Chap. 9
Turing Machines 최상위 automata.

Agenda of Chapter 9

What can we say about the most powerful automata and the limits of computation?

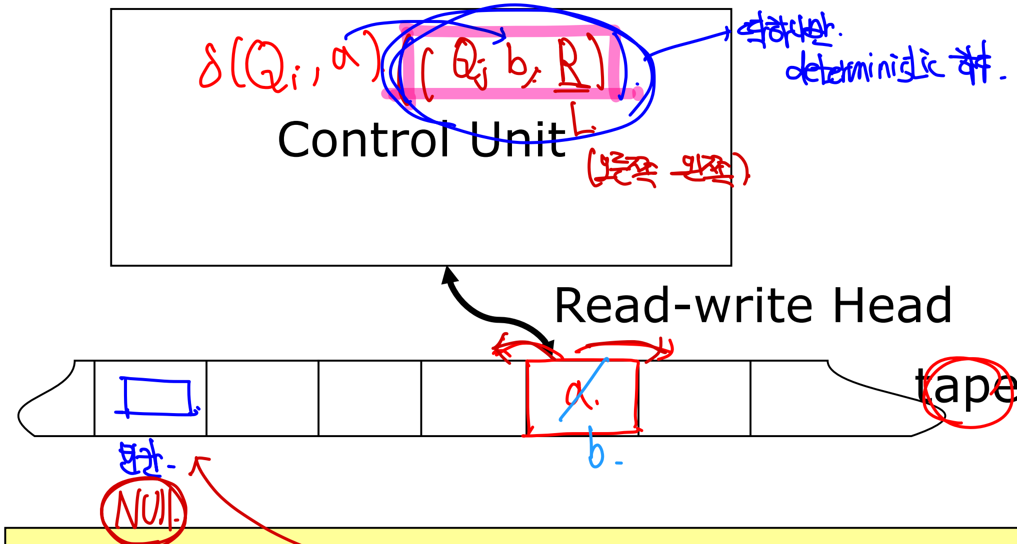
How to define the idea of a mechanical or algorithmic computation?

- The Standard Turing Machine
  - Turing machines as language accepters
  - Turing machines as transducers
- Combining Turing Machines for Complicated Tasks
- Turing's Thesis.

Hyeyoung Park, Dept. Comp. Sci.

Definition of a Turing Machine(1/5)

- Schematic representation of Turing machine



- a symbol on tape.
- current state

movement

- rewrite the current symbol
- change the state
- move read-write head.
- stops at halt state.

Definition] Turing machine M=(Q, Σ, Γ, δ, q_0, ▭ , F)

- Q : set of internal states
- Σ : input alphabet (Assume Σ ⊆ Γ-{▭})
- Γ: tape alphabet (a finite set of symbols)
- δ : Q×Γ → Q×Γ×{L,R} (partial function)
- ▭ ∈ Γ : a special symbol called blank

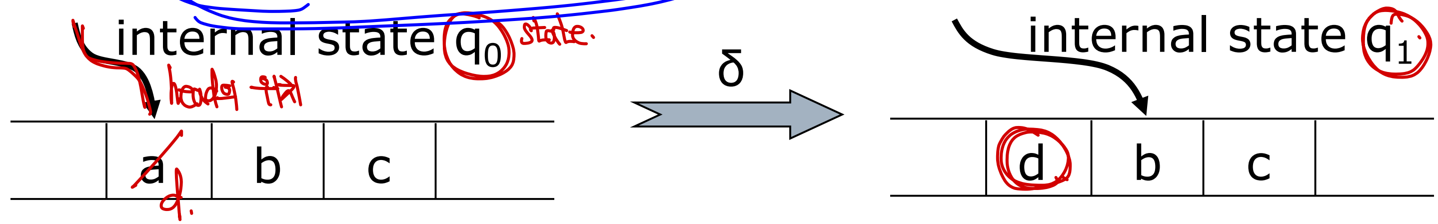
tape alphabet.

→ tape alphabet 안에 있는 symbol

"dead configuration이 있다."

