I238 計算の理論

上原 隆平

2018年I-1期（4-5月）

I238 Computation Theory

by

Prof. Ryuhei Uehara

Term I-1, April-May, 2018

計算(量)の理論

- ゴール1:
  - “計算可能な*関数/問題/言語/集合*”
    - 関数には2種類存在する;
      1. 計算不能(!)な関数
      2. 計算可能な関数
- ゴール2:
  - 「問題の困難さ」を示す方法を学ぶ
    - 計算可能な問題であっても、*手におえない* 場合がある！
      - 計算に必要な資源(時間・領域)が多すぎるとき

Computation Theory/ Computational Complexity

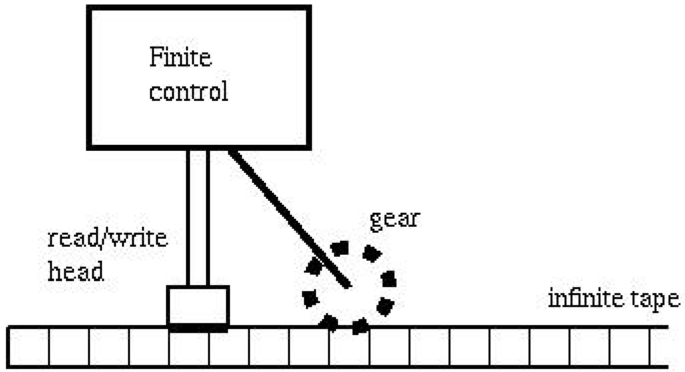
- Goal 1:
  - “*Computable Function/Problem/Language/Set*”
    - We have two functions;
      1. Functions that are not computable!
      2. Functions that are computable.
- Goal 2:
  - How can you show “*Difficulty of Problem*”
    - There are *intractable* problems even if they are computable!
      - because they require too many resources (time/space)!

3.マシンモデルと計算可能性

3. 「計算」とは何か？

チューリングマシンモデルの構成要素

- 有限制御部
- 無限長のテープ
  読み書きヘッド



動作手順は以下の通り:

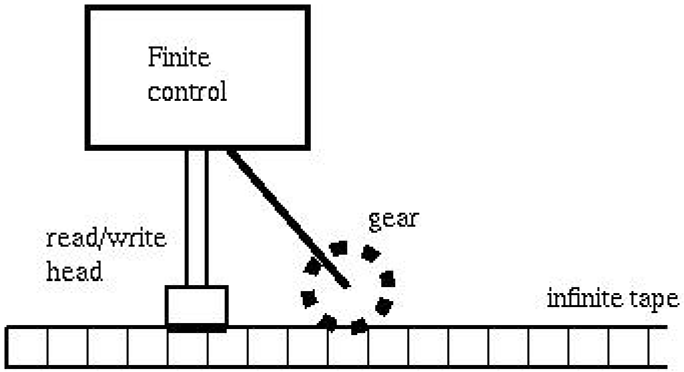
1. 読/書ヘッドが1文字読む
2. 文字の内容に応じて
   1. 文字を上書き
   2. ヘッドを右か左に一つずらす
3. 有限制御部の状態を変える.「受理」状態か「拒否」状態になっていなかったらステップ1に戻る.

3. Machine model & computability

3. Studies on what is a computation.

Turing machine model consists of

- finite control
- infinitely long tape
  with read/write head



It moves as follows;

1. r/w head reads the letter
2. according to the letter,
    1. rewrite the letter
    2. move the head to the left or right neighbor
3. change the state of control and go to step 1 until it comes to “accept” state or “reject” state.

3.マシンモデルと計算可能性

3. 「計算」とは何か？

チューリングマシンの形式定義:

チューリングマシンとは7つ組($Q, \Sigma, \Gamma, \delta, q_0, q_a, q_r$)で,

- Qは状態集合(有限集合)
- Σはブランク記号bを含まない入力アルファベット(有限集合)
- Γはテープアルファベットでb∈Γかつ$\Sigma \subseteq \Gamma$ (有限集合)
- $\delta: Q \times \Gamma \rightarrow Q \times \Gamma \times \{L,R\}$は遷移関数
- $q_0 \in Q$は初期状態
- $q_a \in Q$は受理状態
- $q_r \in Q$は拒否状態で, $q_a \neq q_r$

このあたりの詳細は
教科書を参考のこと.

