MATHEMATICS FOR INFORMATION SCIENCE
NO.4 TURING MACHINE

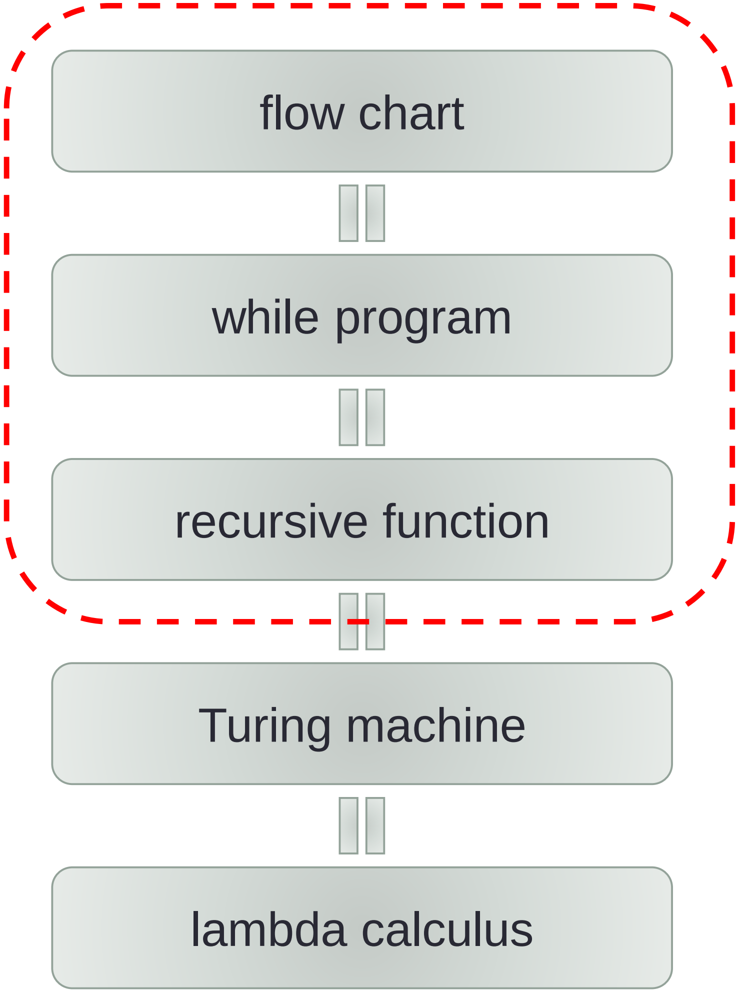
Tatsuya Hagino
hagino@sfc.keio.ac.jp

Slides URL

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

So far

- Computation
  - flow chart program
  - while program
  - recursive function
    - primitive recursive function
    - minimization operator



Finite State Automata

- A Finite State Automaton $M = (Q, \Sigma, \delta, q_0, F)$
  - Q: a finite, non-empty set of states
  - Σ: the input alphabet (a finite, non-empty set of symbols)
  - δ: a state-transition function, $\delta: Q \times \Sigma \rightarrow Q$
  - q_0: an initial state, an element in Q
  - F: a set of final states, a (possibly empty) subset of Q

FA Example (1)

- An automaton which checks whether '1' appears even number of times in a string of '0' and '1'.

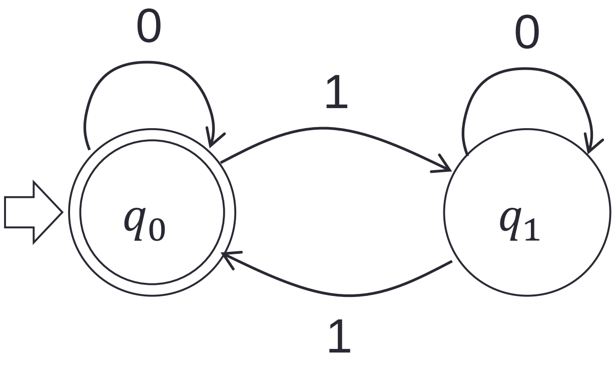
$$M_1 = (\{q_0, q_1\}, \{0,1\}, \delta_1, q_0, \{q_0\})$$