Definition of a Turing Machine(2/5)

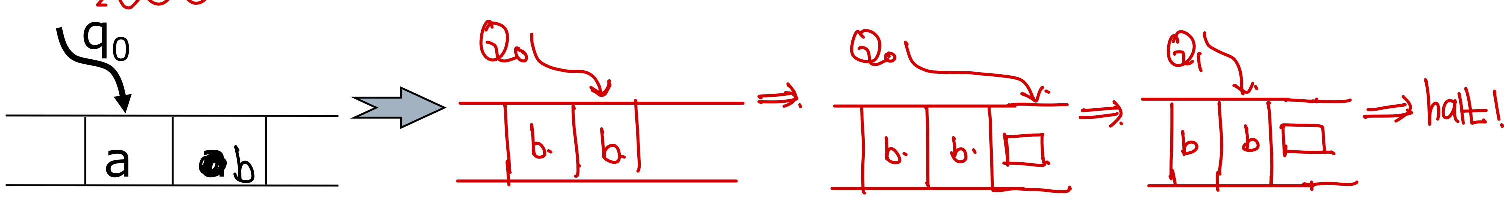
Ex9.1] $\delta(q_0, a) = (q_1, d, R)$



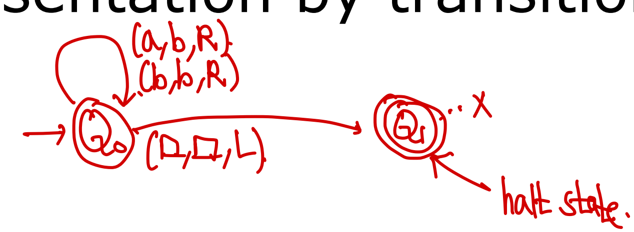
Ex9.2] Turing machine with a halt state

$M=(\{q_0,q_1\}, \{a,b\}, \{a,b, \square\}, \delta, q_0, \square, \{q_1\})$ $\delta(q_1, b) \rightarrow$ 정의 안됨.

$\delta(q_0,a) = (q_0,b,R)$, $\delta(q_0,b) = (q_0,b,R)$, $\delta(q_0,\square) = (q_1,\square,L)$



[Representation by transition graph]



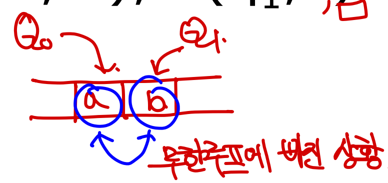
무조건 시험.

Definition of a Turing Machine(3/5)

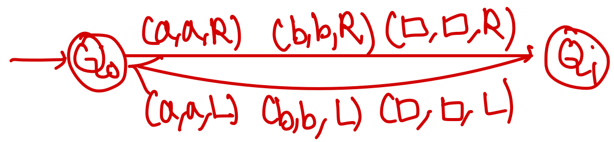
Ex 9.3] Turing machine in an **infinite loop**

$M=(\{q_0,q_1\}, \{a,b\}, \{a,b, \square\}, \delta, q_0, \square, \{\})$

$\delta(q_0,a) = (q_1,a, R)$, $\delta(q_0,b) = (q_1,b,R)$, $\delta(q_0,\square) = (q_1,\square,R)$,
$\delta(q_1,a) = (q_0,a,L)$, $\delta(q_1,b) = (q_0,b, L)$, $\delta(q_1,$ ƀ $)$ $,\square$ $= (q_0,\square, L)$



Transition graph:

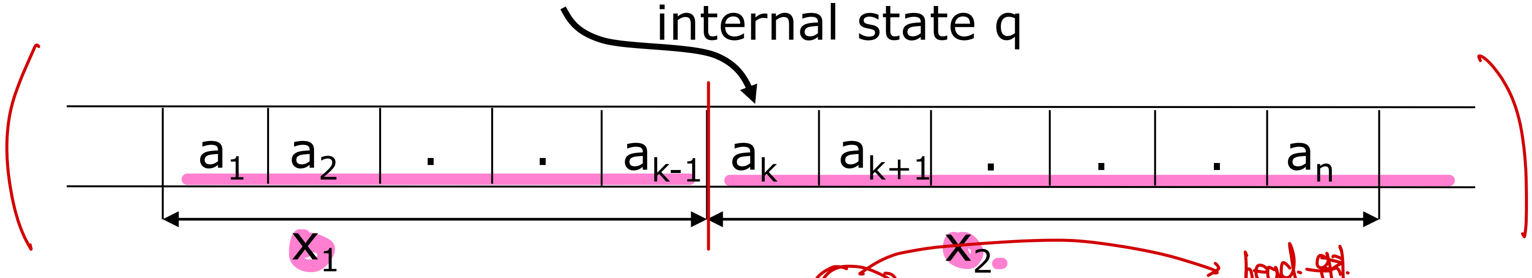


If the tape initially contains ab…, with read-write head on a

- Main features of standard Turing machine
  - An unbounded tape in both directions
  - Deterministic (i.e., at most one move for each configuration)
  - No special input file and no output device.
    - Input : contents of the tape at the initial time
    - Output : contents of the tape when the machine halts.

Definition of a Turing Machine(4/5)

- Instantaneous description for Turing machine configuration



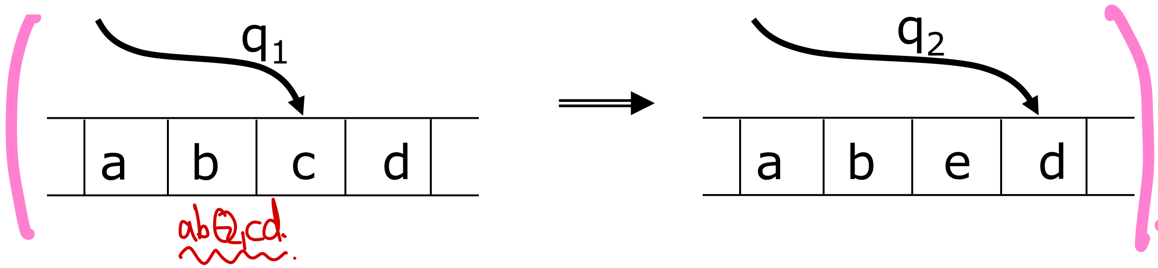
$x_1 q x_2 = a_1 a_2 \ldots a_{k-1} q a_k a_{k+1} \ldots a_n$

head 위치
현재 state.

- Instantaneous description for a movement

$abq_1cd \vdash abeq_2d$

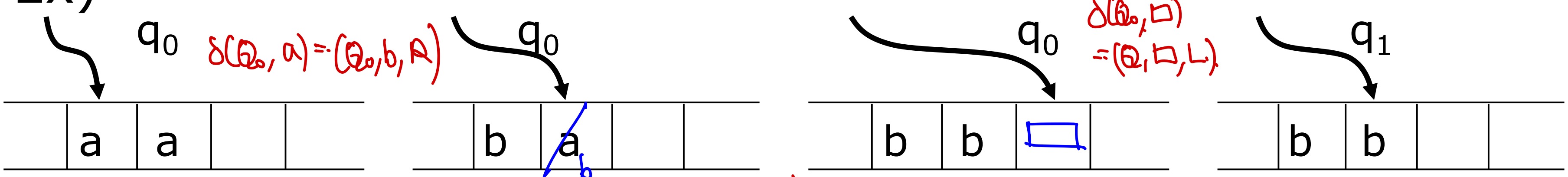
$\delta(q_1, c) = (q_2, e, R)$.



- Ex)

$\delta(q_0, a) = (q_0, b, R)$

$\delta(q_0, \square) = (q_1, \square, L)$



$q_0aa \vdash bq_0a \vdash bbq_0\square \vdash bq_1b\square$

써도 되고 안써도 됨.

Definition of a Turing Machine(5/5)

- **Computation** of a Turing machine
  - Let $M=(Q, \Sigma, \Gamma, \delta, q_0, \square, F)$ be a Truing machine
  - A move $a_1a_2...a_{k-1}\overset{q_2}{\cancel{q_1}}\overset{b}{\cancel{a_k}}a_{k+1}...a_n \vdash a_1a_2...a_{k-1}bq_2a_{k+1}...a_n$ is possible iff $\delta(q_1, a_k)=(q_2,b,R)$.
  - A move $a_1a_2...a_{k-1}\overset{q_2}{\cancel{q_1}}\overset{b}{\cancel{a_k}}a_{k+1}...a_n \vdash a_1a_2...q_2a_{k-1}ba_{k+1}...a_n$ is possible iff $\delta(q_1, a_k)=(q_2,b,L)$.
  - M is said to halt starting from initial configuration $x_1q_ix_2$ if $x_1q_ix_2 \vdash^* y_1q_jay_2$ for any q_j and a, for $\delta(q_j, a)$ is undefined.
  - **Computation** : sequence of configurations leading to a halt sate.