3. Machine model & computability

3. Studies on what is a computation.

Formal definition of Turing machine :

A Turing machine is a 7-tuple $(Q, \Sigma, \Gamma, \delta, q_0, q_a, q_r)$, where

- Q is the set of states,
- Σ is the input alphabet not containing the blank b,
- Γ is the tape alphabet, where $b \in \Gamma$ and $\Sigma \subseteq \Gamma$,
- $\delta: Q \times \Gamma \rightarrow Q \times \Gamma \times \{L,R\}$ is the transition function,
- $q_0 \in Q$ is the start state,
- $q_a \in Q$ is the accept state, and
- $q_r \in Q$ is the reject state, where $q_a \neq q_r$

The details can be found in the textbook.

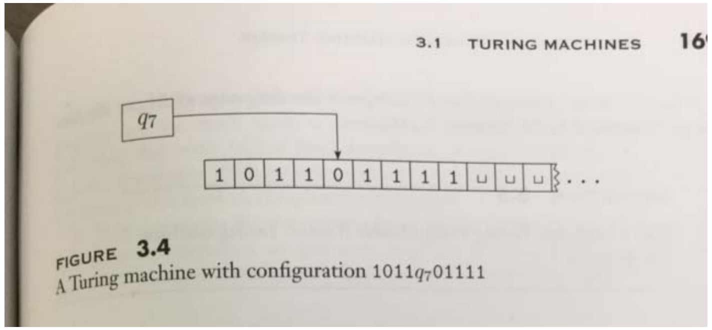
3.マシンモデルと計算可能性

3. 「計算」とは何か？

チューリングマシンの形式的定義：

チューリングマシンの状態uqvとは

- $u \in \Gamma^*$, $v \in \Gamma^*$
- $q \in Q$



チューリングマシンの状態遷移とは：

- 関数$\delta: Q \times \Gamma \rightarrow Q \times \Gamma \times \{L,R\}$により決まる.

チューリングマシンの計算とは：

- 初期状態 q_0 からはじまり,
- 与えられたテープ上の文字列で状態遷移し,
- q_a か q_r に到達したら停止する.
- q_a で停止したら入力を受理，q_r で停止したら入力を拒否と考える.

このあたりの詳細は
教科書を参考のこと.

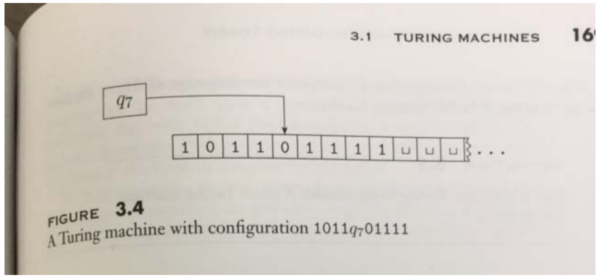
3. Machine model & computability

3. Studies on what is a computation.

Formal definition of Turing machine :

A configuration uqv of a Turing machine;

- $u \in \Gamma^*$, $v \in \Gamma^*$
- $q \in Q$



A transition of a Turing machine:

- Determined by the function $\delta: Q \times \Gamma \rightarrow Q \times \Gamma \times \{L,R\}$

A computation by a Turing machine:

- It starts from the start configuration q_0,
- makes transitions on a given tape string, and
- halts when it reach to q_a or q_r.
- It accepts on q_a, or rejects on q_r.

The details can be found in the textbook.

3.マシンモデルと計算可能性

3. 「計算」とは何か？

チューリングマシンの形式的定義：

チューリングマシンMが受理する文字列の集合をL(M)と書く.