- Define δ_1 as follows:

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

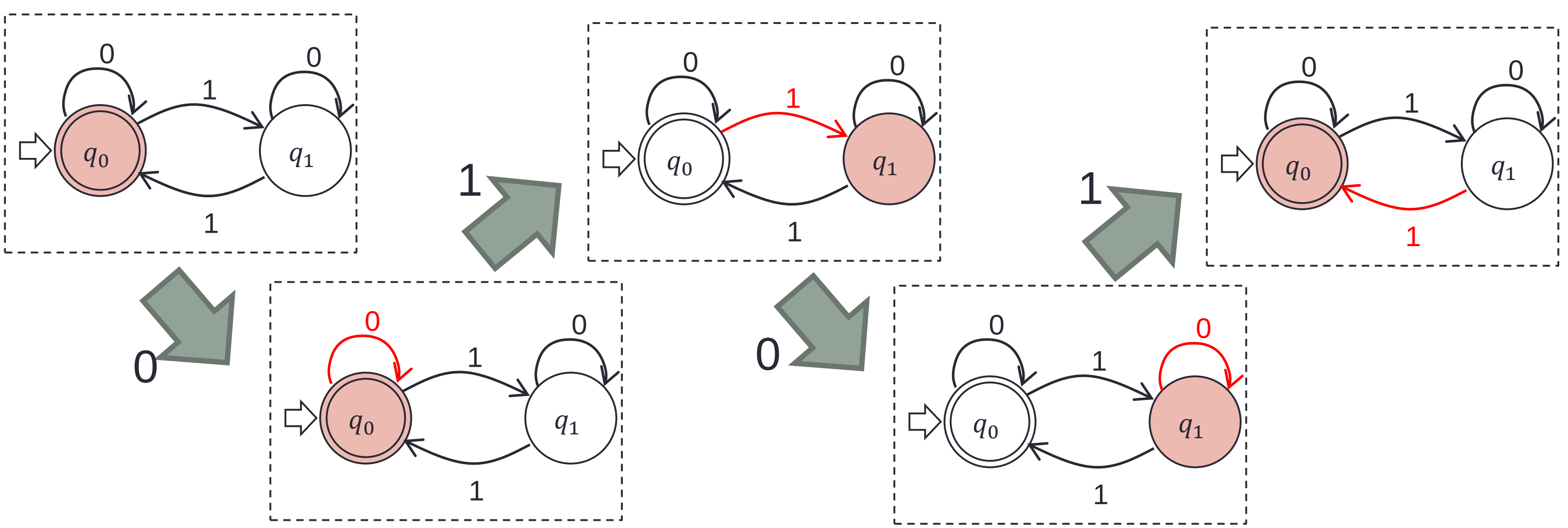
$$\begin{cases} \delta_1(q_0, 0) = q_0 \\ \delta_1(q_0, 1) = q_1 \\ \delta_1(q_1, 0) = q_1 \\ \delta_1(q_1, 1) = q_0 \end{cases}$$

δ_1	0	1
q_0	q_0	q_1
q_1	q_1	q_0



State Transition

- Input "0101" to M_1
  0. The initial state is q_0
  1. Input 0 moves to $\delta_1(q_0, 0) = q_0$
  2. Input 1 moves to $\delta_1(q_0, 1) = q_1$
  3. Input 0 moves to $\delta_1(q_1, 0) = q_1$
  4. Input 1 moves to $\delta_1(q_1, 1) = q_0$
- The automaton M_1 accepts `0101' because $q_0 \in F$.