- Notation for endless loop.
  - $x_1qx_2 \vdash^* \infty$
  - Staring from the initial configuration x_1qx_2, the machine never halts.

Turing Machines as Language Accepters (1/3)

- Turing machine accepting w
  - w is written on tape, with blanks for unused portions. $q_0 w$
  - Start in q_0
  - Read-write head positions on the leftmost symbol of w.
  - Halts after a sequence of moves.
  - Enters a final states

$q_0 w \vdash^* x_1 q_f x_2$, $q_f \in F$ → 중요하지 않다.

(accept되는 string)

Definition] Language accepted by M$=(Q, \Sigma, \Gamma, \delta, q_0, \square, F)$

(halt되면서 $q_f \in F$)

$L(M)=\{w \in \Sigma^+ \mid q_0 w \vdash^* x_1 q_f x_2 \text{ for some } q_f \in F,\ x_1, x_2 \in \Gamma^*\}$ → "아무거나"

- Input w is written on the tape with blank on either side
- When w is not in L(M)
  - ① halts in a nonfinal state → final 아닌 곳에서 halt 되면 reject.
  - ② Enters an infinite loop and never halt.

→ 2가지 경우.

→ reject는 X ∞

Hyeyoung Park, Dept. Comp. Sci.

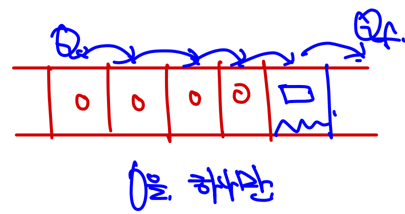
Turing Machines as Language Accepters (2/3)

Ex9.6] $\Sigma=\{0,1\}$ Turing machine accepting 00^*

– $M = (\{q_0, q_f\}, \{0,1\}, \{0,1,\square\}, \delta, q_0, \square, \{q_f\})$

[q_0] continue to head movement during input 0

$\boxed{\delta(q_0,0) = (q_0,0,R)}$



[$q_0 \rightarrow q_f$] At □, go to final state

$\delta(q_0,\square) = (q_f,\square,L)$ or R

For input 1? → reject

$q_0 00 \vdash^* 00 q_0 \square \vdash 0 q_f 0 \square$

$q_0 01 \vdash 0 q_0 1 \vdash$ halt: $q_0 \notin F \rightarrow$ reject

$\delta(q_0,1) =$ undefine → halt

Hyeyoung Park, Dept. Comp. Sci.

Turing Machines as Language Accepters (3/3)

Ex9.7] $\Sigma=\{a,b\}$ Turing machine accepting $L_1=\{a^nb^n \mid n\geq 1\}$

$M=(\{q_0,q_1,q_2,q_3,q_f\}, \{a,b\}, \{a,b,x,y,\square\}, \delta, q_0, \square, \{q_f\})$

Strategy: a,b를 하나씩 번갈아 x, y로 바꾸어감.

더 이상 a가 남아 있지 않을 때 남아있는 b의 개수를 셈.

[$q_0 \rightarrow q_1$] a → x, change state to q_1 $\delta(q_0,a)=(q_1,x,R)$

[q_1] Go to leftmost b $\delta(q_1,a)=(q_1,a,R)$, $\delta(q_1,y)=(q_1,y,R)$

[$q_1 \rightarrow q_2$] b→y, change state to q_2 $\delta(q_1,b)=(q_2,y,L)$

[q_2] Return to the rightmost x $\delta(q_2,a)=(q_2,a,L)$, $\delta(q_2,y)=(q_2,y,L)$

[$q_2 \rightarrow q_0$] reset state as q_0 (repeat q_0, q_1, q_2) $\delta(q_2,x)=(q_0,x,R)$