– チューリングマシンMが言語Lを「認識する」

⇔L(M)=Lである.

[注意]チューリングマシンはいつでも停止するとは限らない. いつでも停止するチューリングマシンで認識できる言語だけを考えることもあるが, 本講義ではここは区別しない.

3. Machine model & computability

3. Studies on what is a computation.

Formal definition of Turing machine :

We denote by L(M) the set of strings accepted by a Turing machine M.

– A Turing machine M recognizes a language L⇔L(M)=L.

[Note] In general, a Turing machine does not halt necessarily. We sometimes consider Turing machines that always halt for any inputs. We will not distinguish between these models in this class.

3.マシンモデルと計算可能性

3. 「計算」とは何か？

形式的定義からいえること:

- Q, Σ, Γ, δ等はすべて有限集合
- あるチューリングマシンMは，これを符号化した<M>で記述できる.

チューリングマシンの模倣:

2つのチューリングマシンM1とM2があり，

M2は入力<M1,x>に対してM1に入力xを与えたときと同じ出力を計算するとき，M2はM1を<u>模倣する</u>という.

（簡単のため，M2(<M1,x>)=M1(x)と略記する．）

3. Machine model & computability

3. Studies on what is a computation.

From its formal definitions:

- Q, Σ, Γ, δ, etc. are all finite set
- Each Turing machine M can be represented by an encoded string <M>.

Simulation of a Turing machine :

Let M1 and M2 be two Turing machines.

If M2 computes the same output for any input <M1,x> as the output of M1 for the input x, we say M2 simulates M1.

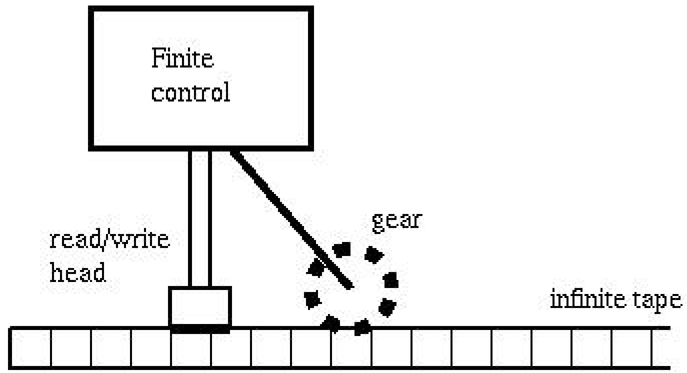
(To simplify, we denote by M2(<M1,x>)=M1(x).)

3.マシンモデルと計算可能性

3. 「計算」とは何か？

チューリングはチューリングマシンの万能性を証明

- *<u>どんな計算</u>*でも模倣できる
- 計算能力は昨今のスーパーコンピュータとも本質的に同じである! (<u>計算時間を無視</u>すれば)

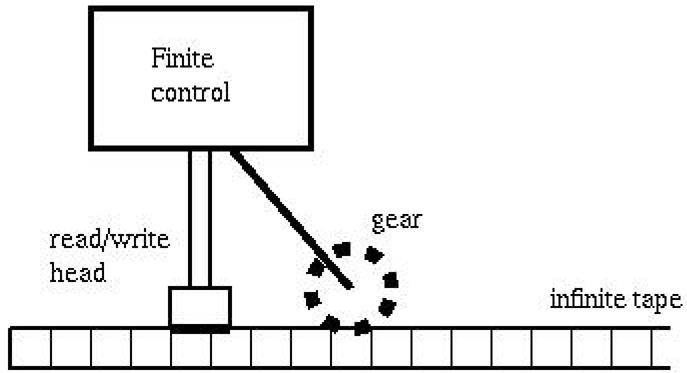


3. Machine model & computability

3. Studies on what is a computation.

Turing showed that Turing machine is *universal*

- it can simulate *any computation*
- it has the same computation power as recent supercomputers! (if you do not mind the *speed*)



