8. Turing 機械入門(1)

8.1. コンピュータで解けない問題

8.2. チューリング機械

8.3. チューリング機械のプログラム技法
8.4. 基本チューリング機械の拡張
8.5. 制限されたチューリング機械
8.6. チューリング機械とコンピュータ

8. Turing Machine (1)

8.1. Unsolvable Problems for Computer

8.2. Turing machine

8.3. Programming Techniques for TM

8.4. Extension of basic TM

8.5. Restricted TM

8.6. Turing machine and real computer

8.2. Turing 機械とは

すべての
命題は証明
できるのか?

No!!

すべての関数は
計算できるのか?

No!!

8.2.1. チューリング機械モデル

「証明」とは何か？「計算」とは何か？193?〜

– クリーネの帰納的関数
– チューリングの Turing Machine モデル
– (Gödelの不完全性定理) …計算の理論

帰納的関数=TMで計算できる関数

Church の提唱: 計算可能な関数

枠外)
DNAコンピュータ
量子コンピュータ

8.2. Turing Machine

Every proposition can be proved?

No!!

Every function can be computed?

No!!

8.2.1. Turing machine model

What is ‘a proof’? What is ‘a computation’?: 193?～

- Kleene: ‘recursive function’
- Turing: Turing machine model
- (Gödel: Incompleteness theorems) …Computational Complexity

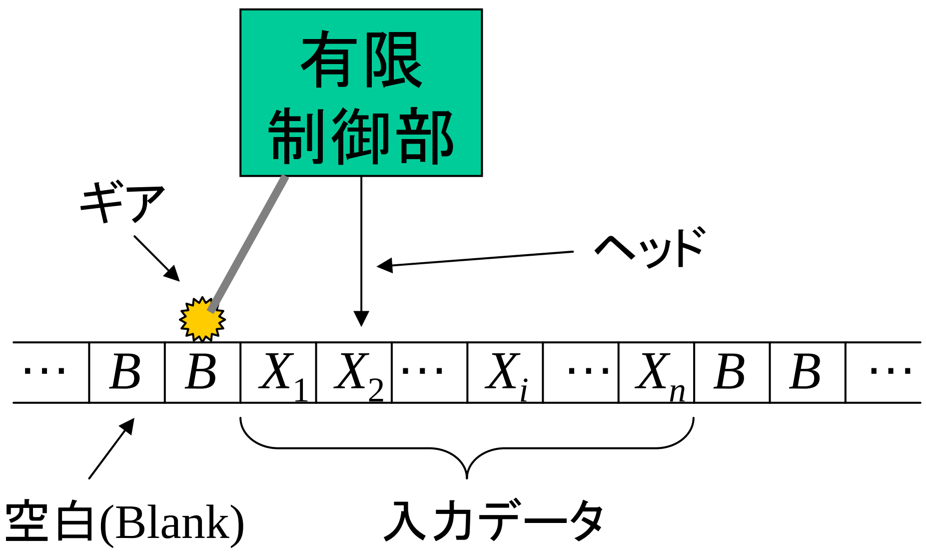
Recursive function = Function computable by TM

Church’s Thesis：Computable function

Exceptions)
DNA Computer, Quantum Computer

8.2. Turing 機械とは

8.2.1. チューリング機械モデル



有限制御部:
　有限個の状態を持つ

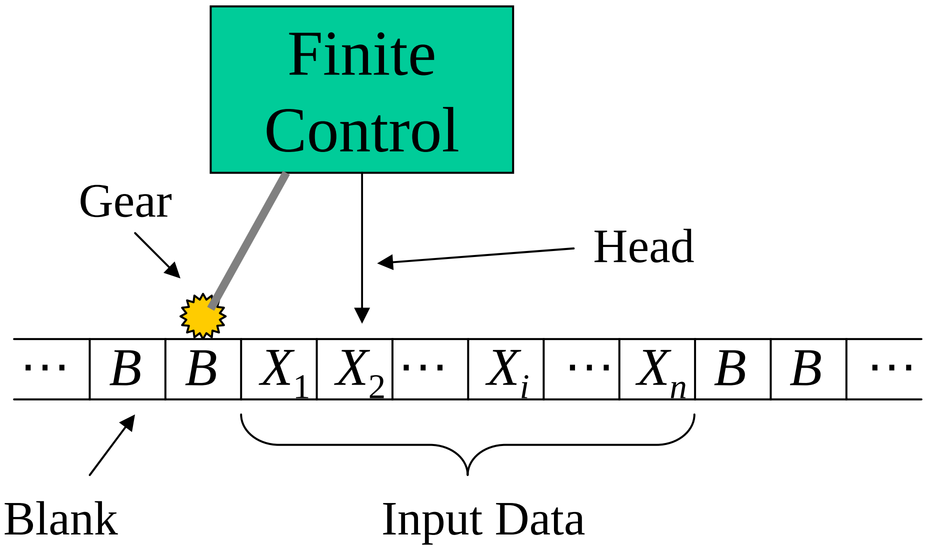
テープ:
　左右に無限の長さを持つ
　データが書いてあり、
　データのないところは
　　Blank が書いてある

ギア: テープを一つづつ左右に移動

ヘッド: テープ上の文字を読み/書きする

8.2. Turing Machine

8.2.1. Turing machine model



Finite Control:
has finite states.

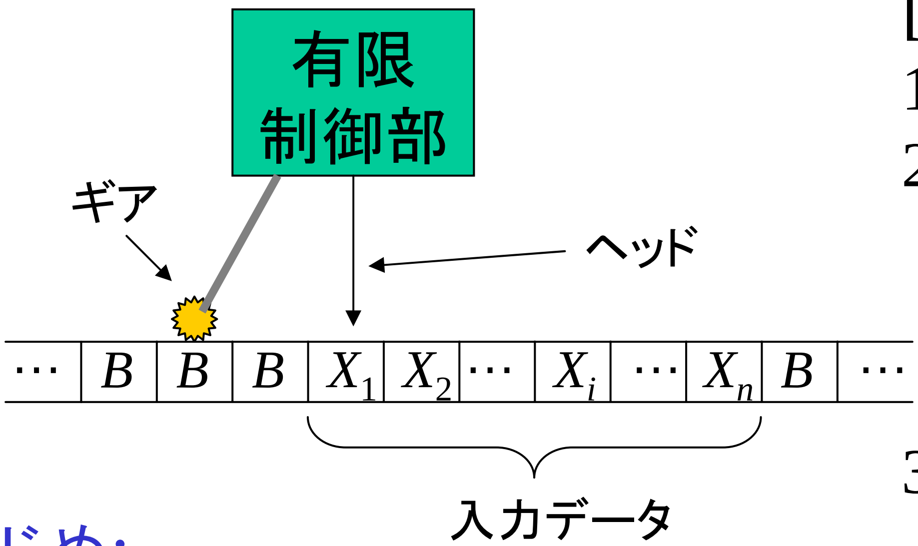
Tape:
Infinitely long to both sides
Input date is on the tape,
and Blank is filled on
the other cells.