그 후로 a를 바꾸다가 y가 나올 때

[q_0] When no more a, change state to q_3 $\delta(q_0,y)=(q_3,y,R)$

[q_3] go to the rightmost y, $\delta(q_3,y)=(q_3,y,R)$

[$q_3 \rightarrow q_4$] change to final state to q_f when no character after y $\delta(q_3,\square)=(q_f,\square,R)$

halt

$q_0\, aab \vdash$ ⊢ $xxy\, q_1 \square$ → not final → reject

$q_0\, aabb \vdash$ ⊢ $xxyy \square q_f \square$ → final → accept

halt

Turing Machines as Transducers (1/5)

- Turing machine transducer M
  - An implementation of a function f: w′ = f(w)
  - provided that $q_0w \vdash^*_M q_fw'$ for some q_f in F.

head가 맨 앞

output

final 에서 halt. / head가 맨 앞에.

- Function f with domain D is **(Turing-) computable**
  - There exists a Turing machine M such that $q_0w \vdash^*_M q_ff(w)$, $q_f \in F$ for all $w \in D$.

튜링머신으로 그 함수를 구현할 수 있다

- All the common mathematical functions are Turing computable.

Turing Machines as Transducers (2/5)

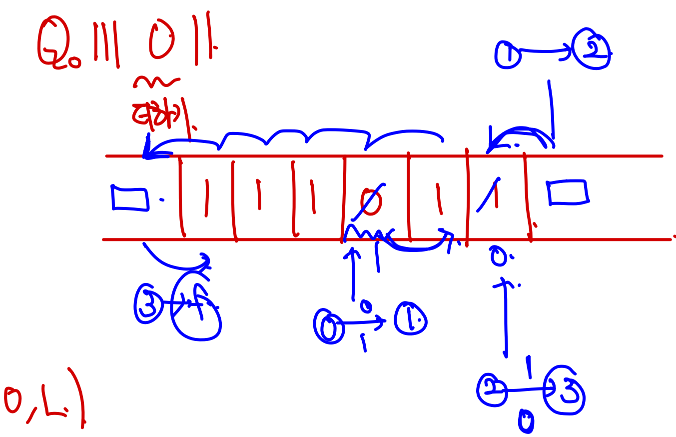
Ex9.9] Turing machine for x+y, (x,y : positive integers)

- Use unary notation $w(x) \in \{1\}^+$, $|w(x)|=x$
- Function mapping: $q_0 w(x)0w(y) \vdash q_f^* w(x+y)0$

$M=(\{q_0,q_1,q_2,q_3,q_f\}, \{1,0\}, \{1,0,\square\}, \delta, q_0, \square, \{q_f\})$

2 = 11
3 = 111

Strategy: 0을 1로 바꾸고 마지막 1을 0으로 바꿈.



[q_0] Go to 0 $\delta(q_0,1)=(q_0,1,R)$

[$q_0 \rightarrow q_1$] 0→1, change state to q_1 $\delta(q_0,0)=(q_1,1,R)$

[q_1] Go to $\square$ $\delta(q_1,1)=(q_1,1,R)$

[$q_1 \rightarrow q_2$] at $\square$, change state to q_2 $\delta(q_1,\square)=(q_2,\square,L)$

[q_2] rightmost 1 → 0, change state to q_3 $\delta(q_2,1)=(q_3,0,L)$

[q_3] return to leftmost 1 $\delta(q_3,1)=(q_3,1,L)$

[$q_3 \rightarrow q_f$] change state to q_f $\delta(q_3,\square)=(q_f,\square,R)$

3+2 : $q_0 111011 \vdash$ ....

$\vdash q_f 11111 0$

Hyeyoung Park, Dept. Comp. Sci.

Turing Machines as Transducers (3/5)

"2배를 하는 함수"

Ex9.10] Turing machine copying string of 1's.

- Function mapping: $q_0w \vdash^{*}_{M} q_fww$ for any $w(x) \in \{1\}^+$.