State Transition in General

- Change state according to input symbols in Σ
  0. The initial state is always q_0
  1. After receiving the first symbol a_1, the state changes to $\delta(q_0, a_1) = q_1$
  2. After receiving the second symbol a_2, the state changes to $\delta(q_1, a_2) = q_2$
  3. After receiving the third symbol a_3, the state changes to $\delta(q_2, a_3) = q_3$

     . . . .

  i. After receiving the i th symbol a_i, the state changes to $\delta(q_{i-1}, a_i) = q_i$

     . . . .

  n. After receiving the n th symbol a_n, the state changes to $\delta(q_{n-1}, a_n) = q_n$

- M accepts $a_1 a_2 \cdots a_n$ when $q_n \in F$

$$q_0 \xrightarrow{a_1} q_1 \xrightarrow{a_2} q_2 \xrightarrow{a_3} q_3 \longrightarrow\cdots\xrightarrow{a_i} q_i \longrightarrow\cdots\xrightarrow{a_n} q_n \overset{?}{\in} F$$

Accepted Language

- Extend δ to a sequence of symbols:
  - $\delta(q, a_1 a_2 a_3 \cdots a_n) = \delta(\cdots \delta(\delta(\delta(q, a_1), a_2), a_3) \cdots, a_n)$
  - $\delta(q, \epsilon) = q$

  where ϵ represents the empty sequence.

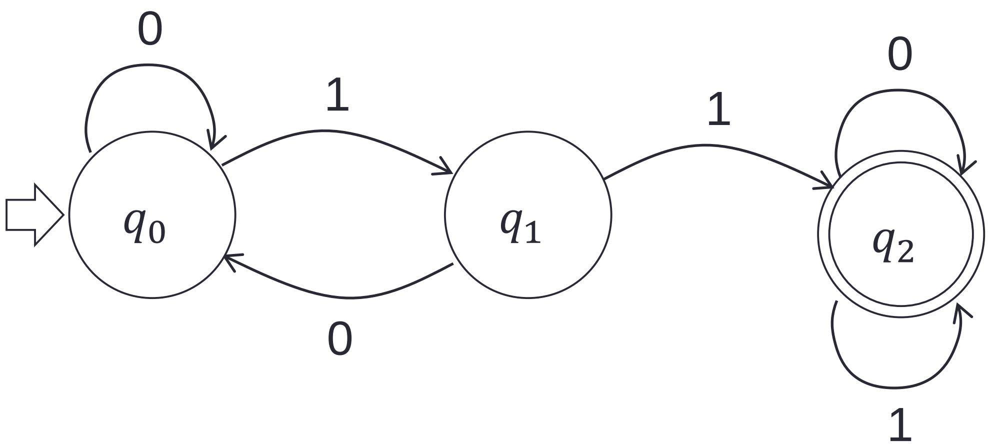
- M accepts $a_1 a_2 \cdots a_n$ when
  - $\delta(q_0, a_1 a_2 \cdots a_n) \in F$

- The language which $M = (Q, \Sigma, \delta, q_0, F)$ accepts can be defined as follows:
  - $L(M) = \{x \in \Sigma^* \mid \delta(q_0, x) \in F\}$

FA Example (2)

- Write the state diagram of the following machine.
  - $M_2 = (\{q_0, q_1, q_2\}, \{0,1\}, \delta_2, q_0, \{q_2\})$

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

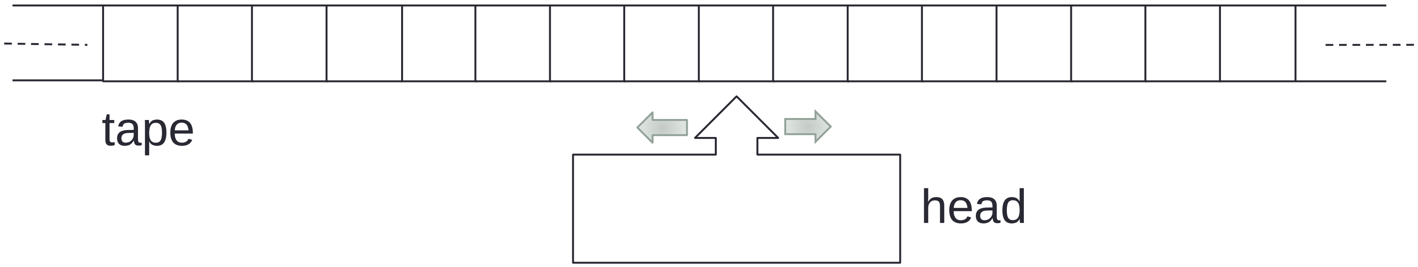


- What is the language $L(M_2)$ which M_2 accepts?
  - Accept when input ....