3.マシンモデルと計算可能性

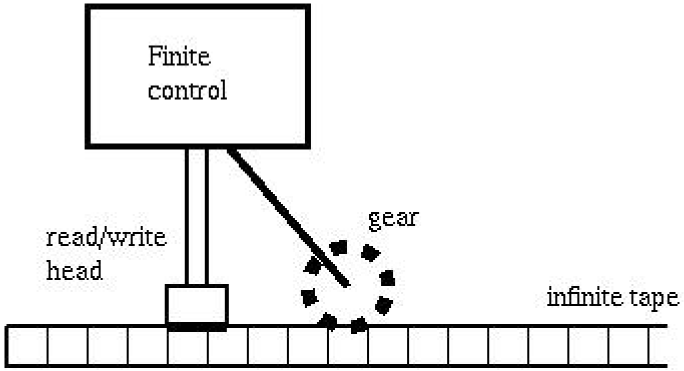
3.「計算」とは何か？

チューリングはチューリングマシンの**万能性**を証明

例1.

テープ上の文字の種類がk = 文字は2進;

それぞれの文字を2進文字列で符号化.



3. Machine model & computability

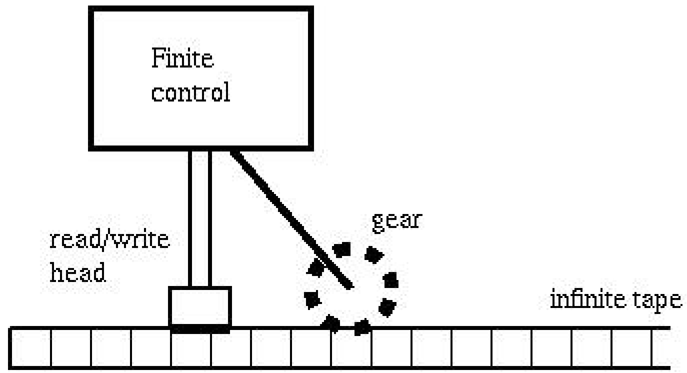
3. Studies on what is a computation.

   Turing showed that Turing machine is *universal*

E.g. 1.

   k letters tape = binary tape;

      each letter can be encoded by a binary string.



3.マシンモデルと計算可能性

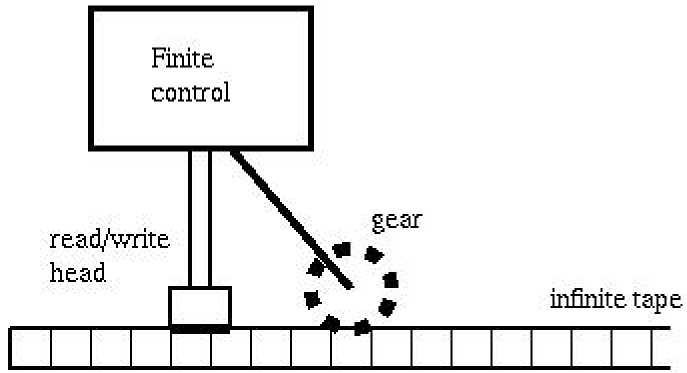
3.「計算」とは何か？

チューリングはチューリングマシンの万能性を証明

例2.

テープは両側無限長 = テープは右側だけ無限長;

1. テープを中央で「折り返」して重ねる
2. 4通りの文字に例1を適用
3. 有限状態部で「どちらのテープか」を管理



3. Machine model & computability

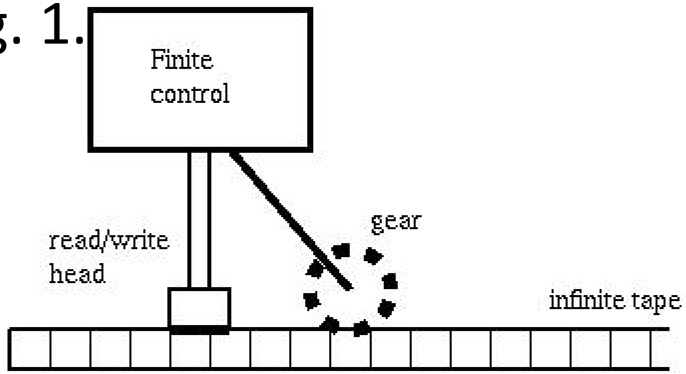
3. Studies on what is a computation.

