Turing Machines

a fantastically simple of computation…with maximal power

1 An extremely simple model of computation

k-tape Turing machines

semi formal definition of Turing machines M

- finite control with finite set of states Z
- k tapes, infinite on both sides, divided into cells
- tape cells can hold symbols from a finite alphabet A which includes a blank symbol $B \in A$. Only finitely many cells have non blank symbols $a \neq B$
- We index tapes by numbers $i \in [1 : k]$. For each i there is a read/write head for tape i. Heads can read and print symbols from A and make head moves from $\{L, N, R\}$ (left, neutral, right)

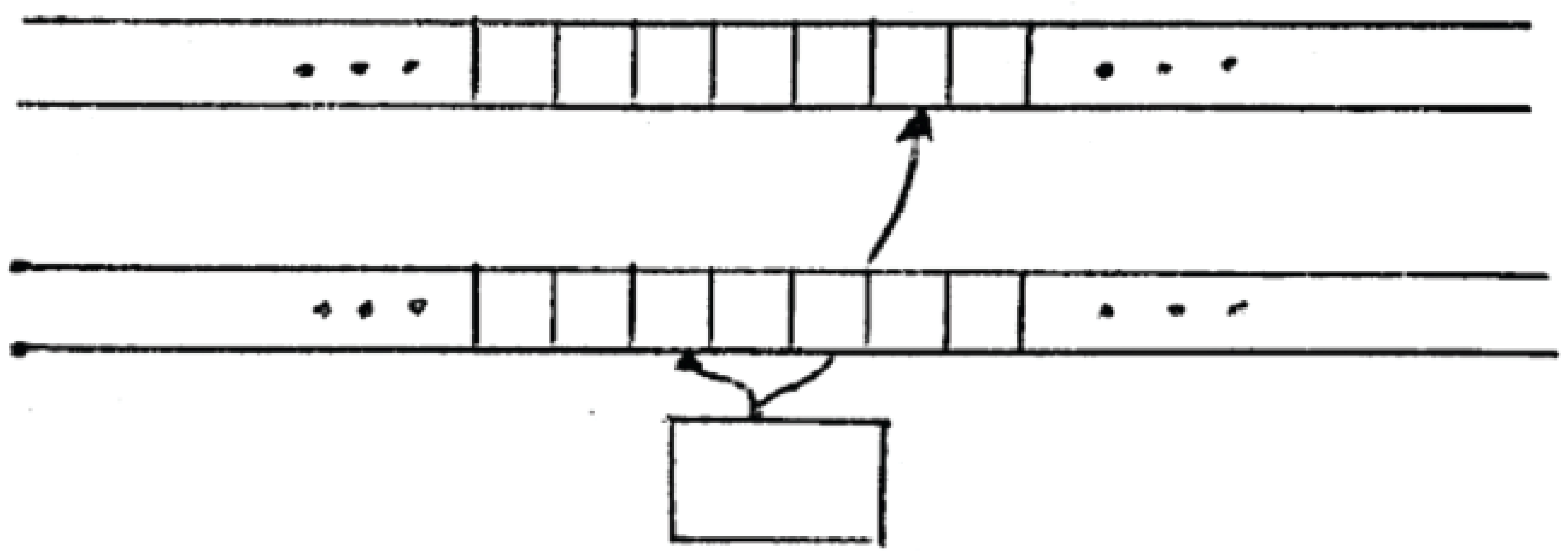

Figure 1: illustration of a 2 tape TM

1 An extremely simple model of computation

k-tape Turing machines

semi formal definition of Turing machines M

- finite control with finite set of states Z
- k tapes, infinite on both sides, divided into cells
- tape cells can hold symbols from a finite alphabet A which includes a blank symbol $B \in A$. Only finitely many cells have non blank symbols $a \neq B$
- We index tapes by numbers $i \in [1:k]$. For each i there is a read/write head for tape i. Heads can read and print symbols from A and make head moves from $\{L,N,R\}$ (left, neutral, right)

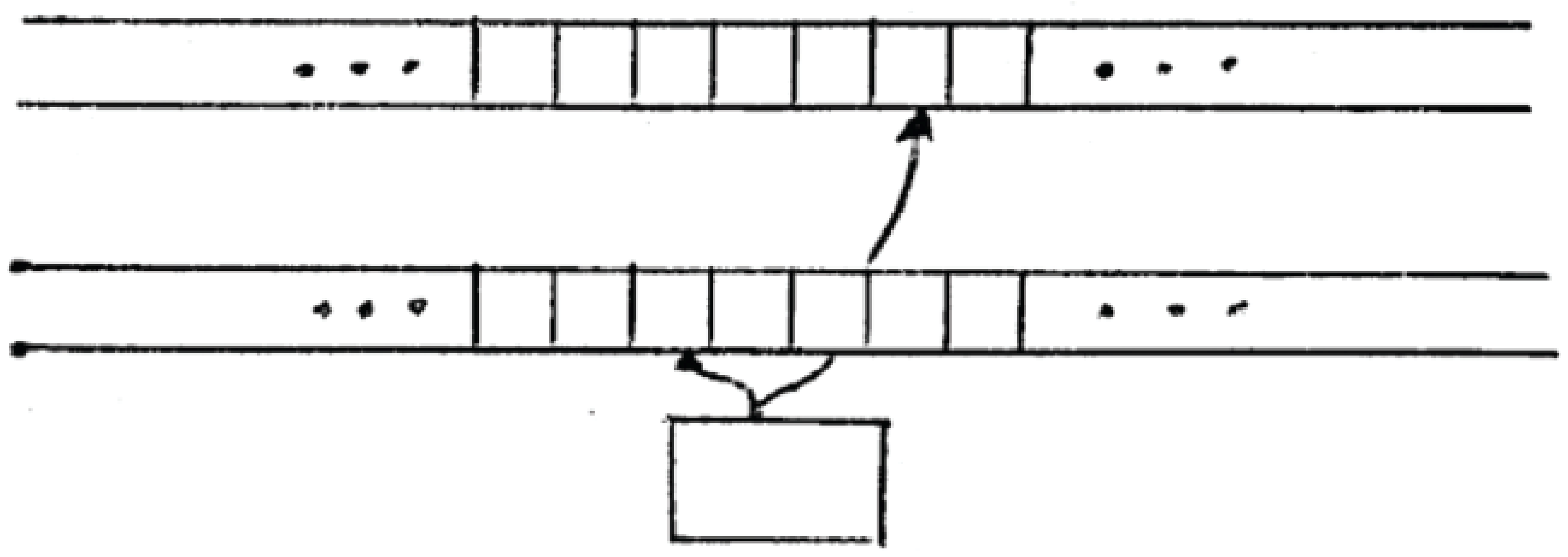

Figure 1: illustration of a 2 tape TM

- transition function

$$\delta : Z \times A^k \to Z \times A^k \times \{L,N,R\}^k$$

  where

$$\delta(z,a_1,\ldots,a_k) = (z',c_1,\ldots,c_k,m_1,\ldots,m_k)$$

  means: if M reads in state z on each tape i symbol a_i, then it goes to state z'; moreover on each tape i it overwrites a_i with c_i and makes head movement m_i.

- initial state $z_0 \in Z$
- set of end states $E \subseteq Z$

1 An extremely simple model of computation

k-tape Turing machines

semi formal definition of Turing machines M

- finite control with finite set of states Z
- k tapes, infinite on both sides, divided into cells
- tape cells can hold symbols from a finite alphabet A which includes a blank symbol $B \in A$. Only finitely many cells have non blank symbols $a \neq B$
- We index tapes by numbers $i \in [1:k]$. For each i there is a read/write head for tape i. Heads can read and print symbols from A and make head moves from $\{L,N,R\}$ (left, neutral, right)