Turing Machine

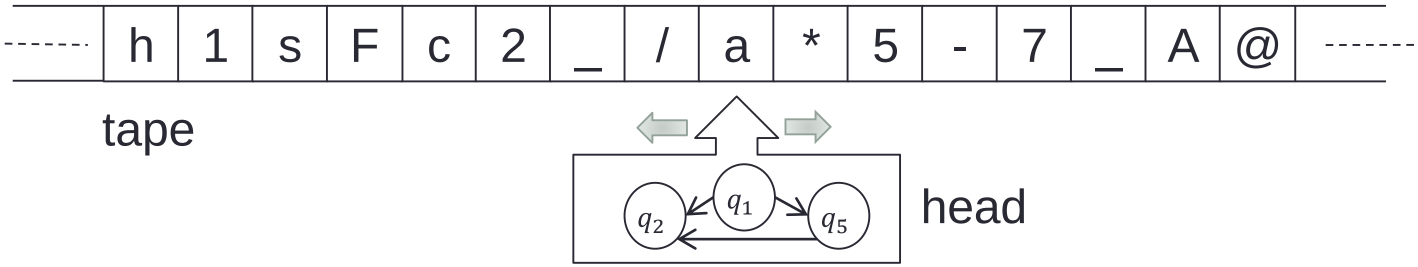
- Alan Turing
  - British Mathematician (1912/6/23〜1954/6/7)
  - "On Computable Numbers, with an Application to the Entscheidungsproblem", 1936/5/28
    - Entscheidungsproblem = decision problem
    - The Entscheidungsproblem = "ask for an algorithm that takes as input a statement of a first-order logic and answers "Yes" or "No" according to whether the statement is valid" by David Hilbert in 1928.

- Turing Machine
  - an infinite length tape
  - a head which can read data on the tape and moves left and right



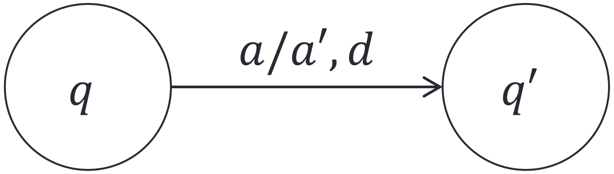
Tape and Head

- Tape
  - One tape width infinite length for left and right
  - The tape is divided into cells.
  - Each cell holds a symbol (an alphabet or a blank symbol).
- Head
  - One head
  - The head is on top of one cell.
  - The head can read and write the symbol in the cell.
  - The head can move left or right one cell at a time.
  - The head has a state.
  - The next state is determined by the current state and the symbol in the cell.



Formal Definition

- A Turing machine M consists of the following three things:
  - A finite set of tape symbols $A = \{a_0, a_1, \cdots, a_{m-1}\}$
    - Let a_0 be the special symbol '_' for blank.
  - A finite set of states $Q = \{q_0, q_1, \cdots, q_{l-1}\}$
    - q_1 is the initial state and q_0 is the final state.
  - A transition function $T: Q \times A \to Q \times A \times \{L, R, N\}$
    - Let q be the current state, and a be the tape symbol.
    - If $T(q, a) = (q', a', d)$,
      - The next state is q',
      - The tape symbol is rewritten from a to a',
      - If $d = L$, the head moves to left one cell,
      - If $d = R$, the head moves to right one cell, and
      - If $d = N$, the head does not move.

