

情報数学

第4回 チューリング機械

萩野 達也

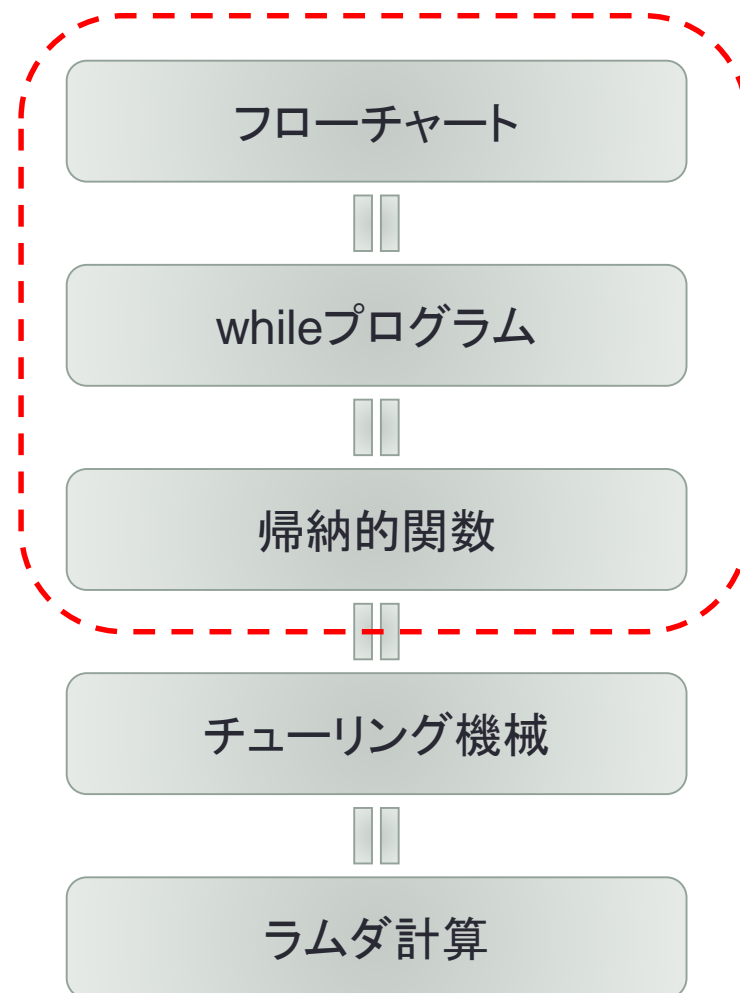
hagino@sfc.keio.ac.jp

Slides URL

<https://vu5.sfc.keio.ac.jp/slide/>

先週まで

- 計算
 - フローチャート
 - whileプログラム
 - 帰納的関数
 - 原始帰納的関数
 - 最小解演算子



有限状態オートマトン

- **有限状態オートマトン** (Finite State Automaton) M
 - Q 状態の集合 (**有限個**, 空でない)
 - Σ 入力される**文字**の集合 (**有限個**, 空でない)
 - δ **状態遷移**関数. $\delta: Q \times \Sigma \rightarrow Q$
 - q_0 **初期状態**. $q_0 \in \Sigma$
 - F **終了状態**の集合. Q の部分集合 (空でもよい)
- $M = (Q, \Sigma, \delta, q_0, F)$

有限状態オートマトン例(1)

- 0 と1 からなる文字列の中に1 が偶数回出現したかどうかを調べる有限状態オートマトン

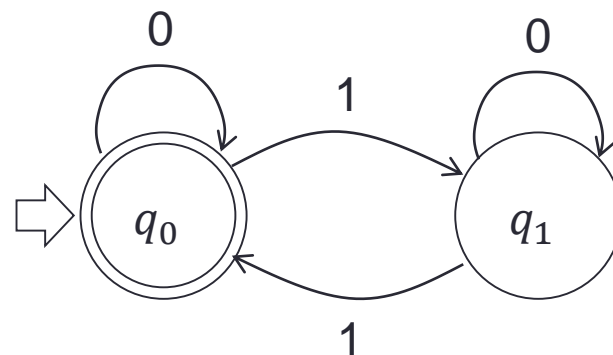
$$M_1 = (\{q_0, q_1\}, \{0, 1\}, \delta_1, q_0, \{q_0\})$$