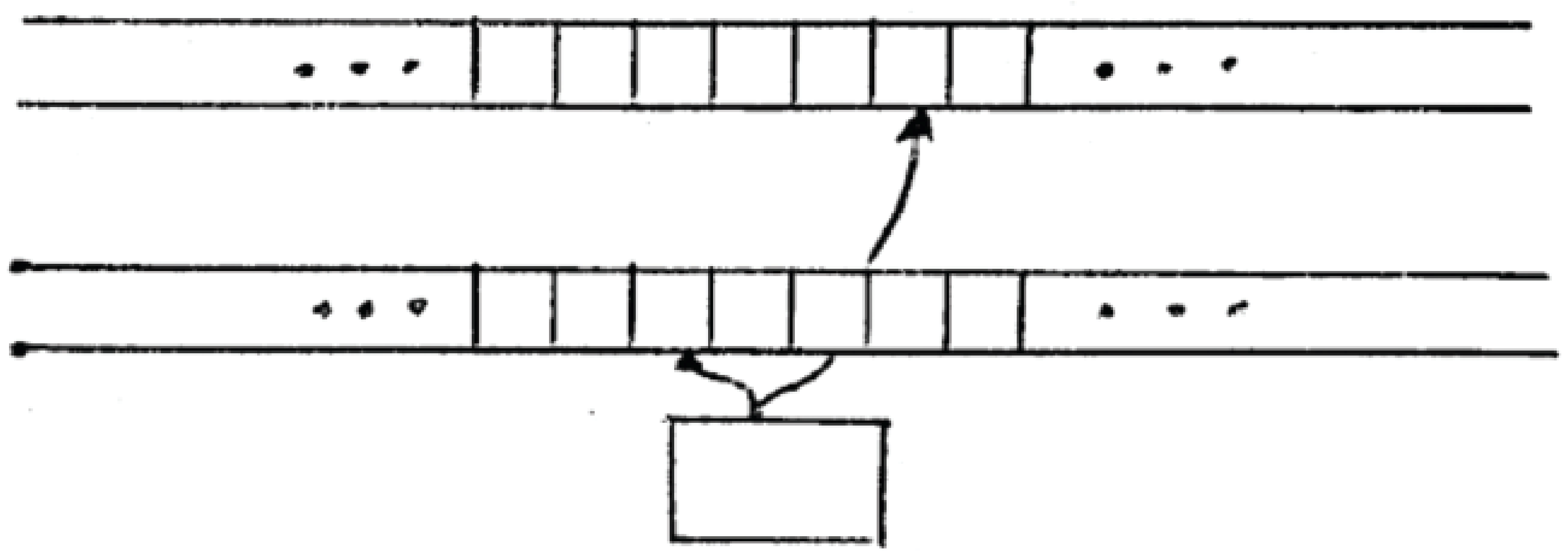

Figure 1: illustration of a 2 tape TM

- transition function

$$\delta : Z \times A^k \to Z \times A^k \times \{L,N,R\}^k$$

where

$$\delta(z,a_1,\ldots,a_k) = (z',c_1,\ldots,c_k,m_1,\ldots,m_k)$$

means: if M reads in state z on each tape i symbol a_i, then it goes to state z'; moreover on each tape i it overwrites a_i with c_i and makes head movement m_i.

- initial state $z_0 \in Z$
- set of end states $E \subseteq Z$

$$M = (Z,A,\delta,z_0,E)$$

as above and

- certain symbols are always available

$$\{0,1,B,\#\} \subseteq A$$

  Symbol # will serve to separate strings in $\mathbb{B}^*$

- in end states (alternative definitions)
  1. there is no next step

$$\delta : (Z \setminus E) \times A^K \to z \times A^k \times \{L,N,R\}^k$$

  2. sometimes useful: looping in end state

$$\delta(z,a) = (z,a,N^k) \text{ for } z \in E$$

  saves sometimes case split between lang and short computations

- transition function

$$\delta : Z \times A^k \to Z \times A^k \times \{L,N,R\}^k$$

where

$$\delta(z,a_1,\dots,a_k) = (z',c_1,\dots,c_k,m_1,\dots,m_k)$$

means: if M reads in state z on each tape i symbol a_i, then it goes to state z'; moreover on each tape i it overwrites a_i with c_i and makes head movement m_i.

- initial state $z_0 \in Z$
- set of end states $E \subseteq Z$

$$M = (Z,A,\delta,z_0,E)$$

as above and

- certain symbols are always available

$$\{0,1,B,\#\} \subseteq A$$

Symbol # will serve to separate strings in $\mathbb{B}^*$

- in end states (alternative definitions)

1. there is no next step

$$\delta : (Z \setminus E) \times A^K \to z \times A^k \times \{L,N,R\}^k$$

2. sometimes useful: looping in end state

$$\delta(z,a) = (z,a,N^k) \text{ for } z \in E$$

saves sometimes case split between lang and short computations

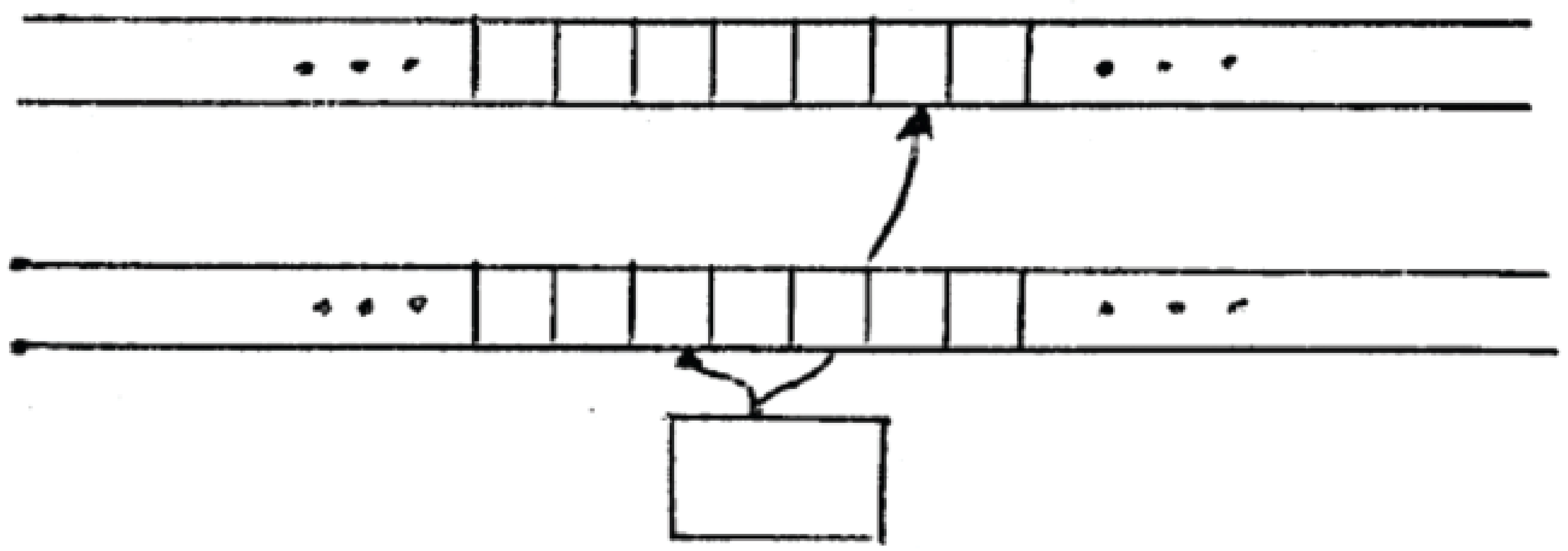

Figure 1: illustration of a 2 tape TM

I/O convention :

- input is non blank portion of tape 1, head 1 on first symbol of input
- all other tapes initially blank
- output is on non blank portion on tape 1, head 1 on first symbol of output

- transition function

$$\delta : Z \times A^k \to Z \times A^k \times \{L,N,R\}^k$$

where

$$\delta(z,a_1,\dots,a_k) = (z',c_1,\dots,c_k,m_1,\dots,m_k)$$

means: if M reads in state z on each tape i symbol a_i, then it goes to state z'; moreover on each tape i it overwrites a_i with c_i and makes head movement m_i.

- initial state $z_0 \in Z$
- set of end states $E \subseteq Z$

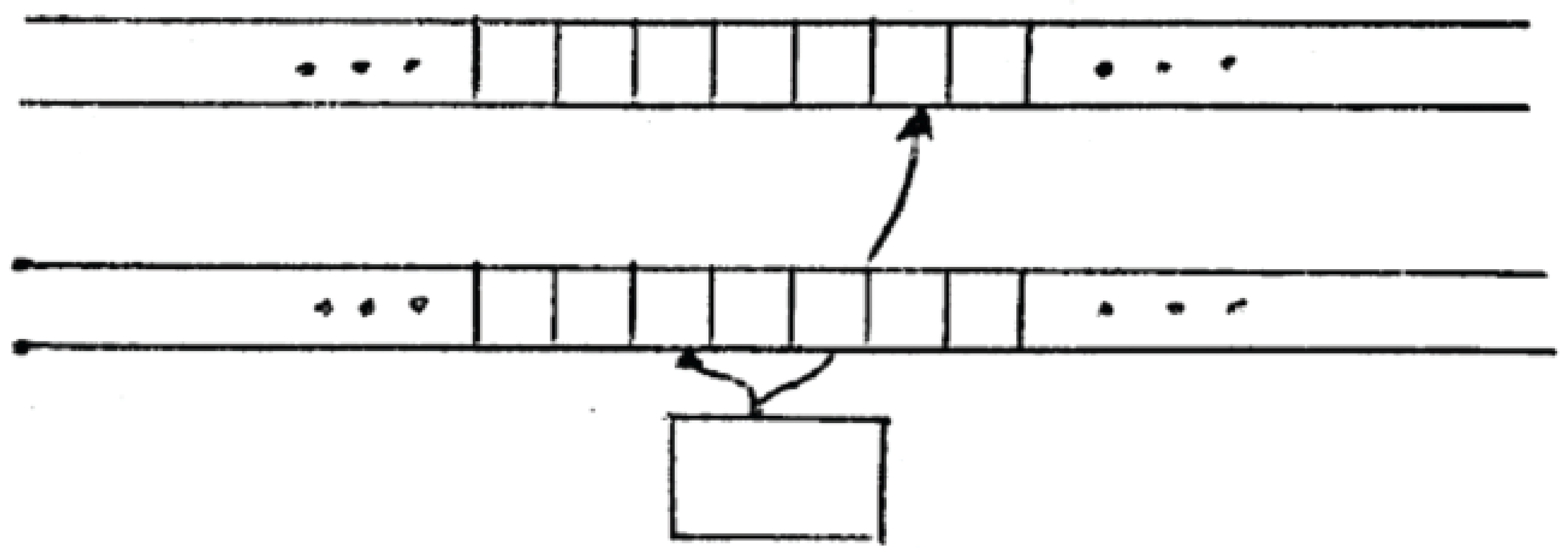

Figure 1: illustration of a 2 tape TM

I/O convention :