Gear: moves the tape 1 cell to left/right

Head: reads/write a character on the cell.

8.2. Turing 機械とは

8.2.1. チューリング機械モデル



はじめ:

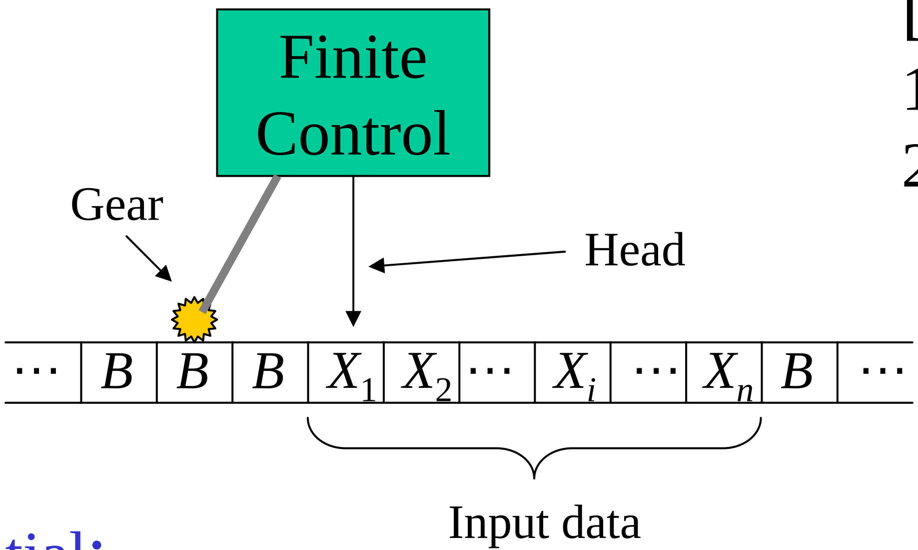
- 有限制御部は初期状態
- テープには入力が書かれている
- ヘッドは X_1 (入力の1文字目)の上にある

[動作プロセス]

1. ヘッド部の文字 X を読む
2. 状態 q と文字 X に応じて
   1. 状態を変更
   2. X を書き換える
   3. ヘッドを右/左に1移動
3. “受理状態”なら停止、さもなくば1へ

8.2. Turing Machine

8.2.1. Turing machine model



Initial:

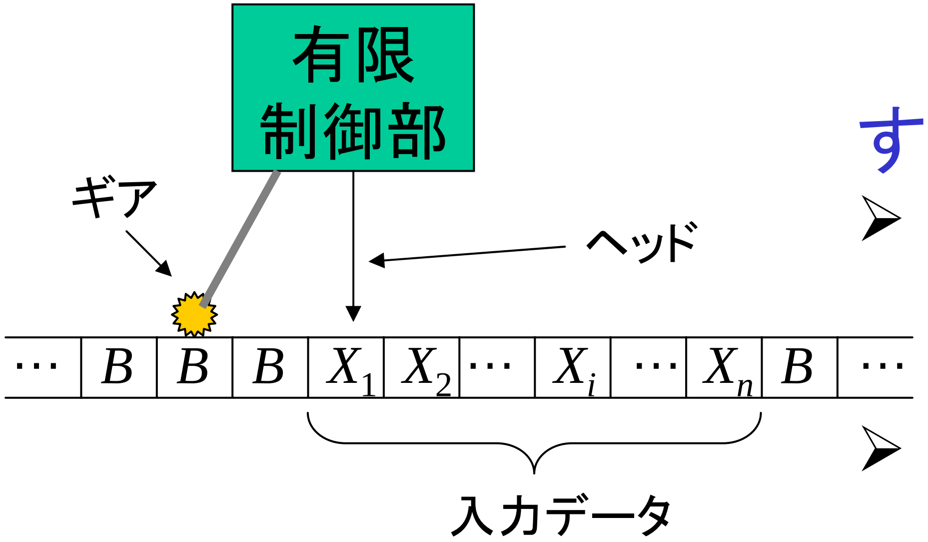
- Finite control is initial state
- Input word is on the tape
- Head is on X_1, the first letter of input.

[Computation Process]

1. Read the letter X on the tape
2. According to the state q and letter X,
    1. change the state
    2. replace X
    3. move the head 1 cell to left or right
3. Halt if "accepting state" or go to step 1.

8.2. Turing 機械とは

8.2.1. チューリング機械モデル

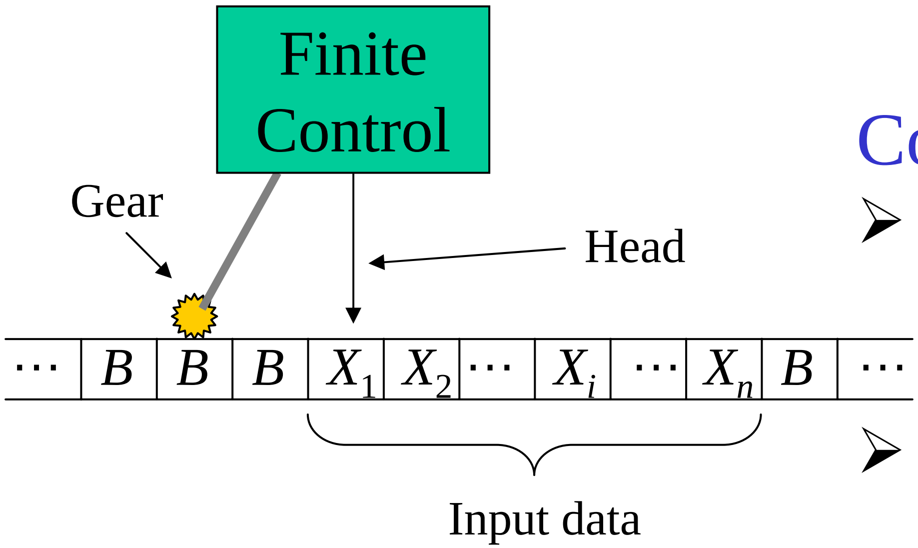


すでに学んだモデルとの関連

- オートマトン:
  - 入力を読むだけ
  - ヘッドを右に動かすだけ
- PDA:
  - オートマトン＋
  - 入力データの書かれていない部分にスタックを作る

8.2. Turing Machine

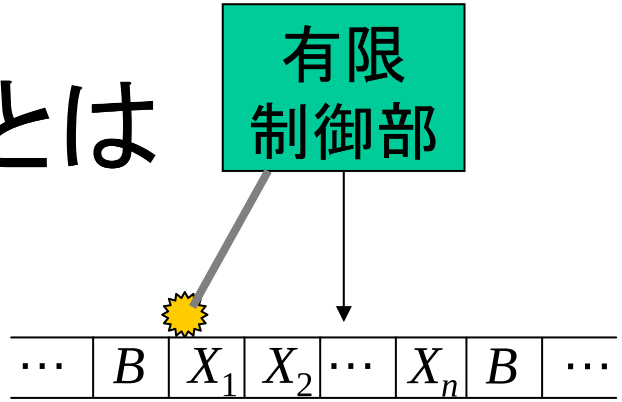
8.2.1. Turing machine model



Comparing to…

- Automaton:
  - Input data is just read.
  - Head only moves to right.
- PDA:
  - Automaton＋
  - We can make a stack on the left side.