Strategy: 모든 1을 x로 바꾼 후, 하나씩 1로 다시 바꾸면서 새로운 1도 추가

- $M=(\{q_0,q_1,q_2,q_f\}, \{1\}, \{1,x,\square\}, \delta, q_0,\square, \{q_f\})$

[q_0] All 1 → x $\delta(q_0,1)=(q_0,x,R)$

[q_0→q_1] at □ change state to q_1 $\delta(q_0,\square)=(q_1,\square,L)$

[q_1] Go to rightmost x $\delta(q_1,1)=(q_1,1,L)$

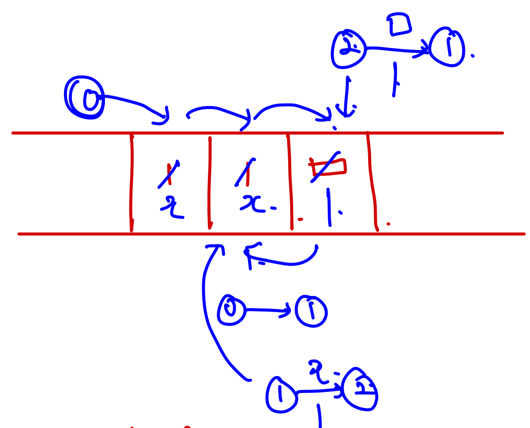
[q_1→q_2] x→ 1, change state q_2 $\delta(q_1,x)=(q_2,1,R)$

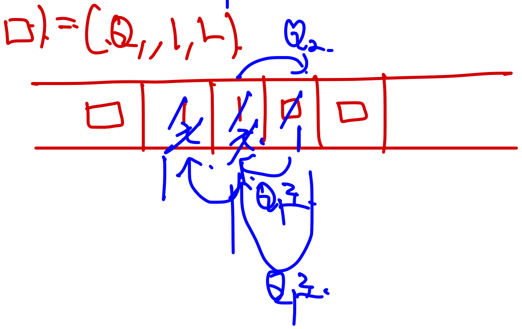
[q_2] Find □ $\delta(q_2,1)=(q_2,1,R)$

[q_2→q_1] □ → 1, reset state to q_1 (repeating q_1, q_2) $\delta(q_2,\square)=(q_1,1,L)$

[q_1→q_f] Change state to q_f when no more x

$\delta(q_1,\square)=(q_f,\square,R)$





2

$q_0 11 \vdash$

$\vdash^{*} q_f 1111$

$2+2$

$= 2 \cdot 2$

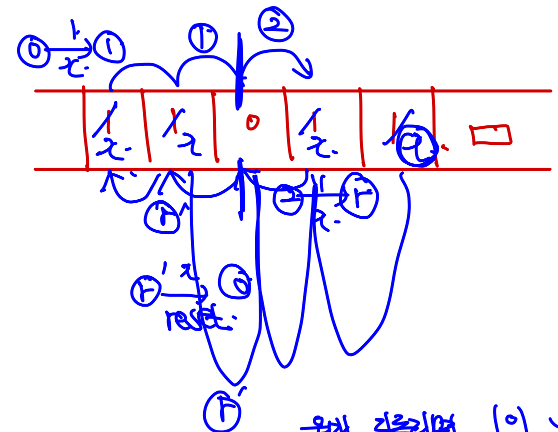
Turing Machines as Transducers (4/5)

Ex9.11] TM for conditional statement

halt in final state q_y if $x \geq y$ ($q_0 w(x)0w(y) \vdash q_y w(x)0w(y)$)

halt in a nonfinal state q_n if $x<y$ ($q_0 w(x)0w(y) \vdash q_n w(x)0w(y)$)

Tips] (ex9.7)과 유사한 방법을 사용



[$q_0 \rightarrow q_1$] First 1 → x, change state to q_1

[q_1] go to w(y) (find 0)

[$q_1 \rightarrow q_2$] at 0, change state to q_2

[$q_2 \rightarrow q_r$] In w(y), first 1 → x, change state to q_r

원래 [illegible] 1이 나와야 하는데 0이 나왔다면.

[$q_r \rightarrow q_{r'} \rightarrow q_0$]Return to last x, reset the state to q_0(repeating q_0, q_1, q_2, q_r, $q_{r'}$)

뒤에 1이 몇개 더 남았는지 count 하는 상황.

[$q_0 \rightarrow q_c$] When no more 1 in w(x), change state to q_c $\delta(q_0, 0) =$

[q_c] count remaining 1 in w(y)

Turing Machines as Transducers (4/5)

Ex9.11 continued]

x랑 y랑 같아졌다.