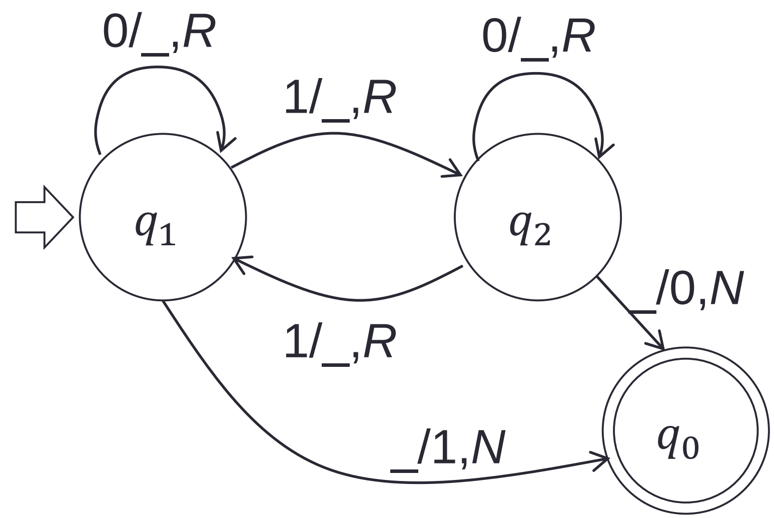


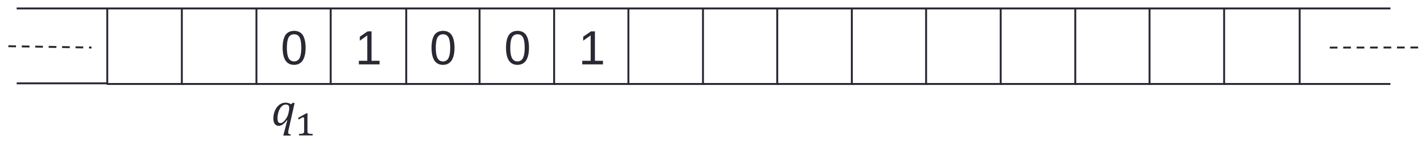
Turing Machine Example (1)

- The following Turing machine writes '1' when there is even number of '1's and '0' otherwise.

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

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



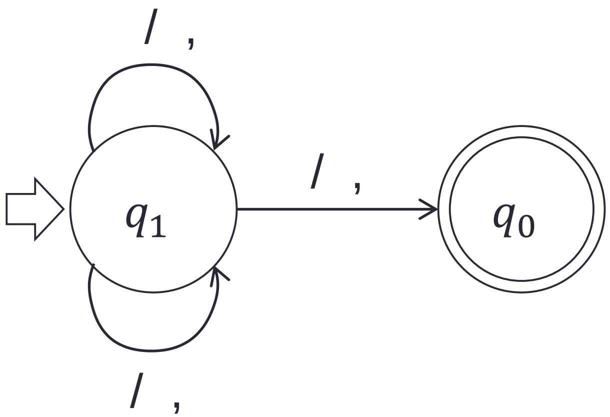


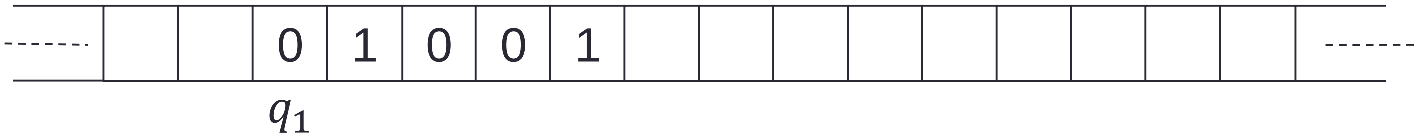
Turing Machine Example (2)

- Write a Turing machine which reverse '1' and '0' (i.e. replace '1' with '0', and replace '0' with '1').