8.2. Turing 機械とは



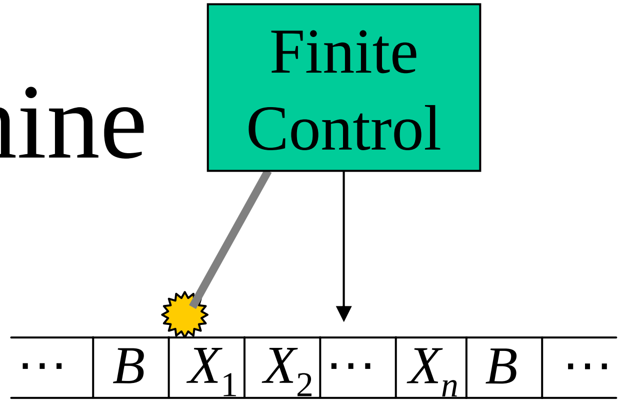
8.2.2. チューリング機械の記法

Turing Machine (TM) は以下の7つ組で表現:

$$M = (Q, \Sigma, \Gamma, \delta, q_0, B, F)$$

- Q: 状態の集合
- Σ: 入力アルファベット
- Γ: テープ上の文字を表現するアルファベット (よって$\Sigma \subset \Gamma$)
- δ: 遷移関数(後述)
- q_0: 初期状態(よって$q_0 \in Q$)
- B: 空白記号。$B \in (\Gamma - \Sigma)$。テープ上の有限個のマス以外は全部Bで埋められている、と仮定する。
- F: 受理状態(よって$F \subseteq Q$)

8.2. Turing Machine



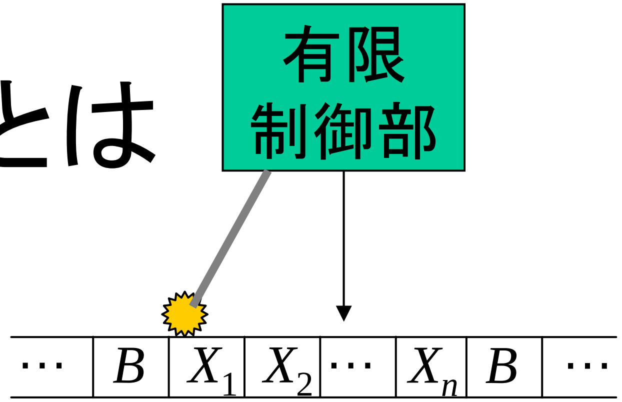
8.2.2. Notations for a TM

Turing Machine (TM) is defined by a 7-tuple:

$$M = (Q, \Sigma, \Gamma, \delta, q_0, B, F)$$

- Q: set of states
- Σ: input alphabets
- Γ: alphabets on the tape (hence $\Sigma \subset \Gamma$)
- δ: transition function (described later)
- q_0: initial state (hence $q_0 \in Q$)
- B: Blank. $B \in (\Gamma - \Sigma)$. We assume that all cells on the tape are filled by B except finite cells.
- F: accepting state (hence $F \subseteq Q$)

8.2. Turing 機械とは



8.2.2. チューリング機械の記法

Turing Machine (TM) の遷移関数 δ :

入力: $Q \times \Gamma$ ← ヘッドが読んでいる文字 X

現在の状態 p

ヘッドを移動する方向(**L**eft,**R**ight)

出力: $Q \times \Gamma \times \{L,R\}$

$$\delta : Q \times \Gamma \rightarrow Q \times \Gamma \times \{L,R\}$$

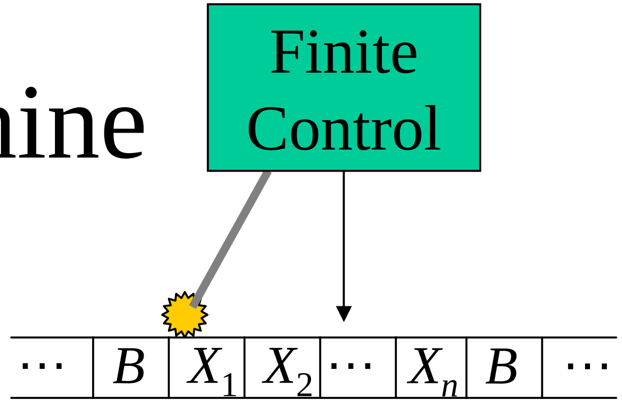
次の状態q　Xを書き換える文字Y

決定性: δ の値はいつでも1つ

非決定性: δ の値が複数ありえる

$$\delta : Q \times \Gamma \rightarrow 2^{Q \times \Gamma \times \{L,R\}}$$

8.2. Turing Machine



8.2.2. Notations for a TM

Transition function δ of a Turing Machine (TM):

Input: $Q \times \Gamma$ ← Letter X read by the head

current state p

Direction to move the head (**L**eft,**R**ight)

Output: $Q \times \Gamma \times \{L,R\}$

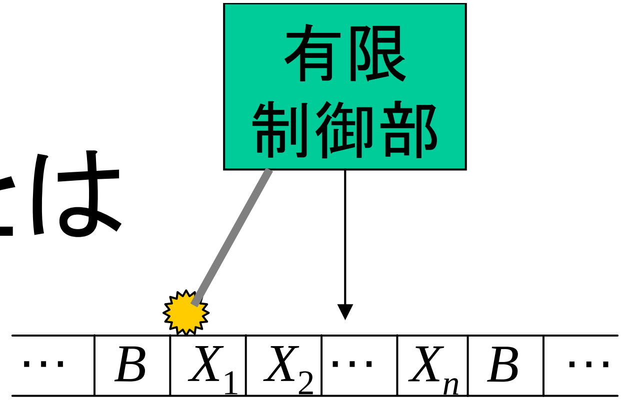
$$\delta : Q \times \Gamma \rightarrow Q \times \Gamma \times \{L,R\}$$

Next state q X is replaced by Y

Deterministic: δ is always determined uniquely

Nondeterministic: δ can have several values

$$\delta : Q \times \Gamma \rightarrow 2^{Q \times \Gamma \times \{L,R\}}$$



8.2. Turing 機械とは

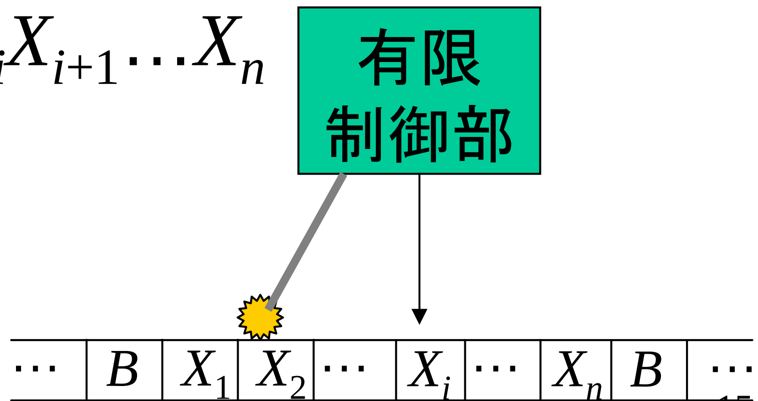
8.2.3. チューリング機械の時点表示(様相)