- δ_1 を次のように定義

$$\delta_1: \{q_0, q_1\} \times \{0, 1\} \rightarrow \{q_0, q_1\}$$

$$\begin{cases} \delta_1(q_0, 0) = q_0 \\ \delta_1(q_0, 1) = q_1 \\ \delta_1(q_1, 0) = q_1 \\ \delta_1(q_1, 1) = q_0 \end{cases}$$

δ_1	0	1
q_0	q_0	q_1
q_1	q_1	q_0



状態遷移

- M_1 に 0101 を入力することを考える.

0. 初期状態 q_0

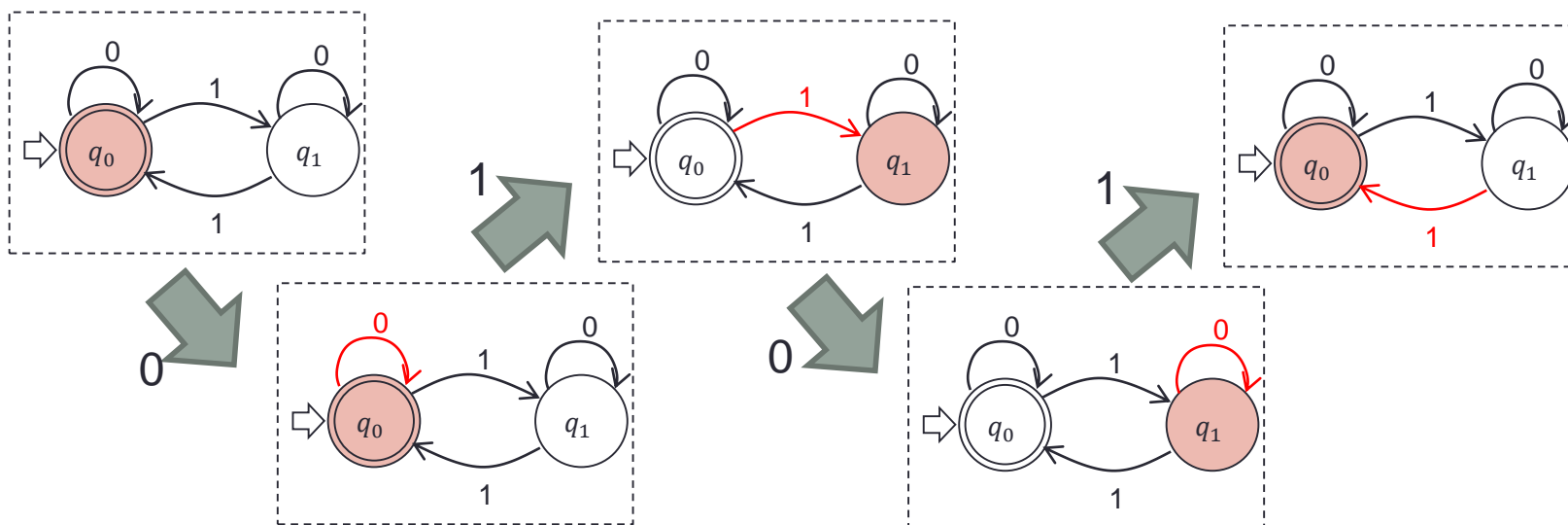
1. 入力 0 により状態は $\delta_1(q_0, 0) = q_0$

2. 入力 1 により状態は $\delta_1(q_0, 1) = q_1$

3. 入力 0 により状態は $\delta_1(q_1, 0) = q_1$

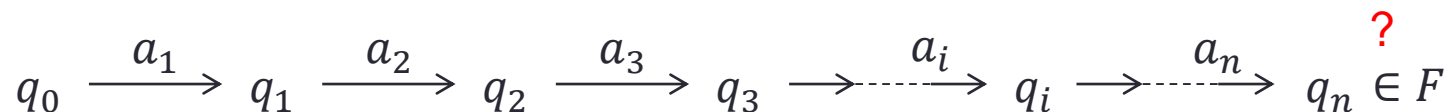
4. 入力 1 により状態は $\delta_1(q_1, 1) = q_0$

- $q_0 \in F$ なので M_1 は 0101 を **受理** (accept) する.