- input is non blank portion of tape 1, head 1 on first symbol of input
- all other tapes initially blank
- output is on non blank portion on tape 1, head 1 on first symbol of output

example: incrementing binary numbers

- input: binary number $bin(n)$
- output: $bin(n+1)$

machine M : $\boxed{tape\ 1 = tape\ 1\ +\ 1}$ has 1 tape

go to right end, state q_i means carry $= i$

$$\begin{aligned} \delta(z_0,a) &= (z_0,a,R) \quad a \in \mathbb{B} \\ \delta(z_0,B) &= (q_0,B,L) \end{aligned}$$

- transition function

$$\delta : Z \times A^k \to Z \times A^k \times \{L,N,R\}^k$$

where

$$\delta(z,a_1,\ldots,a_k) = (z',c_1,\ldots,c_k,m_1,\ldots,m_k)$$

means: if M reads in state z on each tape i symbol a_i, then it goes to state z'; moreover on each tape i it overwrites a_i with c_i and makes head movement m_i.

- initial state $z_0 \in Z$
- set of end states $E \subseteq Z$

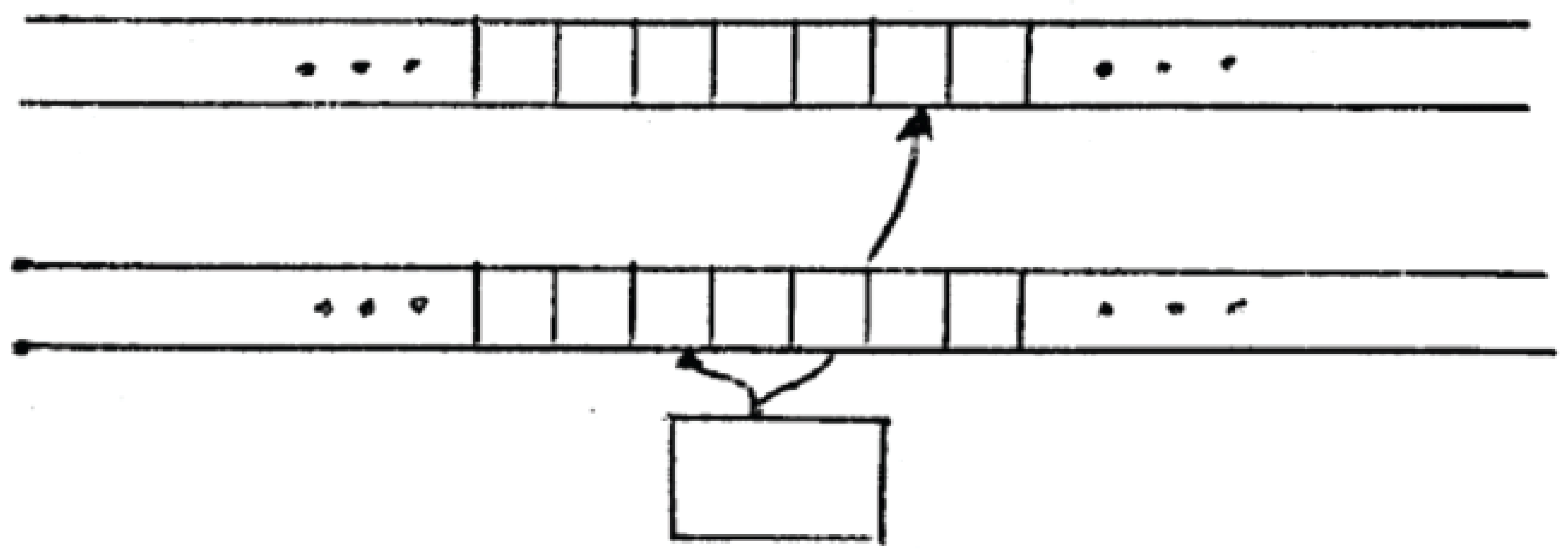

Figure 1: illustration of a 2 tape TM

I/O convention :

- input is non blank portion of tape 1, head 1 on first symbol of input
- all other tapes initially blank
- output is on non blank portion on tape 1, head 1 on first symbol of output

example: incrementing binary numbers

- input: binary number $bin(n)$
- output: $bin(n+1)$

machine M : $\boxed{tape\ 1 = tape\ 1\ +\ 1}$ has 1 tape

go to right end, state q_i means carry $= i$

$$\begin{aligned}\delta(z_0,a) &= (z_0,a,R) \quad a \in \mathbb{B} \\ \delta(z_0,B) &= (q_1,B,L)\end{aligned}$$

add for current position

$$\begin{aligned}\delta(q_0,a) &= (q_0,a,L) \quad a \in \mathbb{B} \\ \delta(q_1,0) &= (q_0,1,L) \\ \delta(q_1,1) &= (q_1,0,L)\end{aligned}$$

return to left end of output

$$\begin{aligned}\delta(q_0,B) &= (z_e,B,R) \\ \delta(q_1;B) &= (z_e,1,R)\end{aligned}$$

end states $E = \{z_e\}$

example: decrementing binary numbers machine M : $\boxed{tape\ 1 = tape\ 1 - 1:}$ exercise

set of configurations K

$$K = A^* \circ Z \circ A^*$$

where

$$k = uzv$$

means:

- non blank part of tape is substring of uv
- head is on v_1
- state is z
- start configuration if $z = z_0$
- end configuration if $z \in E$

2 TM semantics for $k = 1$ tape

for $k \geq 2$: exercise

2 TM semantics for $k = 1$ tape

for $k \geq 2$: exercise

set of configurations K

$$K = A^* \circ Z \circ A^*$$

where

$$k = uzv$$

means:

- non blank part of tape is substring of uv
- head is on v_1
- state is z
- start configuration if $z = z_0$
- end configuration if $z \in E$

def: next state relation $\vdash$

$$\vdash \subset K \times K \quad \text{here a partial function}$$

for

$$u, v \in A^+ \ , \ a, b, c \in A \ , \ z, z' \in Z$$

define by case split (on empty tape around the head)

set of configurations K

$$K = A^* \circ Z \circ A^*$$

where

$$k = uzv$$

means:

- non blank part of tape is substring of uv
- head is on v_1
- state is z
- start configuration if $z = z_0$
- end configuration if $z \in E$

def: next state relation $\vdash$

$$\vdash \subset K \times K \quad \text{here a partial function}$$

for

$$u, v \in A^+ \ , \ a, b, c \in A \ , \ z, z' \in Z$$

define by case split (on empty tape around the head)

- non blank tape on both sides of head

$$ubzav \vdash \begin{cases} uz'bcv & \delta(z,a) = (z',c,L) \\ ubz'cv & \delta(z,a) = (z',c,N) \\ ubcz'v & \delta(z,a) = (z',c,R) \end{cases}$$

set of configurations K

$$K = A^* \circ Z \circ A^*$$

where

$$k = uzv$$

means:

- non blank part of tape is substring of uv
- head is on v_1
- state is z
- start configuration if $z = z_0$
- end configuration if $z \in E$

def: next state relation $\vdash$

$$\vdash \subset \ K \times K \quad \text{here a partial function}$$

for

$$u, v \in A^+ \ , \ a, b, c \in A \ , \ z, z' \in Z$$

define by case split (on empty tape around the head)

- non blank tape on both sides of head

$$ubzav \vdash \begin{cases} uz'bcv & \delta(z,a) = (z',c,L) \\ ubz'cv & \delta(z,a) = (z',c,N) \\ ubcz'v & \delta(z,a) = (z',c,R) \end{cases}$$

- blank tape left of head

$$zav \vdash \begin{cases} z'Bcv & \delta(z,a) = (z',c,L) \\ z'cv & \delta(z,a) = (z',c,N) \\ cz'v & \delta(z,a) = (z',c,R) \end{cases}$$

set of configurations K

$$K = A^* \circ Z \circ A^*$$

where

$$k = uzv$$

means:

- non blank part of tape is substring of uv
- head is on v_1
- state is z
- start configuration if $z = z_0$
- end configuration if $z \in E$

def: next state relation $\vdash$

$$\vdash \subset \; K \times K \quad \text{here a partial function}$$

for

$$u, v \in A^+ \; , \; a, b, c \in A \; , \; z, z' \in Z$$

define by case split (on empty tape around the head)

- non blank tape on both sides of head

$$ubzav \vdash \begin{cases} uz'bcv & \delta(z,a) = (z',c,L) \\ ubz'cv & \delta(z,a) = (z',c,N) \\ ubcz'v & \delta(z,a) = (z',c,R) \end{cases}$$

- blank tape left of head

$$zav \vdash \begin{cases} z'Bcv & \delta(z,a) = (z',c,L) \\ z'cv & \delta(z,a) = (z',c,N) \\ cz'v & \delta(z,a) = (z',c,R) \end{cases}$$

- blank tape right of head

$$ubz \vdash \begin{cases} uz'bc & \delta(z,B) = (z',c,L) \\ ubz'c & \delta(z,B) = (z',c,N) \\ ubcz' & \delta(z,B) = (z',c,R) \end{cases}$$

set of configurations K

$$K = A^* \circ Z \circ A^*$$

where

$$k = uzv$$

means:

- non blank part of tape is substring of uv
- head is on v_1
- state is z
- start configuration if $z = z_0$
- end configuration if $z \in E$

def: next state relation $\vdash$

$$\vdash \subset K \times K \quad \text{here a partial function}$$

for

$$u,v \in A^+ \ , \ a,b,c \in A \ , \ z,z' \in Z$$

define by case split (on empty tape around the head)

- non blank tape on both sides of head

$$ubzav \vdash \begin{cases} uz'bcv & \delta(z,a) = (z',c,L) \\ ubz'cv & \delta(z,a) = (z',c,N) \\ ubcz'v & \delta(z,a) = (z',c,R) \end{cases}$$

- blank tape left of head

$$zav \vdash \begin{cases} z'Bcv & \delta(z,a) = (z',c,L) \\ z'cv & \delta(z,a) = (z',c,N) \\ cz'v & \delta(z,a) = (z',c,R) \end{cases}$$

- blank tape right of head

$$ubz \vdash \begin{cases} uz'bc & \delta(z,B) = (z',c,L) \\ ubz'c & \delta(z,B) = (z',c,N) \\ ubcz' & \delta(z,B) = (z',c,R) \end{cases}$$

- blank tape left and right of head

$$z \vdash \begin{cases} z'Bc & \delta(z,B) = (z',c,L) \\ z'c & \delta(z,B) = (z',c,N) \\ cz' & \delta(z,B) = (z',c,R) \end{cases}$$

set of configurations K

$$K = A^* \circ Z \circ A^*$$

where

$$k = uzv$$

means:

- non blank part of tape is substring of uv
- head is on v_1
- state is z
- start configuration if $z = z_0$
- end configuration if $z \in E$

def: next state relation $\vdash$

$$\vdash \subset K \times K \quad \text{here a partial function}$$

for

$$u,v \in A^+ \;,\; a,b,c \in A \;,\; z,z' \in Z$$

define by case split (on empty tape around the head)

- non blank tape on both sides of head

$$\boxed{ubzav \vdash \begin{cases} uz'bcv & \delta(z,a) = (z',c,L) \\ ubz'cv & \delta(z,a) = (z',c,N) \\ ubcz'v & \delta(z,a) = (z',c,R) \end{cases}}$$

- blank tape left of head

$$zav \vdash \begin{cases} z'Bcv & \delta(z,a) = (z',c,L) \\ z'cv & \delta(z,a) = (z',c,N) \\ cz'v & \delta(z,a) = (z',c,R) \end{cases}$$

- blank tape right of head

$$ubz \vdash \begin{cases} uz'bc & \delta(z,B) = (z',c,L) \\ ubz'c & \delta(z,B) = (z',c,N) \\ ubcz' & \delta(z,B) = (z',c,R) \end{cases}$$

- blank tape left and right of head

$$z \vdash \begin{cases} z'Bc & \delta(z,B) = (z',c,L) \\ z'c & \delta(z,B) = (z',c,N) \\ cz' & \delta(z,B) = (z',c,R) \end{cases}$$

In later applications we can often

- surround input by enough blanks
- then we can ignore rules 2 to 4
- so we have a 3 line definition of steps

def: computations of

$$M = (Z, A, \delta, z_0, E) \quad \text{with input} \quad w \in (A \setminus \{B\})^*$$

- sequence (k_i) of configurations
- with start configuration

$$k_0 = B \dots B z_0 w B \dots B$$

- computation steps

$$k_i \vdash k_{i+1} \quad \text{for all except the last } i$$

- finite of lenght T if

$$k = (k_0, \dots, k_T)$$

  and k_T is end configuration. M started with w *halts*.

- infinite if no k_i is end configuration. M started with w does not halt.
- *canonical* computation if

$$k_0 = z_0 w$$

- tape used: $|w|$ + number of tape cells visited outside of original input during the computation.

def: computations of

$$M = (Z, A, \delta, z_0, E) \quad \text{with input} \quad w \in (A \setminus \{B\})^*$$

- sequence (k_i) of configurations
- with start configuration

$$k_0 = B \dots B z_0 w B \dots B$$

- computation steps

$$k_i \vdash k_{i+1} \quad \text{for all except the last } i$$

- finite of lenght T if

$$k = (k_0, \dots, k_T)$$

  and k_T is end configuration. M started with w *halts*.
- infinite if no k_i is end configuration. M started with w does not halt.
- *canonical* computation if

$$k_0 = z_0 w$$

- tape used: $|w|$ + number of tape cells visited outside of original input during the computation.

example of not halting

$$\delta(z_0, a) = (z_0, a, R) \quad \text{for all } a \in A$$

def: computations of

$$M = (Z, A, \delta, z_0, E) \quad \text{with input} \quad w \in (A \setminus \{B\})^*$$

- sequence (k_i) of configurations
- with start configuration

$$k_0 = B \dots B z_0 w B \dots B$$

- computation steps

$$k_i \vdash k_{i+1} \quad \text{for all except the last } i$$

- finite of lenght T if

$$k = (k_0, \dots, k_T)$$

  and k_T is end configuration. M started with w *halts*.
- infinite if no k_i is end configuration. M started with w does not halt.
- *canonical* computation if

$$k_0 = z_0 w$$

- tape used: $|w|$ + number of tape cells visited outside of original input during the computation.

example of not halting

$$\delta(z_0, a) = (z_0, a, R) \quad \text{for all } a \in A$$

what can be computed by 1 tape TM's?

- Answer: everything that can be computed at all
- this is known as *Church's thesis*.
- We cannot prove it (based on what definition or axiom could we do that?)
- we *can* supply evidence, and we *will* do it. Of course *here*
- for instance that it's the same as the recursive functions

3 Turing Machine programming

def: regular TM's nice for composition of TMs

- at the end of computation head 1 is at start of the non blank portion of tape 1
- tape inscription of tape 1 is not interrupted by B's
- tapes $i > 1$ are blank

3 Turing Machine programming

def: regular TM's

nice for composition of TMs

- at the end of computation head 1 is at start of the non blank portion of tape 1
- tape inscription of tape 1 is not interrupted by B's
- tapes $i > 1$ are blank

copying a tape inscription machine M with name $\boxed{tape\ 2 = tape\ 1}$

$$\begin{array}{rcll}
\delta(z_0, a, B) & = & (z_0, a, a, R, R) & a \in \mathbb{B} \\
\delta(z_0, B, B) & = & (z_1, B, B, L, L) & \\
\delta(z_1, a, a) & = & (z_1, a, a, L, L) & a \in \mathbb{B} \\
\delta(z_1, B, B) & = & (z_e, B, B, R, R) &
\end{array}$$

3 Turing Machine programming

def: regular TM's nice for composition of TMs

- at the end of computation head 1 is at start of the non blank portion of tape 1
- tape inscription of tape 1 is not interrupted by B's
- tapes $i > 1$ are blank

copying a tape inscription machine M with name $\boxed{tape\ 2 = tape\ 1}$

$$\begin{aligned}
\delta(z_0, a, B) &= (z_0, a, a, R, R) \quad a \in \mathbb{B} \\
\delta(z_0, B, B) &= (z_1, B, B, L, L) \\
\delta(z_1, a, a) &= (z_1, a, a, L, L) \quad a \in \mathbb{B} \\
\delta(z_1, B, B) &= (z_e, B, B, R, R)
\end{aligned}$$

concatenate tape inscriptions: machines $\boxed{tape\ 1 \ = tape\ 1\ \#tape\ 2}$ and $\boxed{tape\ 1 = tape\ 2\#tape\ 1 \text{ exercise}}$