Qy로 가서 accept 하기 위한

[$q_c \rightarrow q_{y0}$] If no remaining 1 in w(y), change state to q_{y0} (preparing accept) 준비 작업.

$\delta(Q_c, \square) \rightarrow Q_{y0}$

[$q_c \rightarrow q_{n0}$] If no more than one 1's in w(y), change state to q_{n0} (preparing reject)

만약 □이 나오다 '1'이 나오면 y가 x보다 큰 거니까. n0으로.

n0 → preparing

$\delta(Q_c, 1) \rightarrow Q_{n0}$

[$q_2 \rightarrow q_{y0}$] When no more 1 in w(y), change state to q_{y0} (preparing accept)

2에서 □ 즉 x>y

11이면 [illegible]

$\delta(Q_2, \square) \rightarrow Q_{y0}$.

[q_{y0}] Move head to leftmost position changing all x→1, change state to q_y 되돌아 가면서

x를 1로 되돌려 놓은 state.

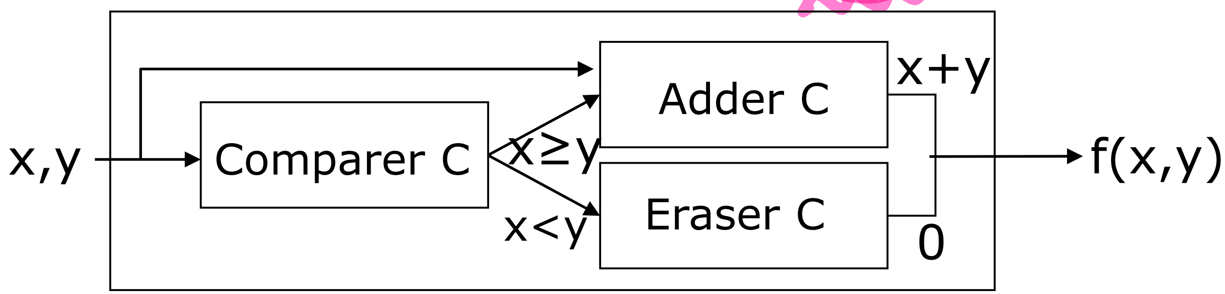
[q_{n0}] Move head to leftmost position changing all x→1, change state to q_n

Turing Machine using Building Bolcks

- TM's for Basic operations → TM for complex instructions
  - Use block diagram and pseudo code

Ex9.12] TM for $f(x,y) = x+y$ if $x \geq y$

$= 0$ if $x < y$



- Comparer C

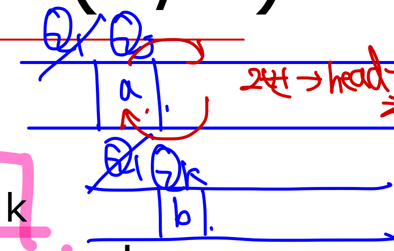
Adder를 시작해야 하니까.

$q_{C,0}w(x)0w(y) \vdash^* q_{A,0}w(x)0w(y)$ when $x \geq y$

$q_{C,0}w(x)0w(y) \vdash^* q_{E,0}w(x)0w(y)$ when $x < y$

- Adder A: $q_{A,0}w(x)0w(y) \vdash^* q_{A,f}w(x+y)0$ when $x \geq y$
- Eraser E: $q_{E,0}w(x)0w(y) \vdash^* q_{E,f}0$ when $x \geq y$

Turing Machine using Pseudo code (1/2)



2번 → head 움직임 ×

□ Macroinstructions for control statements

Ex9.13] conditional statement: if a then q_j else q_k

- Macroinstructions for implementing conditional statement

  $\delta(q_i, a) = (q_{j0}, a, R)$ for all $q_i \in Q$

  $\delta(q_i, b) = (q_{k0}, b, R)$ for all $q_i \in Q$ and all $b \in \Gamma - \{a\}$