状態遷移(一般)

- 入力の Σ の文字を受け取るごとに状態が遷移する.
 0. 初期状態は常に q_0
 1. 最初の文字 a_1 を受け取ると, 状態は $\delta(q_0, a_1) = q_1$ に遷移
 2. 2番目の文字 a_2 を受け取ると, 状態は $\delta(q_1, a_2) = q_2$ に遷移
 3. 3番目の文字 a_3 を受け取ると, 状態は $\delta(q_2, a_3) = q_3$ に遷移
 -
 - i . i 番目の文字 a_i を受け取ると, 状態は $\delta(q_{i-1}, a_i) = q_i$ に遷移
 -
 - n . n 番目の文字 a_n を受け取ると, 状態は $\delta(q_{n-1}, a_n) = q_n$ に遷移
- 有限状態オートマトン M は $q_n \in F$ であるとき, 文字列 $a_1 a_2 \cdots a_n$ を **受理**する.



受理される言語

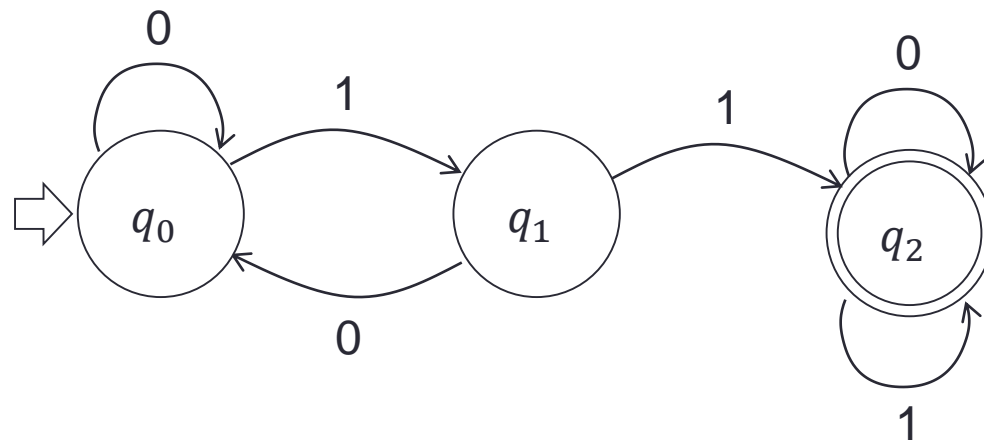
- 状態遷移関数 δ を文字列に拡張する.
 - $\delta(q, a_1 a_2 a_3 \cdots a_n) = \delta(\cdots \delta(\delta(\delta(q, a_1), a_2), a_3) \cdots, a_n)$
 - $\delta(q, \epsilon) = q$ただし. ϵ は空文字列を表す.
- 有限状態オートマト M は次の時に, 文字列 $a_1 a_2 \cdots a_n$ を**受理**する.
 - $\delta(q_0, a_1 a_2 \cdots a_n) \in F$
- 有限状態オートマトン $M = (Q, \Sigma, \delta, q_0, F)$ が**受理**する文字列の集合を, M が受理する**言語**という.
 - $L(M) = \{x \in \Sigma^* \mid \delta(q_0, x) \in F\}$

有限状態オートマトン例(2)