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

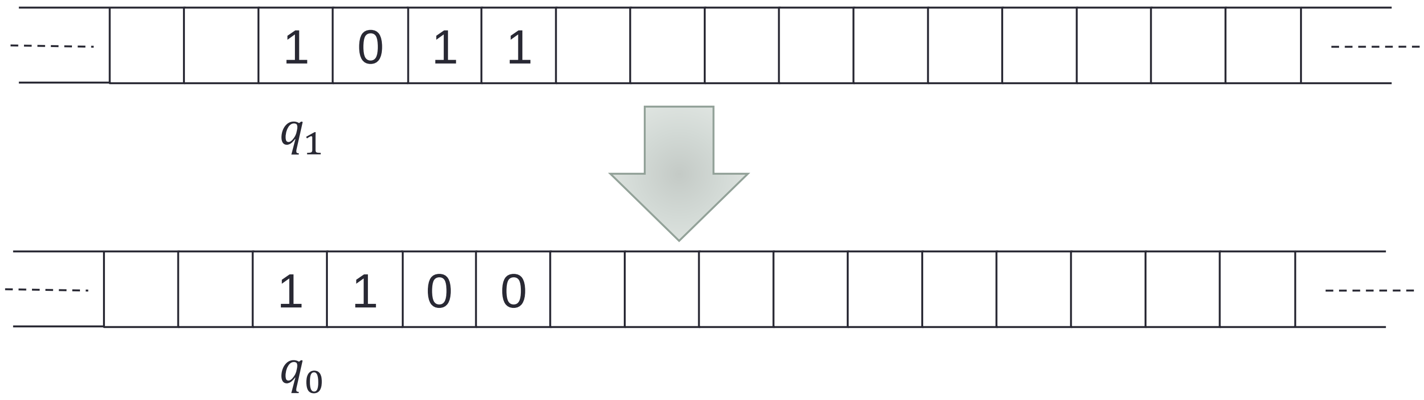
T_4	_	0	1
q_1	(, ,)	(, ,)	(, ,)





Turing Machine Example (3)

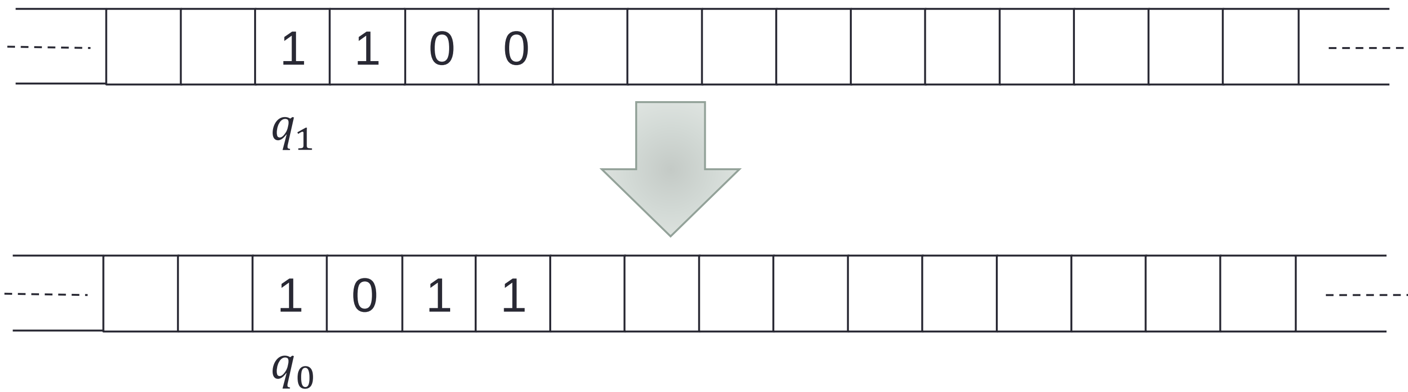
- Write a Turing machine $M_5 = (\{\_, 0,1\}, \{q_0, q_1, q_2, \cdots\}, T_5)$ which adds one to the binary number written on the tape.