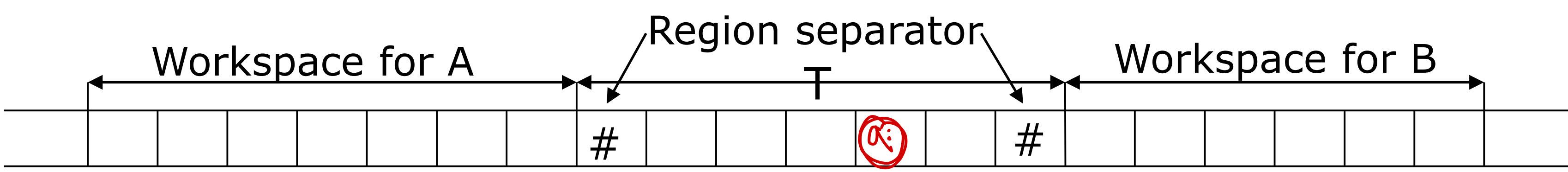
  $\delta(q_{j0}, c) = (q_j, c, L)$ for all $c \in \Gamma$

  $\delta(q_{k0}, c) = (q_k, c, L)$ for all $c \in \Gamma$

head를 움직이지 말고 처리해야 하는데, standard turing machine 에서는 불가능

□ Subprograms (when A calls B)

1. Write A's current state and arguments for B on tape region T
2. A passes controls to B
3. B find input from tape, and start transitions
4. Return results of B to tape and passes control to A



Turing Machine using Pseudo code (2/2)

ex:) w(3)=111.

Ex9.14] TM for multiplying 2 positive integers in unary notation

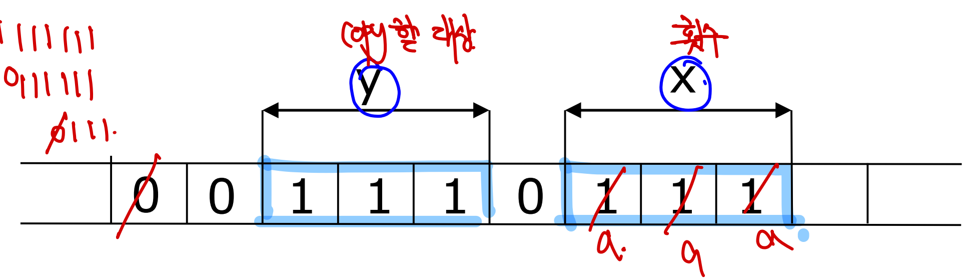
- Function mapping: $Q_0\, 00w(x)\,0w(y) \vdash^* Q_f\, 0w(x*y)\,0w(x)\,0w(y)$.

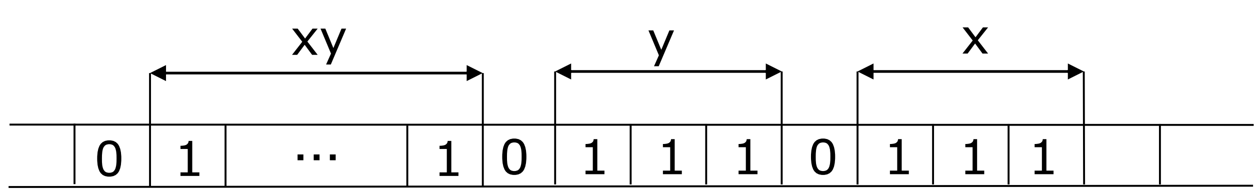
  (전체를 정의해야 한다.)

  1. Repeat the following steps until x contains no more 1's.
     Find a 1 in x and replace it with another symbol a.
     Replace the leftmost 0 by 0y.
  2. Replace all a's with 1's.

$y*x = \overbrace{y+y \cdots +y}^{x}$

Ex) 9.10을 subprogram으로 활용.





Turing's Thesis

[Turing thesis as the definition of mechanical computation]

1. Anything that can be done on any existing computer can also be done by a Turing machine.
2. No one has yet been able to suggest a problem
   - which is solvable by an algorithm
   - for which a TM program cannot be written

   튜링 머신으로 풀 수 없는 문제.
3. No alternative models for mechanical computation is more powerful than the TM model. (TM보다 power가 높은. 다른 모델은 없다)

[Algorithm for a function f : D → R]

- Turing machine M satisfying

  $q_0 d \vdash^{*}_{M} q_f f(d),\ q_f \in F,\ \text{for all } d \in D$

- Based on Turing's these, we can claim that anything we can do on any computer can also be done on a Turing machine.