TM の<u>様相</u>は以下の情報が含まれていればよい:

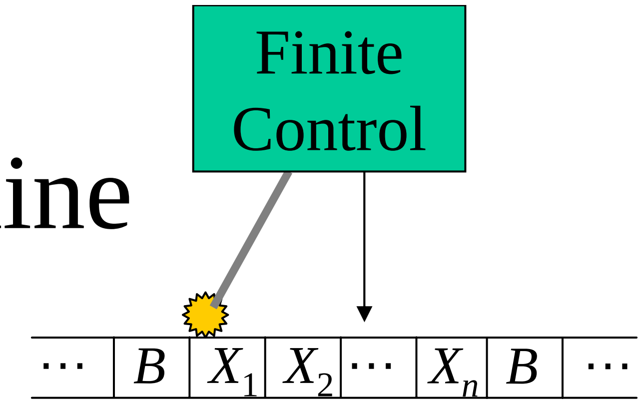
- 状態
- テープの内容
- ヘッドの位置

状態、入力、計算時間は有限なので、テープの内容も有限

TM M の様相: $X_1X_2\ldots X_{i\text{-}1}qX_iX_{i+1}\ldots X_n$



8.2. Turing Machine



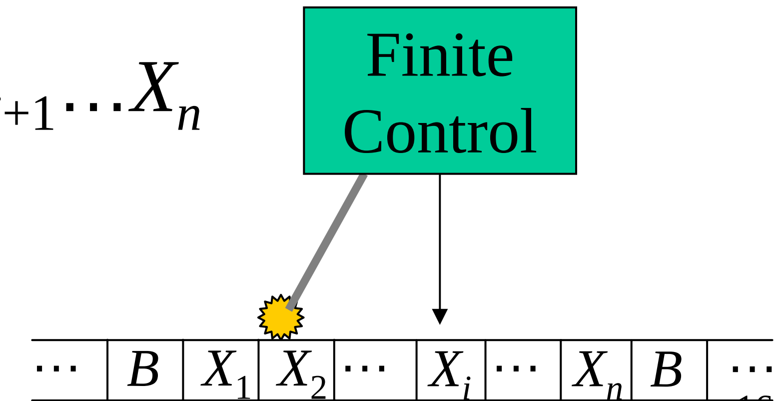
8.2.3. Instantaneous Description (ID) of a TM

The ID of a TM needs the following information:

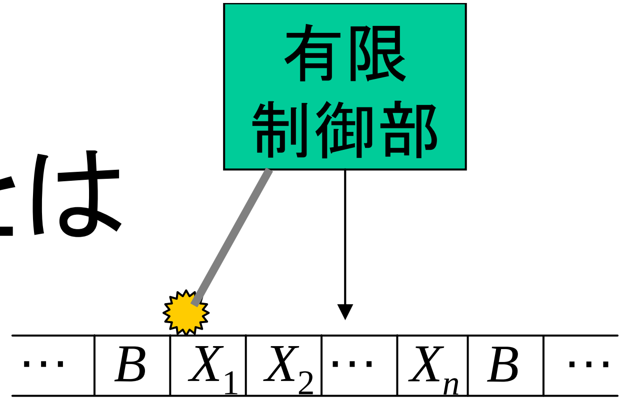
- state
- content of the tape
- position of the head

The contents of a tape is finite since states, input, and computation time are finite.

ID of a TM M: $X_1X_2\ldots X_{i\text{-}1}qX_iX_{i+1}\ldots X_n$



8.2. Turing 機械とは

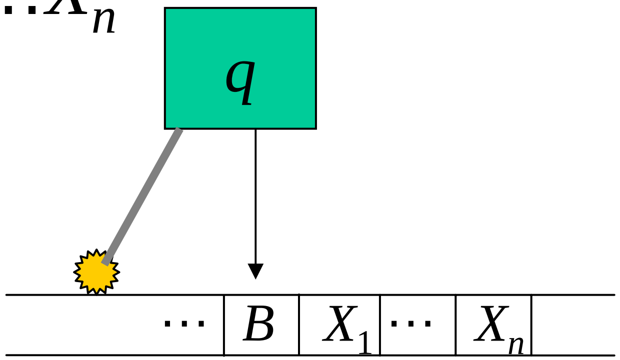


8.2.3. チューリング機械の時点表示(様相)