   Turing showed that Turing machine is *universal*

E.g. 2.

infinite *both* sides = infinite just right side;

1. “fold” the tape at the center
2. for the four letters, apply E.g. 1.
3. finite control has a state for “which tape?”



3.マシンモデルと計算可能性

3. 「計算」とは何か？

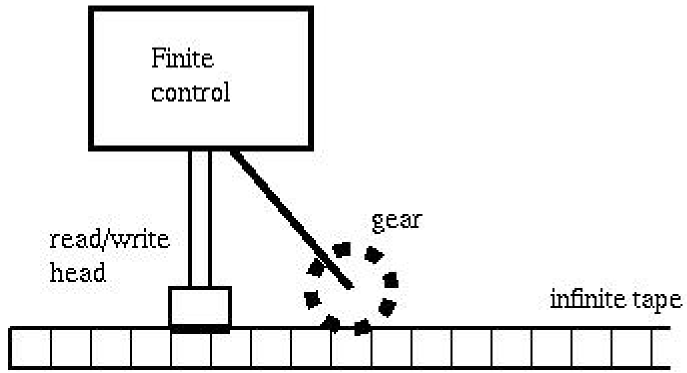
チューリングはチューリングマシンの万能性を証明

例3.

k本の(2進)テープ = 1本の2進テープ;

1. k本のテープを1本のテープに「積み重ね」る
2. 2^k 種類の文字に例1を適用

[練習問題]
各テープヘッドの位置は
どう管理すればよいか？



3. Machine model & computability

3. Studies on what is a computation.

Turing showed that Turing machine is *universal*

E.g. 3.

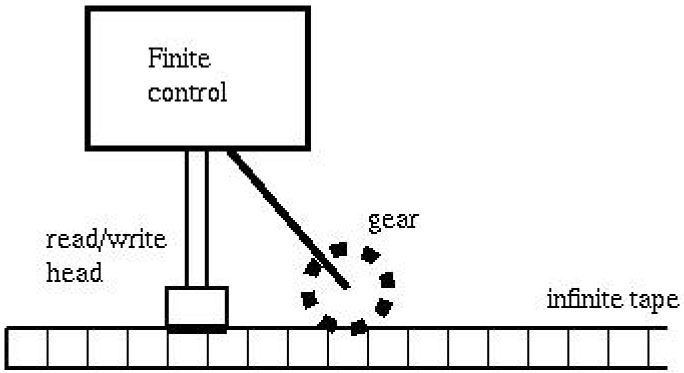
k (binary) tapes = 1 binary tape;

1. "stack" the *k* tapes onto a tape
2. for the 2^k letters, apply E.g. 1.

[Exercise]

How can it manage

each position of tape head?



3.マシンモデルと計算可能性

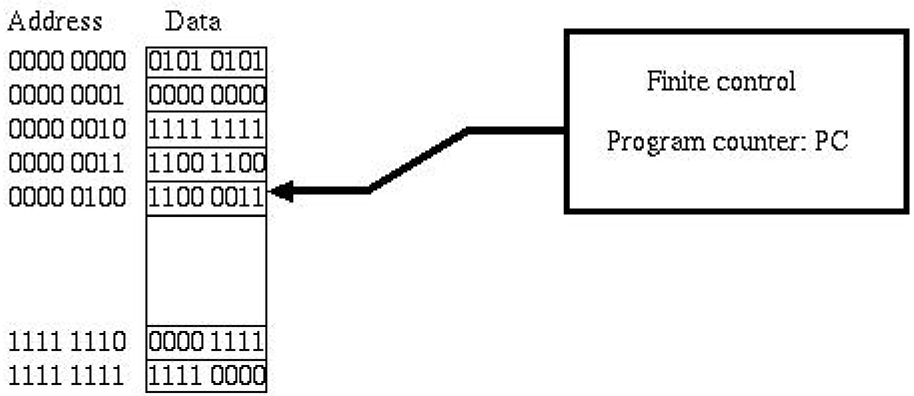
3.「計算」とは何か？

チューリングはチューリングマシンの万能性を証明

例3'.