3 Turing Machine programming

def: regular TM's nice for composition of TMs

- at the end of computation head 1 is at start of the non blank portion of tape 1
- tape inscription of tape 1 is not interrupted by B's
- tapes $i > 1$ are blank

copying a tape inscription machine M with name $\boxed{tape\ 2 = tape\ 1}$

$$\begin{aligned}
\delta(z_0,a,B) &= (z_0,a,a,R,R) \quad a \in \mathbb{B} \\
\delta(z_0,B,B) &= (z_1,B,B,L,L) \\
\delta(z_1,a,a) &= (z_1,a,a,L,L) \quad a \in \mathbb{B} \\
\delta(z_1,B,B) &= (z_e,B,B,R,R)
\end{aligned}$$

concatenate tape inscriptions: machines $\boxed{tape\ 1 \ = tape\ 1\ \#tape\ 2}$ and $\boxed{tape\ 1 = tape\ 2\#tape\ 1 \text{ exercise}}$

erasing a tape machine $\boxed{erase\ tape\ 1}$

$$\begin{aligned}
\delta(z_0,a) &= (z_0,B,R) \quad a \neq B \\
\delta(z_0,B) &= (z_e,B,N)
\end{aligned}$$

3 Turing Machine programming

def: regular TM's nice for composition of TMs

- at the end of computation head 1 is at start of the non blank portion of tape 1
- tape inscription of tape 1 is not interrupted by B's
- tapes $i > 1$ are blank

copying a tape inscription machine M with name $\boxed{tape\ 2 = tape\ 1}$

$$\begin{aligned}
\delta(z_0, a, B) &= (z_0, a, a, R, R) \quad a \in \mathbb{B} \\
\delta(z_0, B, B) &= (z_1, B, B, L, L) \\
\delta(z_1, a, a) &= (z_1, a, a, L, L) \quad a \in \mathbb{B} \\
\delta(z_1, B, B) &= (z_e, B, B, R, R)
\end{aligned}$$

concatenate tape inscriptions: machines $\boxed{tape\ 1 = tape\ 1\,\#tape\ 2}$ and $\boxed{tape\ 1 = tape\ 2\#tape\ 1 \text{ exercise}}$

erasing a tape machine $\boxed{erase\ tape\ 1}$

$$\begin{aligned}
\delta(z_0, a) &= (z_0, B, R) \quad a \neq B \\
\delta(z_0, B) &= (z_e, B, N)
\end{aligned}$$

remembering (finite amouts of) information in the state Machine $\boxed{shiftr\ tape\ 1.}$

shifts inscription $w \in \{0,1,\#\}^*$ of tape one cell to the right

$$\begin{aligned}
Z &= \{z_0, z_e\} \cup \{z^a : a \in A\} \\
\delta(z_0, a) &= (z^a, B, R) \quad a \in \{0,1,\#\} \\
\delta(z^a, b) &= (z^b, a, R) \quad \in \{0,1,\#\} \\
\delta(z^a, B) &= (z_e, a, R) \quad a \in \{0,1,\#\}
\end{aligned}$$

3 Turing Machine programming

def: regular TM's nice for composition of TMs

- at the end of computation head 1 is at start of the non blank portion of tape 1
- tape inscription of tape 1 is not interrupted by B's
- tapes $i > 1$ are blank

copying a tape inscription machine M with name $\boxed{tape\ 2 = tape\ 1}$

$$\begin{aligned}
\delta(z_0,a,B) &= (z_0,a,a,R,R) \quad a \in \mathbb{B} \\
\delta(z_0,B,B) &= (z_1,B,B,L,L) \\
\delta(z_1,a,a) &= (z_1,a,a,L,L) \quad a \in \mathbb{B} \\
\delta(z_1,B,B) &= (z_e,B,B,R,R)
\end{aligned}$$

concatenate tape inscriptions: machines $\boxed{tape\ 1 \ = tape\ 1\ \#tape\ 2}$ and $\boxed{tape\ 1 = tape\ 2\#tape\ 1 \text{ exercise}}$

erasing a tape machine $\boxed{erase\ tape\ 1}$

$$\begin{aligned}
\delta(z_0,a) &= (z_0,B,R) \quad a \neq B \\
\delta(z_0,B) &= (z_e,B,N)
\end{aligned}$$

remembering (finite amouts of) information in the state Machine $\boxed{shiftr\ tape\ 1.}$

shifts inscription $w \in \{0,1,\#\}^*$ of tape one cell to the right

$$\begin{aligned}
Z &= \{z_0,z_e\} \cup \{z^a \ : \ a \in A\} \\
\delta(z_0,a) &= (z^a,B,R) \quad a \in \{0,1,\#\} \\
\delta(z^a,b) &= (z^b,a,R) \quad \in \{0,1,\#\} \\
\delta(z^a,B) &= (z_e,a,R) \quad a \in \{0,1,\#\}
\end{aligned}$$

not regular

3 Turing Machine programming

def: regular TM's nice for composition of TMs

- at the end of computation head 1 is at start of the non blank portion of tape 1
- tape inscription of tape 1 is not interrupted by B's
- tapes $i > 1$ are blank

copying a tape inscription machine M with name $\boxed{tape\ 2 = tape\ 1}$

$$\begin{aligned}
\delta(z_0,a,B) &= (z_0,a,a,R,R) \quad a \in \mathbb{B} \\
\delta(z_0,B,B) &= (z_1,B,B,L,L) \\
\delta(z_1,a,a) &= (z_1,a,a,L,L) \quad a \in \mathbb{B} \\
\delta(z_1,B,B) &= (z_e,B,B,R,R)
\end{aligned}$$

concatenate tape inscriptions: machines $\boxed{tape\ 1\ = tape\ 1\ \#tape\ 2}$ and $\boxed{tape\ 1 = tape\ 2\#tape\ 1 \text{ exercise}}$

erasing a tape machine $\boxed{erase\ tape\ 1}$

$$\begin{aligned}
\delta(z_0,a) &= (z_0,B,R) \quad a \neq B \\
\delta(z_0,B) &= (z_e,B,N)
\end{aligned}$$

remembering (finite amouts of) information in the state Machine $\boxed{shiftr\ tape\ 1.}$

shifts inscription $w \in \{0,1,\#\}^*$ of tape one cell to the right

$$\begin{aligned}
Z &= \{z_0,z_e\} \cup \{z^a\ :\ a \in A\} \\
\delta(z_0,a) &= (z^a,B,R) \quad a \in \{0,1,\#\} \\
\delta(z^a,b) &= (z^b,a,R) \quad \in \{0,1,\#\} \\
\delta(z^a,B) &= (z_e,a,R) \quad a \in \{0,1,\#\}
\end{aligned}$$

not regular

shifting left: with $\boxed{shiftl\ tape\ 1}$ exercise

3 Turing Machine programming

def: regular TM's nice for composition of TMs

- at the end of computation head 1 is at start of the non blank portion of tape 1
- tape inscription of tape 1 is not interrupted by B's
- tapes $i > 1$ are blank

copying a tape inscription machine M with name $\boxed{tape\ 2 = tape\ 1}$

$$\begin{aligned}
\delta(z_0,a,B) &= (z_0,a,a,R,R) \quad a \in \mathbb{B}\\
\delta(z_0,B,B) &= (z_1,B,B,L,L)\\
\delta(z_1,a,a) &= (z_1,a,a,L,L) \quad a \in \mathbb{B}\\
\delta(z_1,B,B) &= (z_e,B,B,R,R)
\end{aligned}$$

concatenate tape inscriptions: machines $\boxed{tape\ 1 = tape\ 1\#tape\ 2}$ and $\boxed{tape\ 1 = tape\ 2\#tape\ 1 \text{ exercise}}$

erasing a tape machine $\boxed{erase\ tape\ 1}$

$$\begin{aligned}
\delta(z_0,a) &= (z_0,B,R) \quad a \neq B\\
\delta(z_0,B) &= (z_e,B,N)
\end{aligned}$$

remembering (finite amouts of) information in the state Machine $\boxed{shiftr\ tape\ 1.}$

shifts inscription $w \in \{0,1,\#\}^*$ of tape one cell to the right

$$\begin{aligned}
Z &= \{z_0,z_e\} \cup \{z^a\ :\ a \in A\}\\
\delta(z_0,a) &= (z^a,B,R) \quad a \in \{0,1,\#\}\\
\delta(z^a,b) &= (z^b,a,R) \quad \in \{0,1,\#\}\\
\delta(z^a,B) &= (z_e,a,R) \quad a \in \{0,1,\#\}
\end{aligned}$$

not regular

shifting left: with $\boxed{shiftl\ tape\ 1}$
exercise

storing a word w in finite control Let $w = w[n-1:0] \in \mathbb{B}^+$

machine $\boxed{tape\ 1 = w}$ writes w on empty tape.