TM M の様相: $X_1X_2 \ldots X_{i\text{-}1}qX_iX_{i+1} \ldots X_n$

– 必要なら B を書く

$$qBX_1 \ldots X_n$$

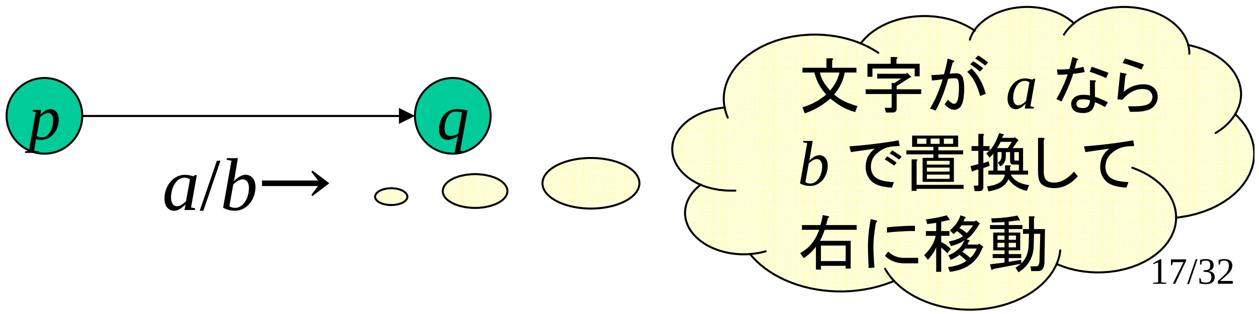


TM M の計算(遷移)の

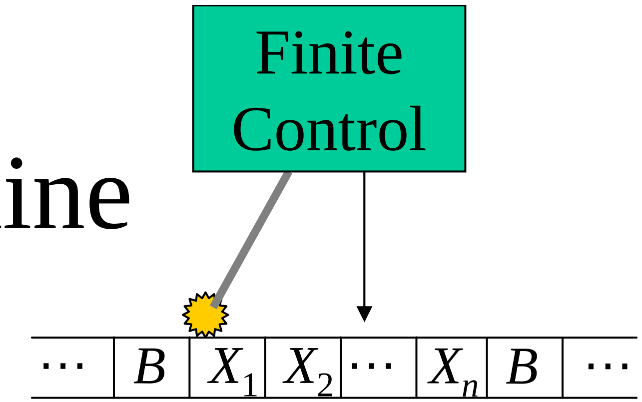
1ステップを $\vdash$ で、

0ステップ以上の遷移を $\overset{*}{\vdash}$ で表現するのは PDA と同様。

TMの遷移図



8.2. Turing Machine

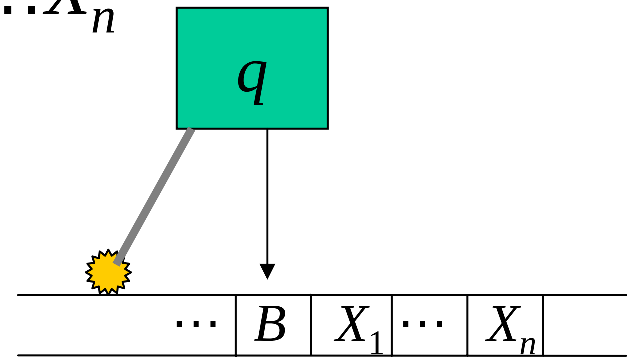


8.2.3. Instantaneous Description (ID) of a TM

ID of a TM *M*: $X_1X_2\ldots X_{i-1}qX_iX_{i+1}\ldots X_n$

– Write *B* if it is necessary

$$qBX_1\ldots X_n$$

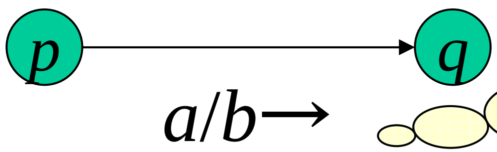


Transitions (computations) of a TM *M*:

1 step is described by $\vdash$,

0 or more steps are described by $\overset{*}{\vdash}$, as PDA.

Diagram for TM



Move the head to right after replacing letter *a* by *b*.

8.2. Turing 機械とは

8.2.3. チューリング機械の時点表示(様相)

例) $L = \{ ww^R \mid w \in \{0,1\}^* \}$

アイデア: 両端が同じ文字なら B で置換していき、全部 B になったら受理

1. 最初の文字が B なら受理
2. 最初の文字が0/1なら、
   ① その文字を「状態」で覚える
   ② その文字を B で上書き
   ③ 右端へ移動
   ④ 同じ文字なら B で上書き
   ⑤ 左端へ戻る
   ⑥ ステップ1へ戻る

8.2. Turing Machine

8.2.3. Instantaneous Description (ID) of a TM

Ex) $L = \{ ww^R \mid w \in \{0,1\}^* \}$

Idea: If leftmost and rightmost letters are the same, replace them by *B*. Accept if all letters become *B*.

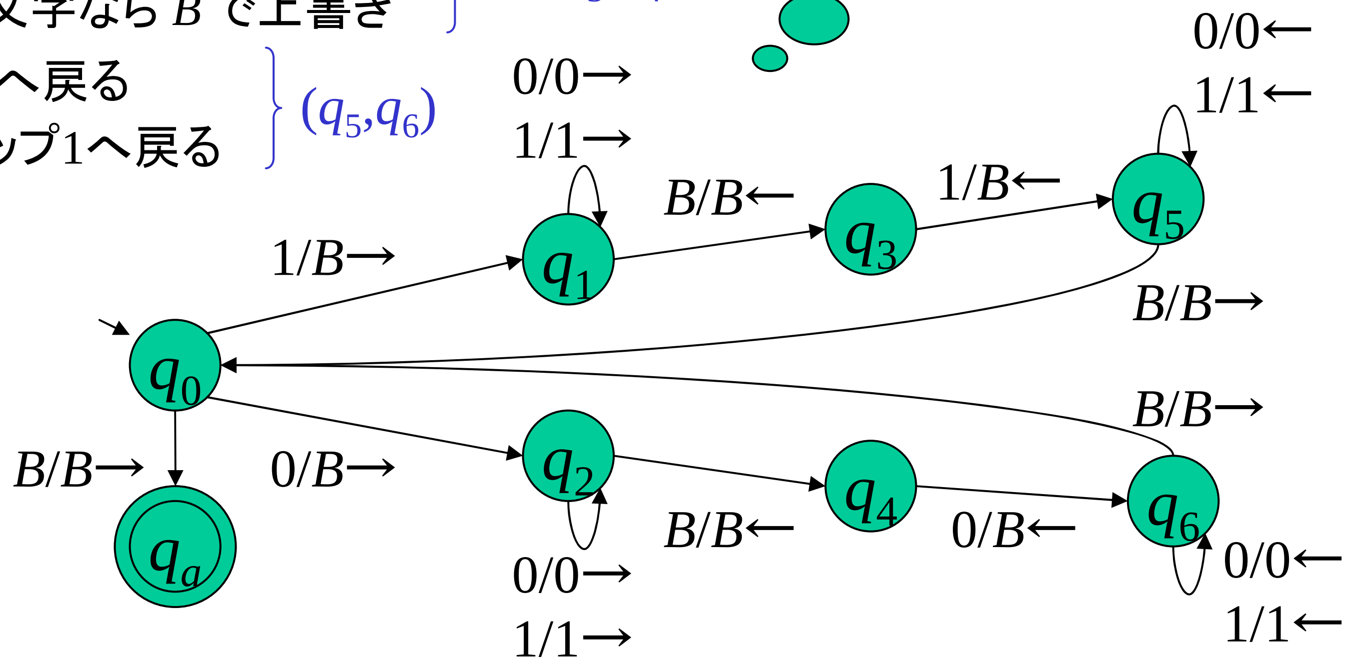
1. Accept if the first letter is *B*
2. If the first letter is 0/1,
   ① store the letter by the state
   ② replace the letter by *B*
   ③ move to the leftmost
   ④ replace the leftmost letter by *B* if it is the same
   ⑤ move to the rightmost
   ⑥ go to step 1.

