えて一つ右の文字に進む
右端まで終ったとき状態が $Q_{受理}$ に属すれば機械は x を**受理**したという

定義
機械 M が言語 A を**認識**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 を受理しない

すべての入力で正しく判断！

入力	ε	a	b	aa	ab	ba	bb	aaa	…
問題 が要求する 答	◯	×	×	◯	◯	×	◯	×	…
機械 による計算の結果	受理	不受理	不受理	受理	受理	不受理	受理	不受理	

演習　a と b からなる文字列のうち
（1）a が奇数回現れるもの全体
（2）左から 3 文字目が a であるもの全体
（3）右から 3 文字目が a であるもの全体
をそれぞれ認識する有限状態機械を作って下さい



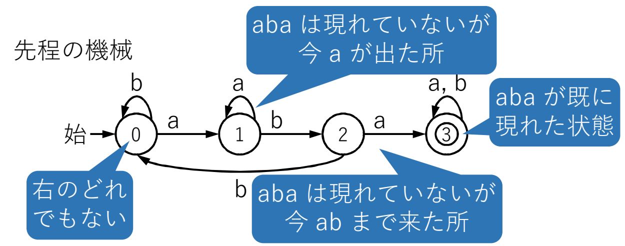
（1）の答（例）

有限状態機械の限界

ABA はなぜ認識できたか？

各時点で「今まで読んだ部分について覚えておくべき情報」が有限種類しかないから

先程の機械

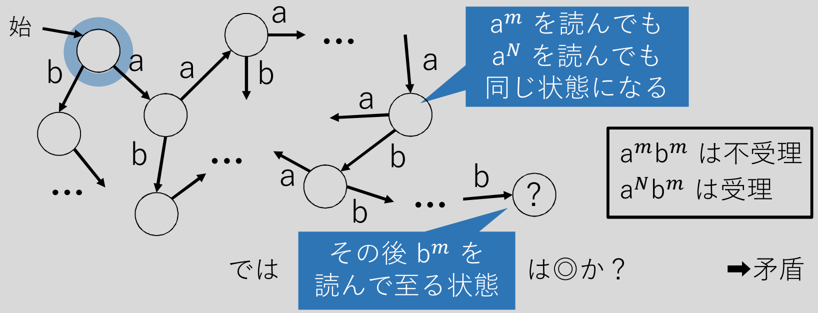


そういう単純な言語しか認識できない　　例えば次の言語は無理

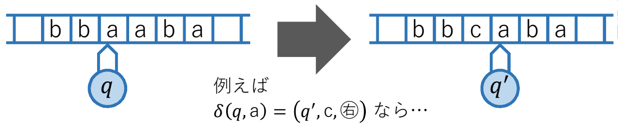
問題 MOREA　与えられた文字列に a が b よりも多く現れるか答えよ

定理
MOREA を認識する有限状態機械は存在しない

証明　そのような機械があるとする
状態は有限個なので　或る数 m と N（$m < N$ とする）が存在して



読むばかりでなく書く機能（と左右に動く機能）があれば？



定義
チューリング機械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** $\delta: Q \times \Gamma \to (Q \times \Gamma \times \{㊧, ㊨\}) \cup \{止\}$

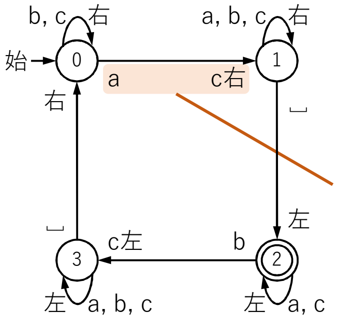
初め機械は始状態 $q_{始}$ にあり　テープ上に与えられた文字列 $x \in \Sigma^*$ の左端から始めて　次のこと（遷移）を繰返す
　状態 $q \in Q$ で文字 σ を読むと　$\delta(q, \sigma)$ が
- **止** ならば停止する
- (q', σ', d) なら　状態を q' にし σ' を書込み d の向きに一歩進む

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

※停止せず永遠に動き続ける場合は　受理していないと考える

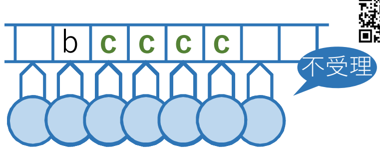
問題 MOREA 与えられた文字列に
a が b よりも多く現れるか答えよ

MOREA を認識する
チューリング機械

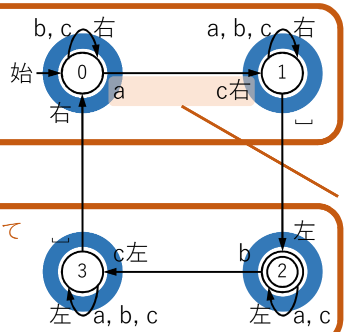


状態 0 で
文字 a を読むと
- 状態を 1 にし
- 文字 c を書込み
- 右へ行く

問題 MOREA 与えられた文字列に
a が b よりも多く現れるか答えよ



初めは状態 0
右に読み進め　最初の a を
c に書き換えて状態 1 になり
そのまま右端まで進む



状態 0 で
文字 a を読むと
- 状態を 1 にし
- 文字 c を書込み
- 右へ行く

右端に達すると状態 2 になって
左へ戻ってゆき　最初の b を
c に書き換えて状態 3 になり
そのまま左端まで進む

辿れる矢印が
ないときは止

読取りのみでは解けない問題を解けるようになった！

定義

言語 A を認識する（読取りのみの）有限状態機械が存在するとき
A は**正則**であるという

定義

言語 A を認識するチューリング機械が存在するとき
A は**認識可能**であるという

チャーチとチューリングの定立

「計算できる」
（機械的な手順で解ける）
＝
（チューリング機械で）
認識可能

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.

I. [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

† 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.

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

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

まとめ　第一日　問題と機械

- 問題＝言語　無限個の入力に正解を定めたもの
- 機械＝算法　有限に記述される計算手順
- 有限状態機械（書込み機能なし）により認識される　＝　正則
- チューリング機械（書込み機能あり）により認識される
  ＝　認識可能　≒　「計算できる」
- 正則でないが認識可能である言語が存在する