$$\begin{aligned}
Z &= \{z_0,\dots z_n\}\\
z_e &= z_n\\
\delta(z_i,B) &= \begin{cases}(z_{i+1},w_i,L) & i < n-1\\ (z_n,w_{n-1},N) & i = n-1\end{cases}
\end{aligned}$$

copying a tape inscription machine M with name $\boxed{tape\ 2 = tape\ 1}$

$$\begin{aligned}
\delta(z_0,a,B) &= (z_0,a,a,R,R) \quad a \in \mathbb{B}\\
\delta(z_0,B,B) &= (z_1,B,B,L,L)\\
\delta(z_1,a,a) &= (z_1,a,a,L,L) \quad a \in \mathbb{B}\\
\delta(z_1,B,B) &= (z_e,B,B,R,R)
\end{aligned}$$

concatenate tape inscriptions: machines $\boxed{tape\ 1\ = tape\ 1\ \#tape\ 2}$ and $\boxed{tape\ 1 = tape\ 2\#tape\ 1 \text{ exercise}}$

erasing a tape machine $\boxed{erase\ tape\ 1}$

$$\begin{aligned}
\delta(z_0,a) &= (z_0,B,R) \quad a \neq B\\
\delta(z_0,B) &= (z_e,B,N)
\end{aligned}$$

remembering (finite amouts of) information in the state Machine $\boxed{shiftr\ tape\ 1.}$

shifts inscription $w \in \{0,1,\#\}^*$ of tape one cell to the right

$$\begin{aligned}
Z &= \{z_0,z_e\} \cup \{z^a\ :\ a \in A\}\\
\delta(z_0,a) &= (z^a,B,R) \quad a \in \{0,1,\#\}\\
\delta(z^a,b) &= (z^b,a,R) \quad \in \{0,1,\#\}\\
\delta(z^a,B) &= (z_e,a,R) \quad z \in Z \setminus \{z_e\}
\end{aligned}$$

not regular

shifting left: with $\boxed{shiftl\ tape\ 1}$
exercise

storing a word w in finite control Let $w = w[n-1:0] \in \mathbb{B}^+$

machine $\boxed{tape\ 1 = w}$ writes w on empty tape.

$$\begin{aligned}
Z &= \{z_0,\dots z_n\}\\
z_e &= z_n\\
\delta(z_i,B) &= \begin{cases}(z_{i+1},w_i,L) & i < n-1\\ (z_n,w_{n-1},N) & i = n-1\end{cases}
\end{aligned}$$

head and tail of tapes For

$$x = x_1\#\dots\#x_r \quad x_i \in \mathbb{B}^*$$

we define

$$\begin{aligned}
hd(x) &= x_1\\
tail(x) &= x_2\#\dots\#x_r
\end{aligned}$$

machines $\boxed{tape\ 2 = hd(tape\ 1)}$ and $\boxed{tape\ 2 = tail(tape\ 1)}$
exercise

making k-tape machine M behave like s tape machine P on tapes

$$\{i_1,\dots,i_s\} \subset \{1,\dots,k\}$$

machine $\boxed{M = P(i_1,\dots i_s):}$ if

$$a \in A^k \ , \ a' = (a_{i_1},\dots,a_{i_s}) \ , \ \delta_P(z,a') = (z',b',r'_1,\dots,r'_s)$$

then

$$\delta_M(z,a) = (z',b,r)$$

where tape i_j of M behaves like tape j of P

$$b_{i_j} = b'_j \ , \ r_{i_j} = r'_j$$

and on other tapes $y \notin \{i_1,\dots,i_s\}$ nothing happens

$$b_j = a_j \ , \ r_j = N$$

making k-tape machine M behave like s tape machine P on tapes

$$\{i_1,\ldots,i_s\} \subset \{1,\ldots,k\}$$

machine $\boxed{M = P(i_1,\ldots i_s):}$ if

$$a \in A^k \ , \ a' = (a_{i_1},\ldots,a_{i_s}) \ , \ \delta_P(z,a') = (z',b',r'_1,\ldots,r'_s)$$

then

$$\delta_M(z,a) = (z',b,r)$$

where tape i_j of M behaves like tape j of P

$$b_{i_j} = b'_j \ , \ r_{i_j} = r'_j$$

and on other tapes $y \notin \{i_1,\ldots,i_s\}$ nothing happens

$$b_j = a_j \ , \ r_j = N$$

example $\boxed{tape\ 1 = tape\ 2}$
realize as

$$tape\ 2 = tape\ 1(2,1)$$

making k-tape machine M behave like s tape machine P on tapes

$$\{i_1,\dots,i_s\} \subset \{1,\dots,k\}$$

machine $\boxed{M = P(i_1,\dots i_s):}$ if

$$a \in A^k \;,\; a' = (a_{i_1},\dots,a_{i_s}) \;,\; \delta_P(z,a') = (z',b',r'_1,\dots,r'_s)$$

then

$$\delta_M(z,a) = (z',b,r)$$

where tape i_j of M behaves like tape j of P

$$b_{i_j} = b'_j \;,\; r_{i_j} = r'_j$$

and on other tapes $y \notin \{i_1,\dots,i_s\}$ nothing happens

$$b_j = a_j \;,\; r_j = N$$

example $\boxed{tape\ 1 = tape\ 2}$
realize as

$$tape\ 2 = tape\ 1(2,1)$$

concatenating machines machine $\boxed{Q = M;P:}$
w.l.o.g $Z \cap Z' = \emptyset$ and $z_{0,P} \notin Z_{E,P}$

$$\begin{aligned}
Z_Q &= Z_M \cup Z_P \\
z_{0,Q} &= z_{0,M} \\
Z_{E,Q} &= Z_{E,P} \\
\delta_Q(z,a) &= \begin{cases} \delta_M(z,a) & z \in Z_M \setminus Z_{E,M} \\ \delta_P(z_{0,,P},a) & z \in Z_{E,M} \\ \delta_P(z,a) & z \in Z_P \end{cases}
\end{aligned}$$

making k-tape machine M behave like s tape machine P on tapes

$$\{i_1,\dots,i_s\} \subset \{1,\dots,k\}$$

machine $\boxed{M = P(i_1,\dots i_s):}$ if

$$a \in A^k \ , \ a' = (a_{i_1},\dots,a_{i_s}) \ , \ \delta_P(z,a') = (z',b',r'_1,\dots,r'_s)$$

then

$$\delta_M(z,a) = (z',b,r)$$

where tape i_j of M behaves like tape j of P

$$b_{i_j} = b'_j \ , \ r_{i_j} = r'_j$$

and on other tapes $y \notin \{i_1,\dots,i_s\}$ nothing happens

$$b_j = a_j \ , \ r_j = N$$

example $\boxed{tape\ 1 = tape\ 2}$
realize as

$$tape\ 2 = tape\ 1(2,1)$$

concatenating machines machine $\boxed{Q = M;P:}$
w.l.o.g $Z \cap Z' = \emptyset$ and $z_{0,P} \notin Z_{E,P}$

$$\begin{aligned}
Z_Q &= Z_M \cup Z_P \\
z_{0,Q} &= z_{0,M} \\
Z_{E,Q} &= Z_{E,P} \\
\delta_Q(z,a) &= \begin{cases} \delta_M(z,a) & z \in Z_M \setminus Z_{E,M} \\ \delta_P(z_{0,,P},a) & z \in Z_{E,M} \\ \delta_P(z,a) & z \in Z_P \end{cases}
\end{aligned}$$

unrolling a finite loop for Machines M_i we abbreviate

$$M_1;\dots;M_k$$

as

$$\boxed{for\ i = 1\ to\ k\ do\ M_i}$$

testing tape 1 for all zeros: *regular* machine $\boxed{tape\ 1 = 0?}$

$$
\begin{aligned}
Z &= \{z_0, z_1, yes', no', yes, no\} \\
\delta(z_0, a) &= (no', a, L) \quad a \neq 0 \\
\delta(z_0; 0) &= (z_0, 0, R) \\
\delta(z_0, B) &= (yes'; B, L) \\
\delta(q, a) &= (q, a, L) \quad q \in \{yes', no'\}\ ,\ a \neq B \\
\delta(yes', B) &= (yes, B, R) \\
\delta(no', B) &= (no, B, R)
\end{aligned}
$$

testing tape 1 for all zeros: *regular* machine $\boxed{tape\ 1 = 0?}$

$$
\begin{aligned}
Z &= \{z_0, z_1, yes', no', yes, no\} \\
\delta(z_0, a) &= (no', a, L) \quad a \neq 0 \\
\delta(z_0; 0) &= (z_0, 0, R) \\
\delta(z_0, B) &= (yes'; B, L) \\
\delta(q, a) &= (q, a, L) \quad q \in \{yes', no'\}\ ,\ a \neq B \\
\delta(yes', B) &= (yes, B, R) \\
\delta(no', B) &= (no, B, R)
\end{aligned}
$$

while loop machine $\boxed{Q : while\ tape\ i \neq 0\ do\ M}$
regular machine M changes tape i. States differ from states of last machine.