例) $L = \{ ww^R \mid w \in \{0,1\}^* \}$

アイデア: 両端が同じ文字なら B で置換していき、全部 B になったら受理

1. 最初の文字が B なら受理 (q_0,q_a)
2. 最初の文字が0/1なら、
   - ① その文字を「状態」で覚える
   - ② その文字を B で上書き (q_1,q_2)
   - ③ 右端へ移動
   - ④ 同じ文字なら B で上書き (q_3,q_4)
   - ⑤ 左端へ戻る
   - ⑥ ステップ1へ戻る (q_5,q_6)

TMの動作の正当性は入力長に関する帰納法

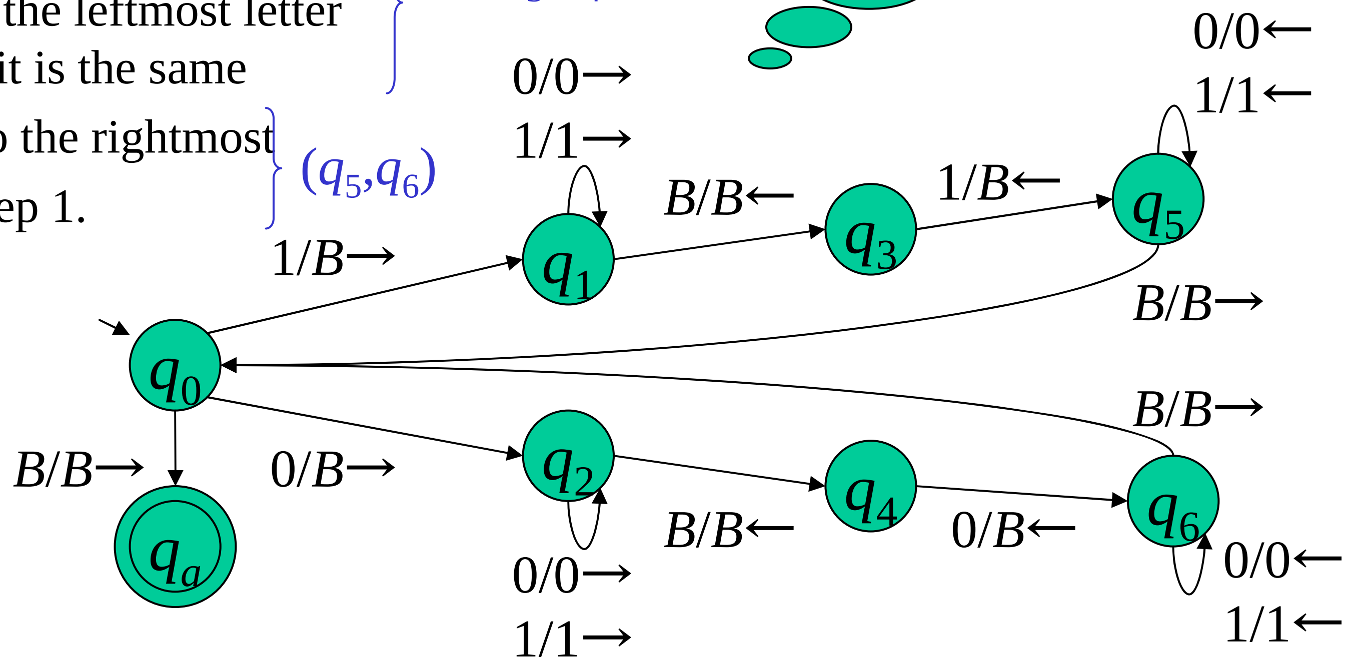


Ex) $L = \{ ww^R \mid w \in \{0,1\}^* \}$

Idea: If leftmost and rightmost letters are the same, replace them by *B*. Accept if all letters become *B*.

1. Accept if the first letter is *B* (q_0, q_a)
2. If the first letter is 0/1,
   ① store the letter by the state
   ② replace the letter by *B* (q_1, q_2)
   ③ move to the leftmost
   ④ replace the leftmost letter by *B* if it is the same (q_3, q_4)
   ⑤ move to the rightmost
   ⑥ go to step 1. (q_5, q_6)

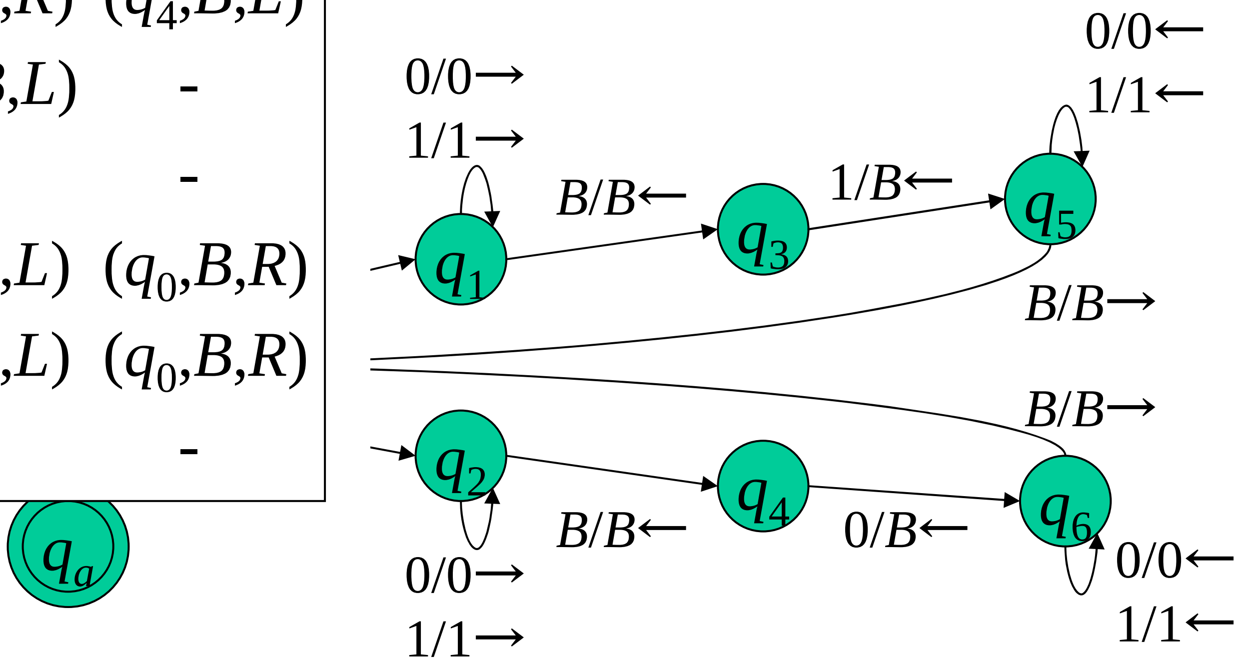
Correctness of the TM is proved by induction for the length of the input.