- 次の有限状態オートマトンの状態遷移図を書きなさい。

- $M_2 = (\{q_0, q_1, q_2\}, \{0, 1\}, \delta_2, q_0, \{q_2\})$

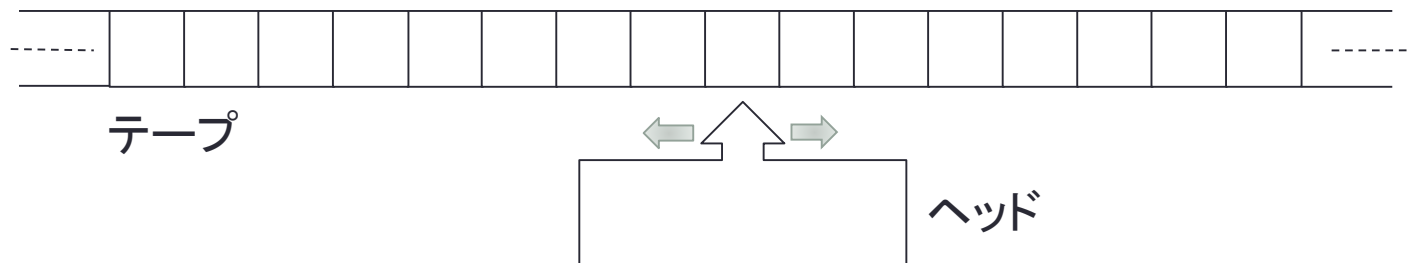
δ_2	0	1
q_0	q_0	q_1
q_1	q_0	q_2
q_2	q_2	q_2



- このオートマトンが受理する言語 $L(M_2)$ は何ですか？

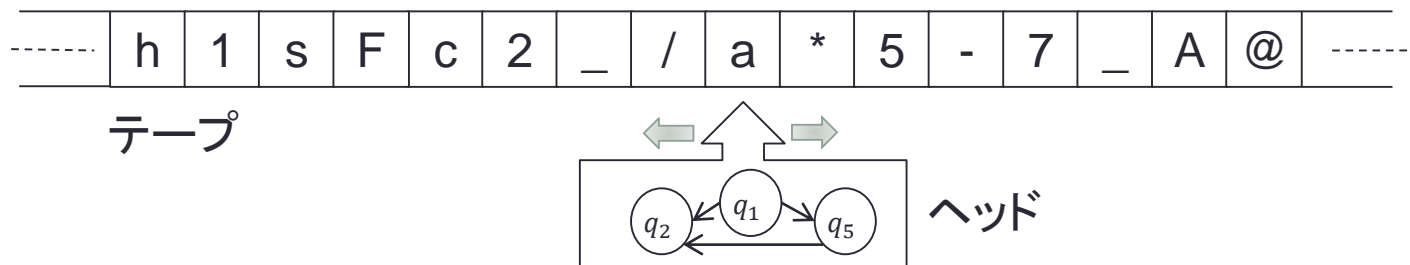
チューリング機械

- アラン・チューリング (Alan Turing)
 - イギリスの数学者 (1912 年6月23日 ~ 1954年6月7日)
 - 「On Computable Numbers, with an Application to the Entscheidungsproblem」1936年5月28日
 - Entscheidungsproblem = 決定問題
 - The Entscheidungsproblem = 「与えられた論理式が証明可能かどうかを決定する手法はあるか」ヒルベルトが1928年に出した問題.
- チューリング機械
 - 左右無限に長いテープ
 - テープ上のデータを読み書きし, 左右に動くヘッド



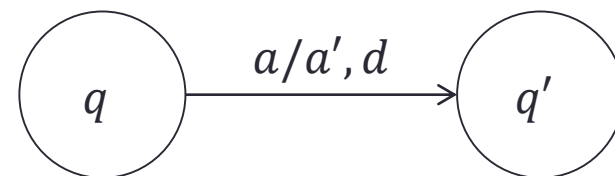
テープとヘッド

- テープ
 - 左右に無限に長いテープが1本ある.
 - テープはマスに分けられている.
 - マスには記号(文字)が書かれている.
 - 記号が書かれていない所は空白(という記号)が書かれている.
- ヘッド
 - テープの情報を読み書きするヘッドが1つある.
 - ヘッドはテープのどこかのマス上にあり, そのマスの記号を読み書きできる.
 - 内部に状態を持つ.
 - マスの記号と現在の状態から次の状態が決まる.
 - ヘッドは右が左に1つずつ移動する.