T_5	_	0	1
q_1	(, ,)	(, ,)	(, ,)
q_2	(, ,)	(, ,)	(, ,)
q_3	(, ,)	(, ,)	(, ,)

Turing Machine Example (4)

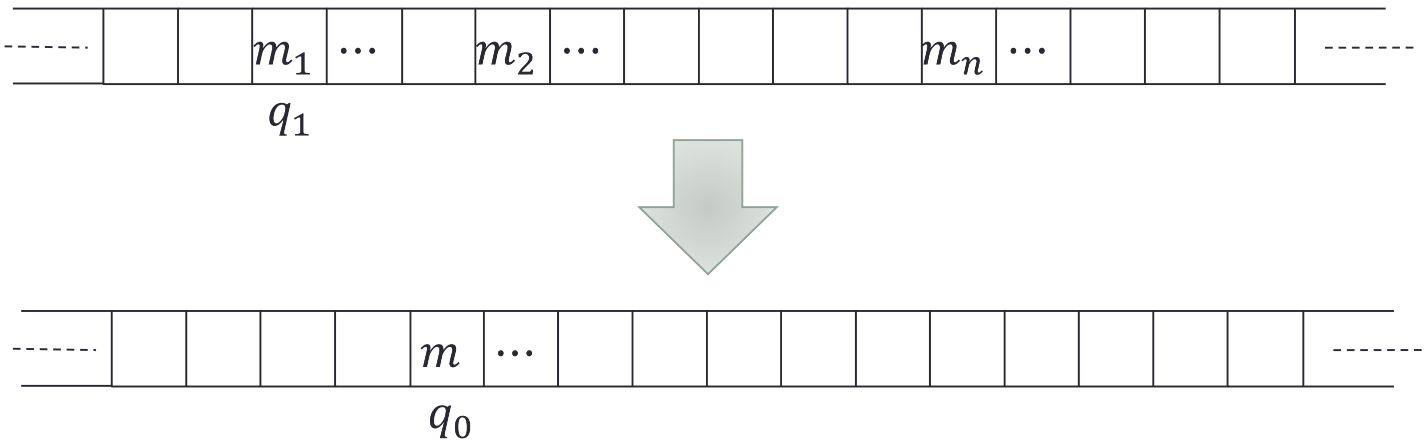
- Write a Turing machine $M_6 = (\{\_, 0,1\}, \{q_0, q_1, q_2, \cdots\}, T_6)$ which subtracts one from the given binary number on the tape.



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

Computation

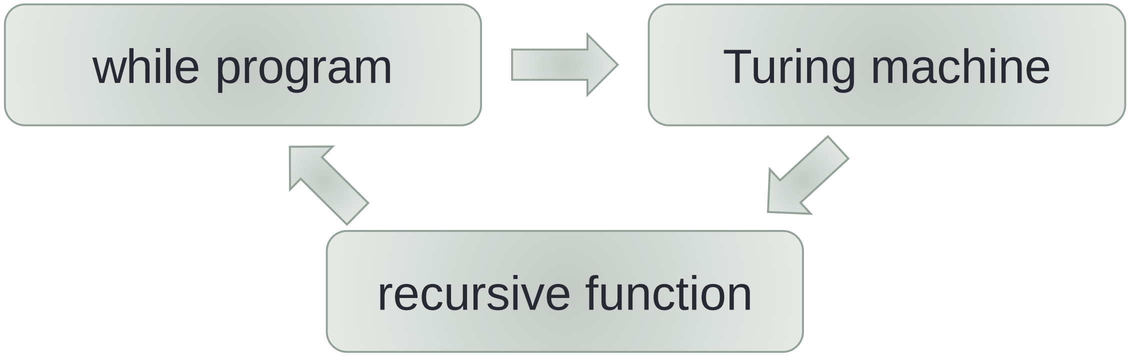
- A Turing machine M computes $f: N^n \rightarrow N$ when:
  - Place $m_1, m_2, \cdots, m_n$ on the tape with decimal numbers separated with a blank
  - Start M with the head at the leftmost number position.
  - When M terminates, the number at the head is the decimal number of $f(m_1, m_2, \cdots, m_n)$.



Computation and Program

- A Turing machine may not terminate.
  - The function it computes is not total, but partial.

- **Theorem**
  - If a Turing machine can compute $f: N^n \to N$, it can be computed by a while program.
  - If $f: N^n \to N$ is a recursive function, there is a Turing machine which can compute the same function.



Summary

- Finite State Automata
  - a finite set of states
  - a state transition function
- Turing Machine
  - an infinite tape and a head
- Computation
  - flow chart program
  - while program
  - recursive function
  - Turing machine