$$Q : tape\ 1 = 0?;\ M;\ S$$

for all $z \in Z_{E,M}$ and $a \in A$

$$\delta_S(z, a) = \delta_{tape\ 1=0?}(z_0, a)$$

testing tape 1 for all zeros: *regular* machine $\boxed{tape\ 1 = 0?}$

$$
\begin{aligned}
Z &= \{z_0, z_1, yes', no', yes, no\} \\
\delta(z_0, a) &= (no', a, L) \quad a \neq 0 \\
\delta(z_0; 0) &= (z_0, 0, R) \\
\delta(z_0, B) &= (yes'; B, L) \\
\delta(q, a) &= (q, a, L) \quad q \in \{yes', no'\}\ ,\ a \neq B \\
\delta(yes', B) &= (yes, B, R) \\
\delta(no', B) &= (no, B, R)
\end{aligned}
$$

while loop machine $\boxed{Q : while\ tape\ i \neq 0\ do\ M}$
regular machine M changes tape i. States differ from states of last machine.

$$Q : tape\ 1 = 0?;\ M;\ S$$

for all $z \in Z_{E,M}$ and $a \in A$

$$\delta_S(z, a) = \delta_{tape\ 1=0?}(z_0, a)$$

4 functions computed by Turing machines

here: $bin(y)$ without leading zeros

def: function f_M computed by TM M Let

$$f : \mathbb{N}_0^r \to \mathbb{N}_0$$

we say that TM M computes f resp. $f = f_M$ is the function computed by M if for all

$$x = (x_1, \ldots, x_r) \in \mathbb{N}_0^r$$

machine M started with

$$bin(x_1)\#\ldots\#bin(x_r)$$

outputs

$$bin(f(x_1, \ldots, x_r))$$

f is TM-computable if $f = f_M$ for some TM M

5 μ-recursive functions are TM-computable

Lemma 1. *All constant functions c_s^r are computed by regular TM's.*

Proof.

$$erase\ tape\ 1;\ tape\ 1 = bin(s)$$

$\square$

5 μ-recursive functions are TM-computable

Lemma 1. *All constant functions c_s^r are computed by regular TM's.*

Proof.

$$erase\ tape\ 1;\ tape\ 1 = bin(s)$$

$\square$

Lemma 2. *The successor function is TM-computable by a regular TM:*

Proof.

$$tape\ 1 = tape\ 1\ + 1$$

$\square$

5 μ-recursive functions are TM-computable

Lemma 1. *All constant functions c^r_s are computed by regular TM's.*

Proof.

$$erase\ tape\ 1;\ tape\ 1 = bin(s)$$

$\square$

Lemma 2. *The successor function is TM-computable by a regular TM:*

Proof.

$$tape\ 1 = tape\ 1\ + 1$$

$\square$

Lemma 3. *all projections p^r_i are computed by regular TM's*

Proof.

$$tape\ 1 = tail(tape\ 1);\ \ldots;\ tape\ 1 = tail(tape\ 1);\quad (i-1 \text{ times});$$
$$tape\ 1 = hd(tape\ 1)$$

$\square$

5.1 function composition

Lemma 4. *If the following functions are all computable by regular TM's*

$$f : \mathbb{N}_0^r \to \mathbb{N} \text{ and } g_1, \ldots, g_r : \mathbb{N}_0^m \to \mathbb{N}_0$$

then also h is computable by a regular TM, where

$$h : \mathbb{N}_0^m \to \mathbb{N}_0$$

$$h(x) = f(g_1(x), \ldots, g_r(x))$$

5.1 function composition

Lemma 4. *If the following functions are all computable by regular TM's*

$$f : \mathbb{N}_0^r \to \mathbb{N} \text{ and } g_1, \dots, g_r : \mathbb{N}_0^m \to \mathbb{N}_0$$

then also h is computable by a regular TM, where

$$h : \mathbb{N}_0^m \to \mathbb{N}_0$$

$$h(x) = f(g_1(x), \dots, g_r(x))$$

- For all i let g_i be computed by k_i-tape machine G_i and f by k'-tape machine F. We compute h by k-tape TM M with

$$k = \max\{k_1, \dots k_r, k'\} + r$$

5.1 function composition

Lemma 4. *If the following functions are all computable by regular TM's*

$$f : \mathbb{N}_0^r \to \mathbb{N} \text{ and } g_1, \dots, g_r : \mathbb{N}_0^m \to \mathbb{N}_0$$

then also h is computable by a regular TM, where

$$h : \mathbb{N}_0^m \to \mathbb{N}_0$$

$$h(x) = f(g_1(x), \dots, g_r(x))$$

- For all i let g_i be computed by k_i-tape machine G_i and f by k'-tape machine F. We compute h by k-tape TM M with

$$k = \max\{k_1, \dots k_r, k'\} + r$$

- Let the input of tape 1 be

$$bin(x_1)\#\dots\#bin(x_r)$$

  and

$$x = (x_1, \dots, x_r) \in \mathbb{N}_0^r$$

5.1 function composition

Lemma 4. *If the following functions are all computable by regular TM's*

$$f : \mathbb{N}_0^r \to \mathbb{N} \text{ and } g_1, \ldots, g_r : \mathbb{N}_0^m \to \mathbb{N}_0$$

then also h is computable by a regular TM, where

$$h : \mathbb{N}_0^m \to \mathbb{N}_0$$

$$h(x) = f(g_1(x), \ldots, g_r(x))$$

- For all i let g_i be computed by k_i-tape machine G_i and f by k'-tape machine F. We compute h by k-tape TM M with

$$k = \max\{k_1, \ldots k_r, k'\} + r$$

  - Let the input of tape 1 be

$$bin(x_1)\#\ldots\#bin(x_r)$$

    and

$$x = (x_1, \ldots, x_r) \in \mathbb{N}_0^r$$

  - copy tape 1 to tapes $2, \ldots, r+1$; then erase tape 1

    $for\ i = 1\ to\ r\ \{tape\ i+1 = tape\ 1\};\ erase\ tape\ 1$

    We get the situation from table 1.

tape	content
1	$B \ldots B$
2	$bin(x_1)\#\ldots\#bin(x_r)$
	$\ldots$
$r+1$	$bin(x_1)\#\ldots\#bin(x_r)$
$r+2$	$B \ldots B$
	$\ldots$

Table 1: after copying input to tapes $2, \ldots, r+1$.

5.1 function composition

Lemma 4. *If the following functions are all computable by regular TM's*

$$f : \mathbb{N}_0^r \to \mathbb{N} \text{ and } g_1, \dots, g_r : \mathbb{N}_0^m \to \mathbb{N}_0$$

then also h is computable by a regular TM, where

$$h : \mathbb{N}_0^m \to \mathbb{N}_0$$

$$h(x) = f(g_1(x), \dots, g_r(x))$$

- For all i let g_i be computed by k_i-tape machine G_i and f by k'-tape machine F. We compute h by k-tape TM M with

$$k = \max\{k_1, \dots k_r, k'\} + r$$

- Let the input of tape 1 be

$$bin(x_1)\#\dots\#bin(x_r)$$

  and

$$x = (x_1, \dots, x_r) \in \mathbb{N}_0^r$$

- copy tape 1 to tapes $2, \dots, r+1$; then erase tape 1

  $for\ i = 1\ to\ r\ \{tape\ i+1 = tape\ 1\};\ erase\ tape\ 1$

  We get the situation from table 1.

tape	content
1	$B \dots B$
2	$bin(x_1)\#\dots\#bin(x_r)$
	$\dots$
$r+1$	$bin(x_1)\#\dots\#bin(x_r)$
$r+2$	$B \dots B$
	$\dots$

Table 1: after copying input to tapes $2, \dots, r+1$.

- for all $i = 1$ to r compute $tape\ i+1 = bin(g_i(x))$ on tapes $i+1, r+2, \dots, r+1+k_i$:

  $$for\ i = 1\ to\ k\ do\ \{G_i(i+1, r+2, \dots, r+1+k_i)\}$$

5.1 function composition

Lemma 4. *If the following functions are all computable by regular TM's*

$$f : \mathbb{N}_0^r \to \mathbb{N} \text{ and } g_1,\dots,g_r : \mathbb{N}_0^m \to \mathbb{N}_0$$

then also h is computable by a regular TM, where

$$h : \mathbb{N}_0^m \to \mathbb{N}_0$$

$$h(x) = f(g_1(x),\dots,g_r(x))$$

- For all i let g_i be computed by k_i-tape machine G_i and f by k'-tape machine F. We compute h by k-tape TM M with

$$k = \max\{k_1,\dots k_r,k'\} + r$$

- Let the input of tape 1 be

$$bin(x_1)\#\dots\#bin(x_r)$$

  and

$$x = (x_1,\dots,x_r) \in \mathbb{N}_0^r$$

- copy tape 1 to tapes $2,\dots,r+1$; then erase tape 1

  $for\ i = 1\ to\ r\ \{tape\ i+1 = tape\ 1\};\ erase\ tape\ 1$

  We get the situation from table 1.

tape	content
1	$B\dots B$
2	$bin(x_1)\#\dots\#bin(x_r)$
	$\dots$
$r+1$	$bin(x_1)\#\dots\#bin(x_r)$
$r+2$	$B\dots B$
	$\dots$

Table 1: after copying input to tapes $2,\dots,r+1$.