形式的定義

- チューリング機械 M は3つの (A, Q, T) からなる:
 - **テープ記号**の有限集合 $A = \{a_0, a_1, \dots, a_{m-1}\}$
 - a_0 を空白を表す記号とし「_」を使う.
 - **ヘッドの内部状態**の有限集合 $Q = \{q_0, q_1, \dots, q_{l-1}\}$
 - q_1 を初期状態,
 - q_0 を終了(停止)状態とする.
 - **遷移を表す関数** $T: Q \times A \rightarrow Q \times A \times \{L, R, N\}$
 - 内部状態が q で, 現在のテープ記号が a のとき,
 - $T(q, a) = (q', a', d)$ であれば,
 - 次の状態は q' となり,
 - テープ記号は a から a' に書き換え,
 - $d = L$ ならば, ヘッドは左に1つ移動し,
 - $d = R$ ならば, ヘッドは右に1つ移動し,
 - $d = N$ ならば, ヘッドは移動せずに同じ場所にいる.

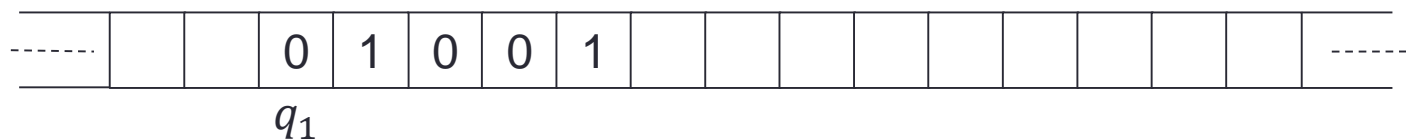
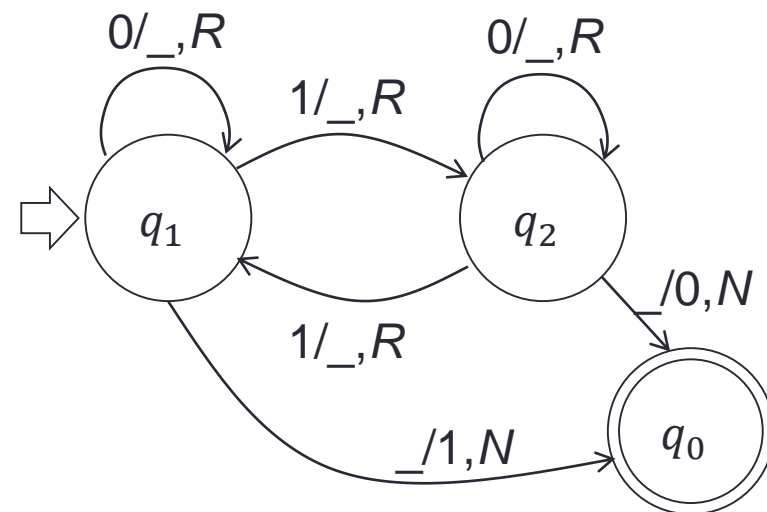


チューリング機械の例(1)

- 偶数個の「1」がテープ上に書かれていれば「1」を書き, そうでなければ「0」を書いて停止する(もとの記号はテープから消す).

$$M_3 = (\{_, 0, 1\}, \{q_0, q_1, q_2\}, T_3)$$

T_3	_	0	1
q_1	$(q_0, 1, N)$	$(q_1, _, R)$	$(q_2, _, R)$
q_2	$(q_0, 0, N)$	$(q_2, _, R)$	$(q_1, _, R)$