*k*テープ = いわゆる「フォン・ノイマン式」計算機

=普通の *k* ビットのコンピュータ



3. Machine model & computability

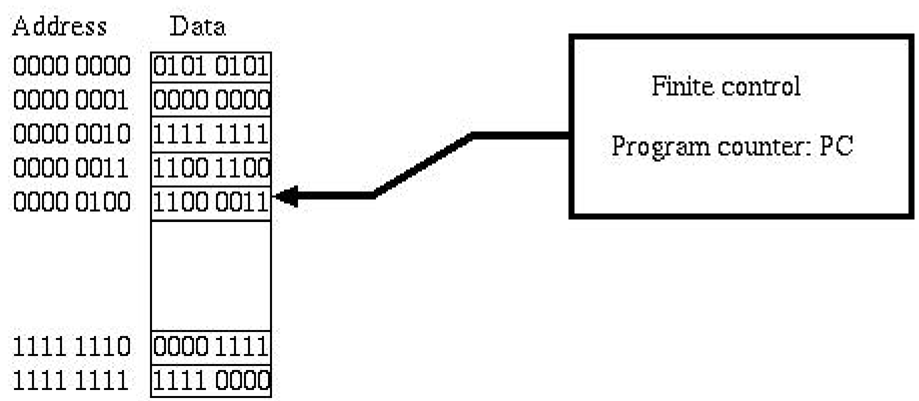
3. Studies on what is a computation.

   Turing showed that Turing machine is *universal*

E.g. 3’.

k tapes = so-called “von Neumann computer”

= k bit computer on your desktop



3.マシンモデルと計算可能性

3.「計算」とは何か？

チューリングはチューリングマシンの万能性を証明

二つの重要なアイデア;

1. チューリングマシン T は以下の2つの内容をつないだ(とても長い)2進文字列で表現することができる.
   1. 有限制御部を書き下した文字列
   2. テープの内容を書き下した文字列
2. 万能チューリングマシン U は2進文字列で表現された任意のチューリングマシン T の動作を模倣できる

   (マシン U は一種の「シミュレータ」)

3. Machine model & computability

3. Studies on what is a computation.

   Turing showed that Turing machine is *universal*

   Two crucial ideas;

   1. A Turing machine T can be encoded as a (looooong) binary string that consists of
      1. string that represents the finite control
      2. string that represents the contents on the tape
   2. A universal Turing machine U simulates any Turing machine T represented in the binary string.

      (The machine U is a kind of "simulator")

3.マシンモデルと計算可能性

3. 「計算」とは何か？

チューリングはチューリングマシンの万能性を証明

「関数」*Fu* を使って示せば:

入力 $<T,x>$ → 関数 *Fu* → 出力 $y=T(x)$

入力 $<T,x>$: 「Tを符号化したもの」と「Tへの入力x」を符号化したものの連結
出力 $T(x)$: マシンTに入力 xを与えたときの出力

[定理] (チューリング1936)
与えられた任意のチューリングマシンTとそれへの入力 xに対して,
$T(x)$ を計算(模倣)する(万能)チューリングマシンUが存在する.