例) $L = \{ ww^R \mid w \in \{0,1\}^* \}$ を受理するTM

$M=(\{q_a,q_0,q_1,\ldots,q_6\},\{0,1\},\{0,1,B\}, \delta, q_0, B, \{q_a\})$
の形式的定義: δ は以下の通り

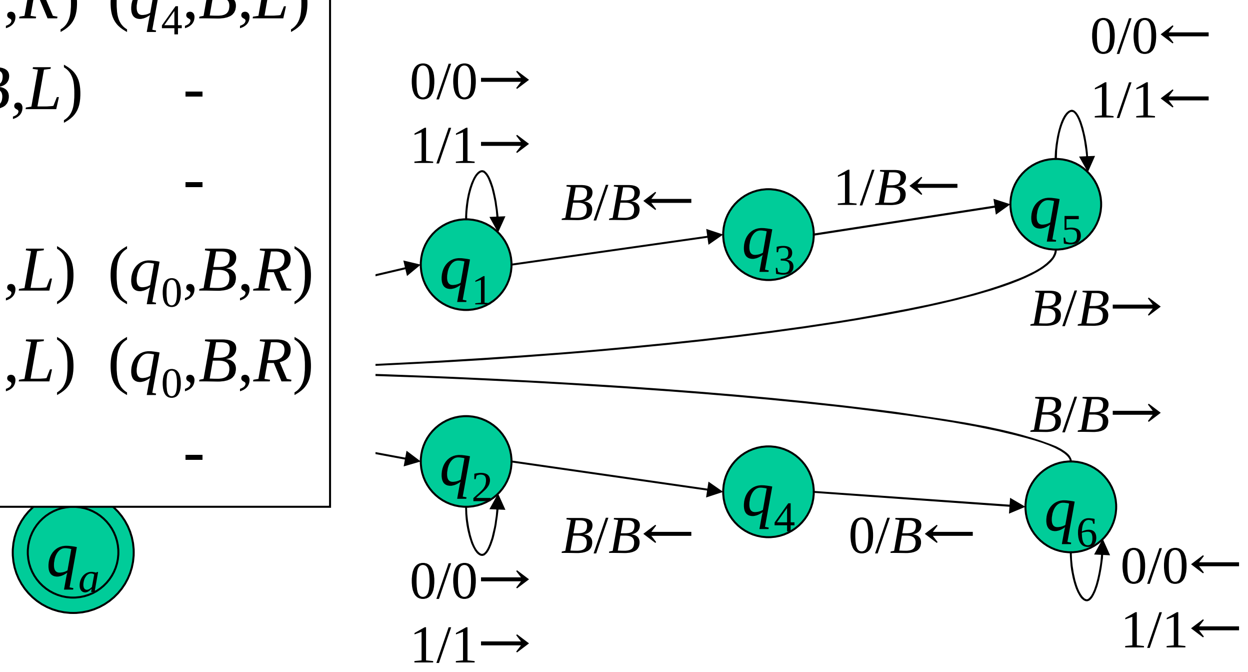
	0	1	B
q_0	(q_2,B,R)	(q_1,B,R)	(q_a,B,R)
q_1	$(q_1,0,R)$	$(q_1,1,R)$	(q_3,B,L)
q_2	$(q_2,0,R)$	$(q_2,1,R)$	(q_4,B,L)
q_3	-	(q_5,B,L)	-
q_4	(q_6,B,L)	-	-
q_5	$(q_5,0,L)$	$(q_5,1,L)$	(q_0,B,R)
q_6	$(q_6,0,L)$	$(q_6,1,L)$	(q_0,B,R)
q_a	-	-	-



Ex) $L = \{ ww^R \mid w \in \{0,1\}^* \}$ is accepted by TM

$M=(\{q_a,q_0,q_1,\ldots,q_6\},\{0,1\},\{0,1,B\}, \delta, q_0, B, \{q_a\})$, where δ is defined as follows:

	0	1	B
q_0	(q_2,B,R)	(q_1,B,R)	(q_a,B,R)
q_1	$(q_1,0,R)$	$(q_1,1,R)$	(q_3,B,L)
q_2	$(q_2,0,R)$	$(q_2,1,R)$	(q_4,B,L)
q_3	-	(q_5,B,L)	-
q_4	(q_6,B,L)	-	-
q_5	$(q_5,0,L)$	$(q_5,1,L)$	(q_0,B,R)
q_6	$(q_6,0,L)$	$(q_6,1,L)$	(q_0,B,R)
q_a	-	-	-



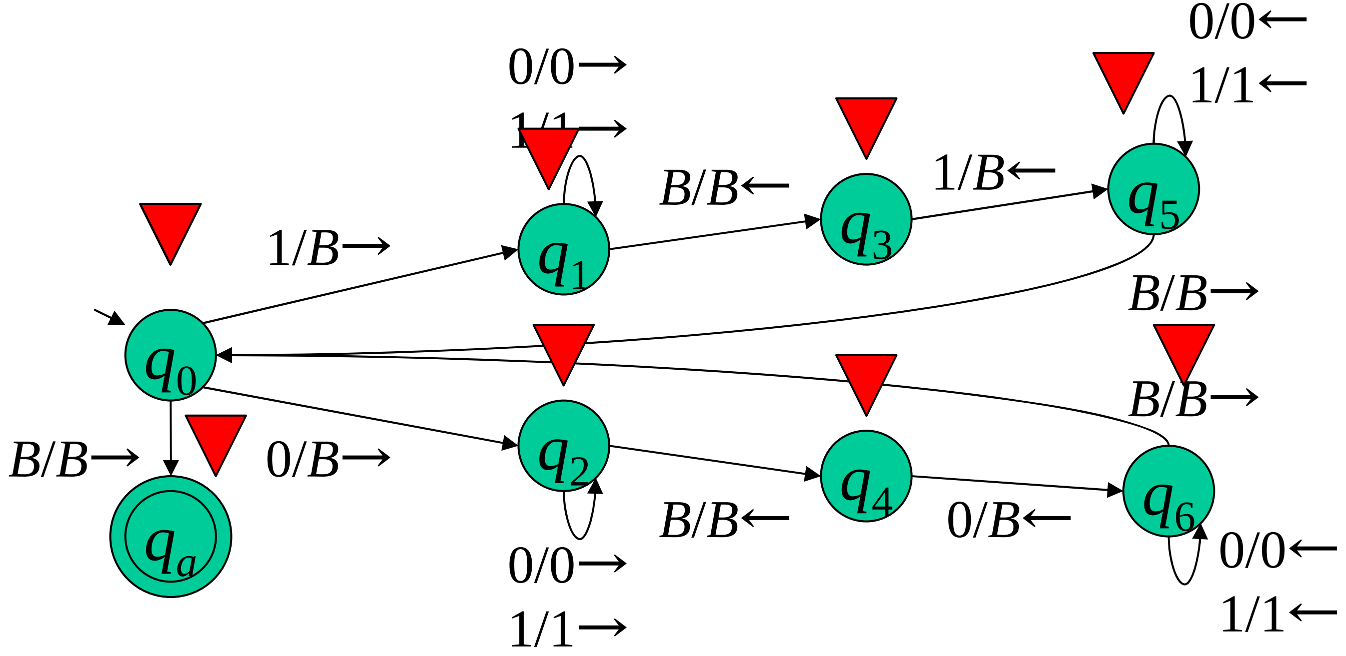
例) $L = \{ ww^R \mid w \in \{0,1\}^* \}$ を受理するTM