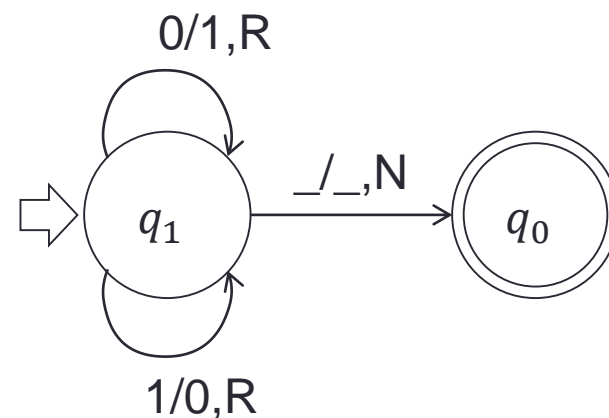


チューリング機械の例(2)

- テープ上の「1」と「0」を反転させるチューリング機械を作りなさい。

$$M_4 = (\{_, 0, 1\}, \{q_0, q_1\}, T_4)$$

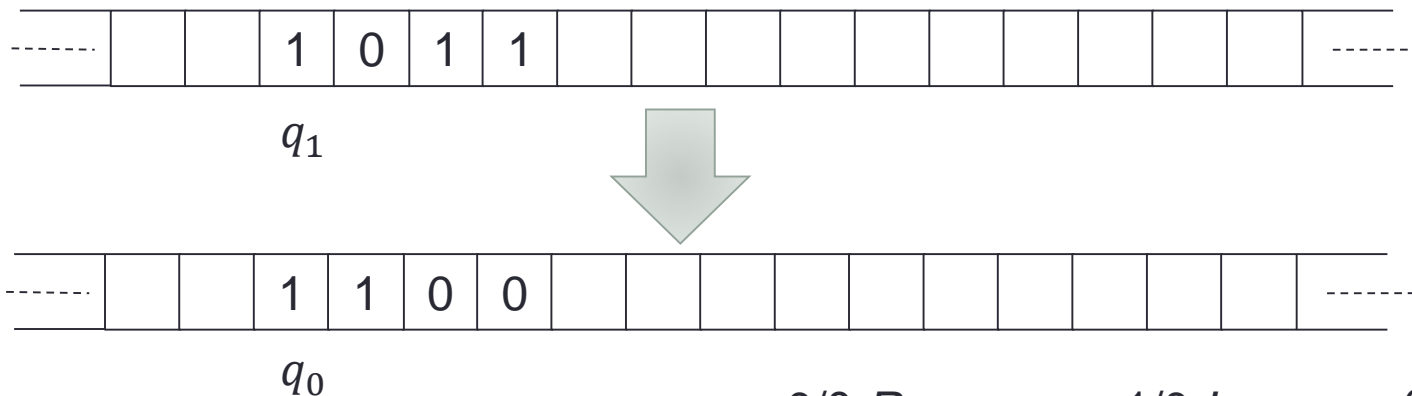
T_4	$_$	0	1
q_1	(, ,)	(, ,)	(, ,)



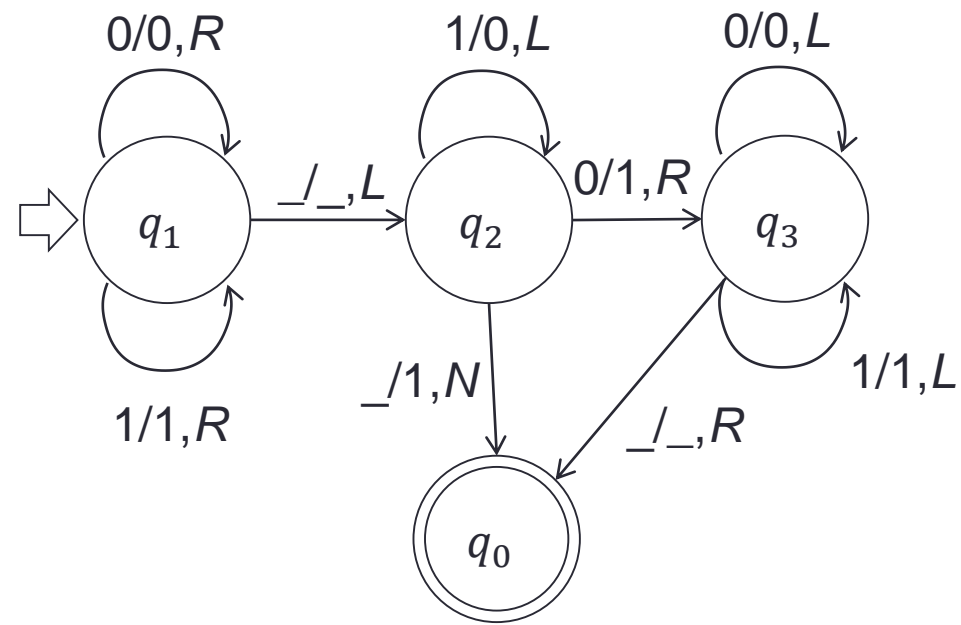
チューリング機械の例(3)