3. Machine model & computability

3. Studies on what is a computation.

Turing showed that Turing machine is ***universal***

In the term of "function" *Fu*:

input $<T,x>$ $\longrightarrow$ $\boxed{\text{Function } Fu}$ $\longrightarrow$ output $y=T(x)$

input $<T,x>$: that represents "the code of *T*" and "the code *x* of the input to *T*"
output *T(x)*: the output of *T* with its input *x*

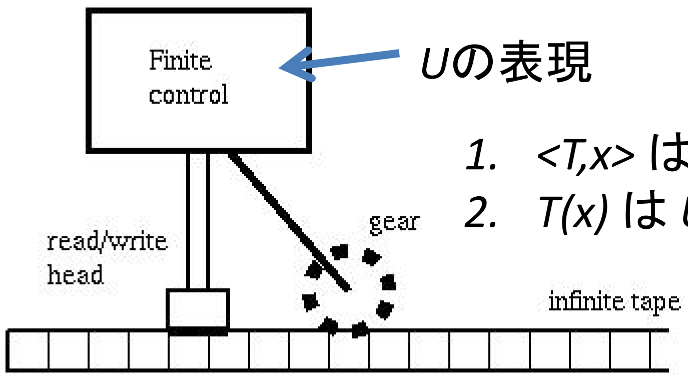
[Theorem] (Turing 1936)
There is a (universal) Turing machine *U* such that it computes *T(x)* for any given Turing machine *T* and its input *x*.

3.マシンモデルと計算可能性

3. 「計算」とは何か？

チューリングはチューリングマシンの万能性を証明

「チューリングマシン」モデルで表現すると:



*U*の表現

1. *<T,x>* は符号化されて最初にテープに書かれている
2. *T(x)* は *U* によって有限時間内にテープに書かれる

(*T(x)* が停止しないなら，*U*も同様．)

入力 *<T,x>*: 「*T* を符号化したもの」と「*T* への入力*x*」を符号化したものの連結

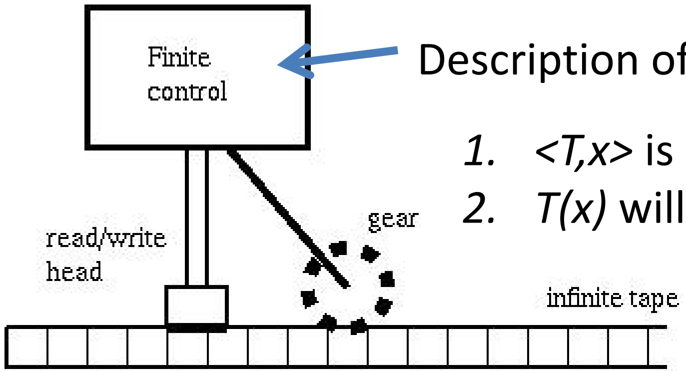
出力 *T(x)*: マシン *T* に入力 *x* を与えたときの出力

3. Machine model & computability

3. Studies on what is a computation.

Turing showed that Turing machine is *universal*

In the term of “Turing machine”:



Description of *U*

1. *<T,x>* is encoded and written on the tape at first
2. *T(x)* will be written on the tape by *U* in finite time (if *T(x)* does not halt, so is U.)

input *<T,x>*: that represents “the code of *T*” and “the code *x* of the input to *T*”
output *T(x)*: the output of *T* with its input *x*

4. 計算不能性と対角線論法

4. 計算不能な問題

以下の問題を解くチューリングマシンは存在しない：

停止性判定問題HALT (停止するかどうかを決定する問題)

入力：チューリングマシン*T*と
それへの入力*x*を符号化した文字列<*T*,*x*>

出力: *T* に入力 *x* を与えると，停止するか？

Yes: *T*(*x*) は(有限時間内に)停止する

No: 停止しない(無限ループ)

正確に言えば，停止性判定問題を解くチューリングマシン *U'* は存在しない．

…証明は「対角線論法」を用いて行う

4. Undecidability and Diagonalization

4. Undecidable problem

The following problem cannot be solved by any Turing machine:

Problem HALT(Problem of deciding halting)

input：a code <*T,x*> of Turing machine *T* and an input *x*

output: *T* will terminates for the input *x*?

Yes: if *T*(*x*) terminates

No: otherwise.

Precisely, we can show that there is no Turing machine *U'* that computes the halting problem

…Proof is done by “diagonalization”