Mが入力 1001 を受理する計算は以下の通り:

$$q_0 1001 \vdash q_1 001 \vdash 0q_1 01 \vdash 00q_1 1 \vdash 001q_1 \vdash 00q_3 1$$

$$\vdash 0q_5 0 \vdash q_5 00 \vdash q_5 B00 \vdash q_0 00 \vdash q_2 0$$

$$\vdash 0q_2 \vdash q_4 0 \vdash q_6 \vdash q_0 \vdash q_a$$



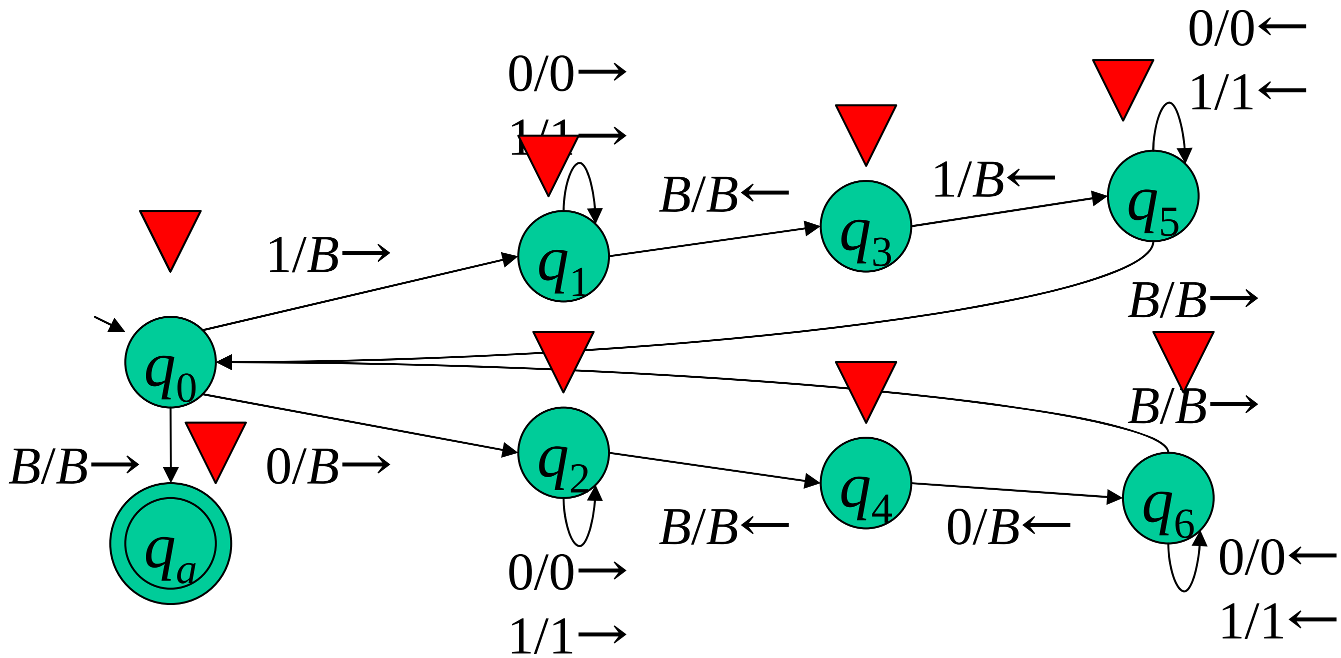
Ex) $L = \{\ ww^R \mid w \in \{0,1\}^*\ \}$ is accepted by TM M.

The computation of M for the input 1001 is:

$$q_0 1001 \vdash q_1 001 \vdash 0q_1 01 \vdash 00q_1 1 \vdash 001q_1 \vdash 00q_3 1$$

$$\vdash 0q_5 0 \vdash q_5 00 \vdash q_5 B00 \vdash q_0 00 \vdash q_2 0$$

$$\vdash 0q_2 \vdash q_4 0 \vdash q_6 \vdash q_0 \vdash q_a$$



8.2. Turing 機械とは

8.2.5. チューリング機械の受理言語

TM M によって受理される言語$L(M)$:

Mを入力wの元で動作させたとき、受理状態になる

形式的には…

$M=(Q,\Sigma,\Gamma,\delta,q_0,B,F)$ に対して、

$L(M)$ = { $w \in \Sigma^*$ | ある $p \in F$ が存在し、

$q_0 w \overset{*}{\vdash} \alpha p \beta$ ($\alpha, \beta \in \Gamma^*$)となる。}

★Mの停止性は問題にしていない

- とにかく途中で受理状態になれば受理する
- $L(M)$に入らない語は、受理状態にならなければよい。デッドロックでも無限ループでもよい。端的には停止しなくても良い。

8.2. Turing Machine

8.2.5. Language accepted by a TM

The language $L(M)$ accepted by a TM M:

M will be in an accepting state under input w

Formally…

For $M=(Q, \Sigma, \Gamma, \delta, q_0, B, F)$,

$$L(M) = \{ w \in \Sigma^* \mid \exists p \in F, q_0 w \overset{*}{\vdash} \alpha p \beta \ (\alpha, \beta \in \Gamma^*)\}$$

★We do not mind if M halts or not.

- Anyway, it accepts if it is in an accepting state.
- For a word not in $L(M)$, we only say that M never accepts. It is OK if it is in dead-lock or infinite-loop. Namely, it is OK if it does not stop.

8.2. Turing 機械とは

8.2.6. チューリング機械の停止性

[定義] TM M において $\delta(q,X)$ が未定義のとき、M は動作を停止すると定義する。

★$L(M)$の定義で、受理状態では TM は動作を停止するとしても定義される言語は変わらない。

★$L(M)$に属さない語 w の振る舞いはわからないことに注意する。(停止 or 無限ループ)

$w \in L$ と $w \notin L$ が非対称

帰納的可算言語: 上記の定義に基づく TM で

∪　受理できる言語

帰納的言語: $L(M)$ に属さない語 w に対しても TM Mが動作を停止する、という制限を加えた言語

8.2. Turing Machine

8.2.6. Halting property of TM

[Definition] For a TM M, if $\delta(q,X)$ is not defined, we define that M halts (namely, it stops computing).

★In the definition of $L(M)$, the language does not change if we define 'TM halts in an accepting state'.

★We do not mind for the word w not in $L(M)$. (Halt or infinite-loop)

$w \in L$ and $w \notin L$ are **not** symmetric.

Recursively enumerable language:

$\cup$ The set of languages accepted by above TMs

Recursive language: The set of languages accepted by TMs that always halt (especially, the words not in $L(M)$).

8. Turing 機械入門(1)

8.*. チューリング機械の意義

- 「計算」の数学的モデルとして
  - 「計算できる関数」が扱えるようになった
- 「計算する機械」のモデルとして
  - TMは万能性を持っている

    通常のフォン・ノイマン型計算機で計算できる関数は、すべてTMで計算できる。
  - 計算の効率を測るための尺度に使える

    アルゴリズムの効率は TM での時間量、領域量が計測のベースになっている。

8. Turing Machine

8.*. Meaning of a Turing machine

- Mathematical model of a ‘computation’
  - Computable function can be considered.
- Model for a ‘computer’
  - TM has universality.

    Every function computable by any von-Newman type computer can be computed by a universal Turing machine.
  - TM can be used as a measure for efficiency of a computation.

    The efficiency of an algorithm is evaluated by the time complexity and space complexity of a TM.