- テープ上の二進数の数字に1を加えるチューリング機械を作りなさい。

$$M_5 = (\{_, 0, 1\}, \{q_0, q_1, q_2, \dots\}, T_5)$$



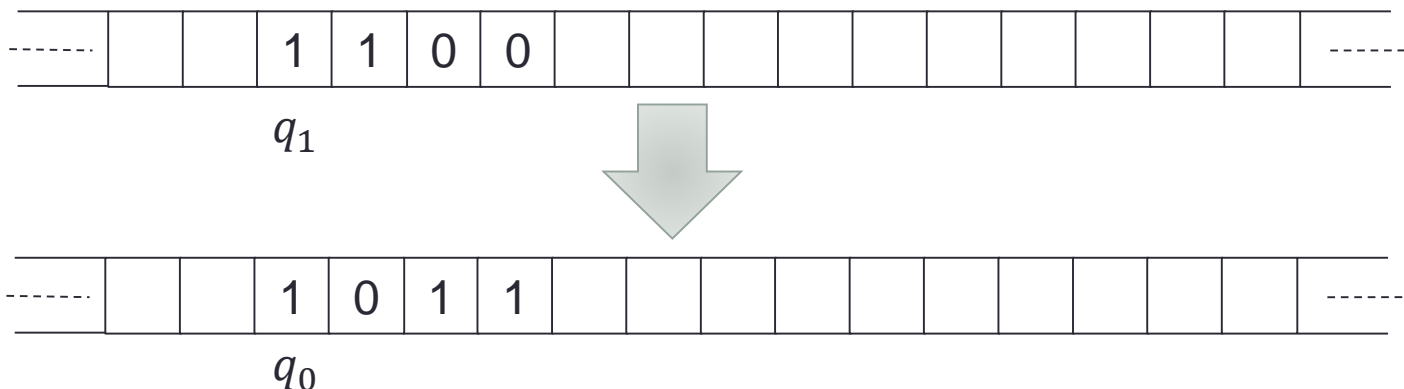
T_5	_	0	1
q_1	$(q_2, _, L)$	$(q_1, 0, R)$	$(q_1, 1, R)$
q_2	$(q_0, 1, N)$	$(q_3, 1, L)$	$(q_2, 0, L)$
q_3	$(q_0, _, R)$	$(q_3, 0, L)$	$(q_3, 1, L)$



チューリング機械の例(4)

- テープ上の二進数の数字から1を引くチューリング機械を作りなさい.

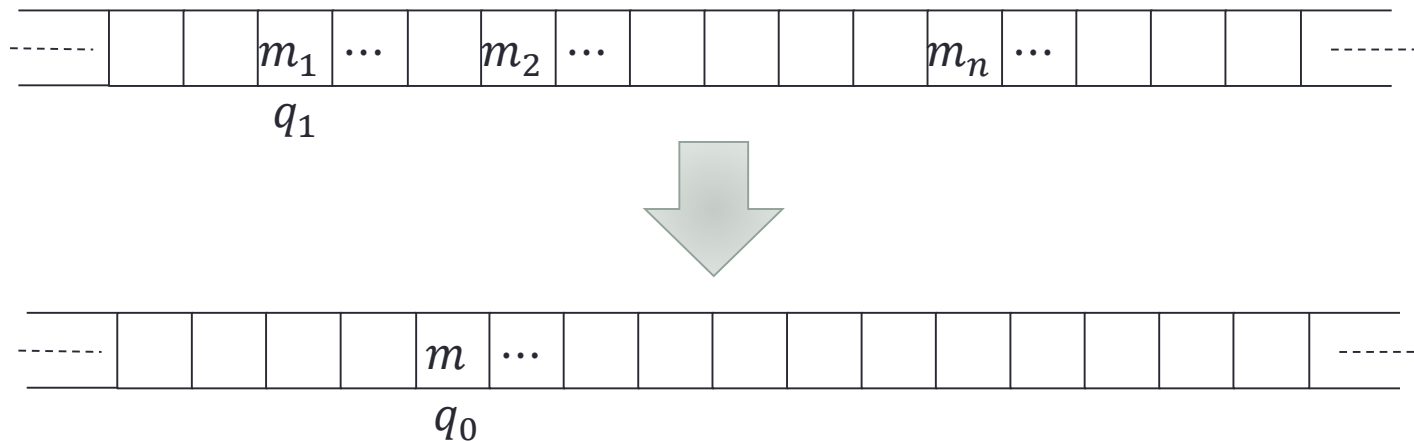
$$M_6 = (\{ _ , 0, 1 \}, \{ q_0, q_1, q_2, \dots \}, T_6)$$



T_6	_	0	1
q_1	(, ,)	(, ,)	(, ,)
q_2	(, ,)	(, ,)	(, ,)
q_3	(, ,)	(, ,)	(, ,)

チューリング機械の計算

- チューリング機械 M が関数 $f: N^n \rightarrow N$ を計算するとは,
 - 数字 m_1, m_2, \dots, m_n をテープ上に空白で区切って書いておく.
 - ヘッドを左端にセットして M を開始する.
 - M が停止した時に, テープ上に $f(m_1, m_2, \dots, m_n)$ の値が書かれる.

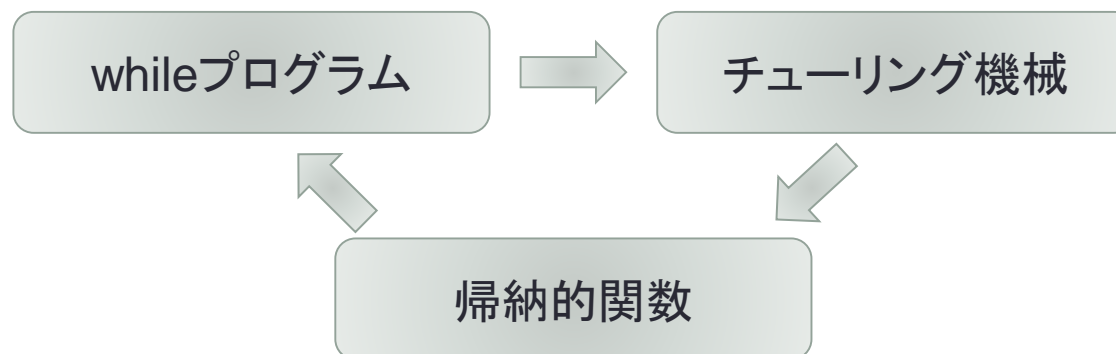


チューリング機械とプログラム

- チューリング機械は一般には停止しない.
 - 計算する関数は全域的ではなく部分的

• 定理

- チューリング機械が計算する関数 $f: N^n \rightarrow N$ はwhileプログラムで計算できる.
- 帰納的関数 $f: N^n \rightarrow N$ はチューリング機械で計算することができる.



まとめ

- 有限状態オートマトン
 - 有限個の状態の集合
 - 状態遷移関数
- チューリング機械
 - 無限のテープとヘッド
- 計算
 - フローチャート
 - whileプログラム
 - 帰納的関数
 - チューリング機械