- for all $i = 1$ to r compute $tape\ i+1 = bin(g_i(x))$ on tapes $i+1, r+2,\dots,r+1+k_i$:

  $for\ i = 1\ to\ k\ do\ \{G_i(i+1,r+2,\dots,r+1+k_i)\}$

  We get the situation from table 2.

tape	content
1	$B\dots B$
2	$bin(g_1(x))$
	$\dots$
$r+1$	$bin(g_r(x))$
$r+2$	$B\dots B$
	$\dots$

Table 2: after copying input to tapes $2,\dots,r+1$.

5.1 function composition

Lemma 4. *If the following functions are all computable by regular TM's*

$$f : \mathbb{N}_0^r \to \mathbb{N} \text{ and } g_1, \ldots, g_r : \mathbb{N}_0^m \to \mathbb{N}_0$$

then also h is computable by a regular TM, where

$$h : \mathbb{N}_0^m \to \mathbb{N}_0$$

$$h(x) = f(g_1(x), \ldots, g_r(x))$$

- For all i let g_i be computed by k_i-tape machine G_i and f by k'-tape machine F. We compute h by k-tape TM M with

$$k = \max\{k_1, \ldots k_r, k'\} + r$$

- Let the input of tape 1 be

$$bin(x_1)\#\ldots\#bin(x_r)$$

and

$$x = (x_1, \ldots, x_r) \in \mathbb{N}_0^r$$

We get the situation from table 2.

tape	content
1	$B \ldots B$
2	$bin(g_1(x))$
	$\ldots$
$r+1$	$bin(g_r(x))$
$r+2$	$B \ldots B$
	$\ldots$

Table 2: after copying input to tapes $2, \ldots, r+1$.

5.1 function composition

Lemma 4. *If the following functions are all computable by regular TM's*

$$f : \mathbb{N}_0^r \to \mathbb{N} \text{ and } g_1, \dots, g_r : \mathbb{N}_0^m \to \mathbb{N}_0$$

then also h is computable by a regular TM, where

$$h : \mathbb{N}_0^m \to \mathbb{N}_0$$

$$h(x) = f(g_1(x), \dots, g_r(x))$$

- For all i let g_i be computed by k_i-tape machine G_i and f by k'-tape machine F. We compute h by k-tape TM M with

$$k = \max\{k_1, \dots k_r, k'\} + r$$

  - Let the input of tape 1 be

$$bin(x_1)\#\dots\#bin(x_r)$$

  and

$$x = (x_1, \dots, x_r) \in \mathbb{N}_0^r$$

We get the situation from table 2.

tape	content
1	$B \dots B$
2	$bin(g_1(x))$
	$\dots$
$r+1$	$bin(g_r(x))$
$r+2$	$B \dots B$
	$\dots$

Table 2: after copying input to tapes $2, \dots, r+1$.

- compute $tape\ 1 = bin(g_1(x))\#\dots\#bin(g_r(x))$

  $tape\ 1 = tape\ 2;\ erase\ tape\ 2;$
  $for\ i = 2\ to\ k\ do\ \{tape\ 1 = tape1\#tape\ 1+i;\ erase\ tape\ 1+i\}$

  We get the situation from table 3.

tape	content
1	$bin(g_1(x))\#\dots\#bin(g_r(x))$
2	$B \dots B$
	$\dots$

Table 3: after copying sequence of $bin(g_i(x))$ on tape 1.

5.1 function composition

Lemma 4. *If the following functions are all computable by regular TM's*

$$f : \mathbb{N}_0^r \to \mathbb{N} \text{ and } g_1, \ldots, g_r : \mathbb{N}_0^m \to \mathbb{N}_0$$

then also h is computable by a regular TM, where

$$h : \mathbb{N}_0^m \to \mathbb{N}_0$$

$$h(x) = f(g_1(x), \ldots, g_r(x))$$

- For all i let g_i be computed by k_i-tape machine G_i and f by k'-tape machine F. We compute h by k-tape TM M with

$$k = \max\{k_1, \ldots k_r, k'\} + r$$

- Let the input of tape 1 be

$$bin(x_1)\#\ldots\#bin(x_r)$$

and

$$x = (x_1, \ldots, x_r) \in \mathbb{N}_0^r$$

We get the situation from table 2.

tape	content
1	$B\ldots B$
2	$bin(g_1(x))$
	$\ldots$
$r+1$	$bin(g_r(x))$
$r+2$	$B\ldots B$
	$\ldots$

Table 2: after copying input to tapes $2, \ldots, r+1$.

- compute $tape\ 1 = bin(g_1(x))\#\ldots\#bin(g_r(x))$

$tape\ 1 = tape\ 2;\ erase\ tape\ 2;$
$for\ i = 2\ to\ k\ do\ \{tape\ 1 = tape1\#tape\ 1+i;\ erase\ tape\ 1+i\}$

We get the situation from table 3.

tape	content
1	$bin(g_1(x))\#\ldots\#bin(g_r(x))$
2	$B\ldots B$
	$\ldots$

Table 3: after copying sequence of $bin(g_i(x))$ on tape 1.

- compute result by: F

5.2 primitive recursion

Lemma 5.

If the following functions are computable by regular TM's

$$g : \mathbb{N}_0^r \to \mathbb{N}_0 \; , \; h : \mathbb{N}_0^{r+2} \to \mathbb{N}_0$$

then also f is computable by a regular TM, where

$$f : \mathbb{N}_0^{r+1} \to \mathbb{N}_0$$

$$\begin{aligned} f(0,x) &= g(x) \\ f(n+1,x) &= h(n, f(n,x), x) \end{aligned}$$

5.2 primitive recursion

Lemma 5.

If the following functions are computable by regular TM's

$$g : \mathbb{N}_0^r \to \mathbb{N}_0 \ , \ h : \mathbb{N}_0^{r+2} \to \mathbb{N}_0$$

then also f is computable by a regular TM, where

$$f : \mathbb{N}_0^{r+1} \to \mathbb{N}_0$$

$$\begin{aligned} f(0,x) &= g(x) \\ f(n+1,x) &= h(n, f(n,x), x) \end{aligned}$$

- Let g be computed by regular k-tape machine G, and let h be computed by regular k'-tape machine H. Compute f by the following s- tape machine with
$$s = \max\{k, k'\} + 3$$

- Let the input of tape 1 be
$$bin(n)\#bin(x_1)\#\ldots\#bin(x_r)$$
and
$$n \in \mathbb{N}_0 \ , \ x = (x_1, \ldots, x_r) \in \mathbb{N}_0^r$$

5.2 primitive recursion

Lemma 5.

If the following functions are computable by regular TM's

$$g : \mathbb{N}_0^r \to \mathbb{N}_0 \ , \ h : \mathbb{N}_0^{r+2} \to \mathbb{N}_0$$

then also f is computable by a regular TM, where

$$f : \mathbb{N}_0^{r+1} \to \mathbb{N}_0$$

$$\begin{aligned} f(0,x) &= g(x) \\ f(n+1,x) &= h(n, f(n,x), x) \end{aligned}$$

- Let g be computed by regular k-tape machine G, and let h be computed by regular k'-tape machine H. Compute f by the following s- tape machine with

$$s = \max\{k, k'\} + 3$$

- Let the input of tape 1 be

$$bin(n)\#bin(x_1)\#\ldots\#bin(x_r)$$

and

$$n \in \mathbb{N}_0 \ , \ x = (x_1, \ldots, x_r) \in \mathbb{N}_0^r$$

- with
$tape\ 2 = hd(tape\ 1);\ tape\ 3 = tail(tape(1));\ tape\ 1 = 0;$
$tape\ 4 = tape\ 3;\ G(4, \ldots k+3\}$

we get for the situation of table 4 for i=4

tape	content
1	0
2	$bin(n)$
3	$bin(x_1))\#\ldots\#bin(x_r)$
4	$bin(g(x))$
5	$B\ldots B$
	$\ldots$

Table 4: after copying sequence of $bin(g_i(x))$ on tape 1.

5.2 primitive recursion

Lemma 5.

If the following functions are computable by regular TM's

$$g : \mathbb{N}_0^r \to \mathbb{N}_0 \ , \ h : \mathbb{N}_0^{r+2} \to \mathbb{N}_0$$

then also f is computable by a regular TM, where

$$f : \mathbb{N}_0^{r+1} \to \mathbb{N}_0$$

$$\begin{aligned} f(0,x) &= g(x) \\ f(n+1,x) &= h(n, f(n,x), x) \end{aligned}$$

- Let g be computed by regular k-tape machine G, and let h be computed by regular k'-tape machine H. Compute f by the following s- tape machine with

$$s = \max\{k,k'\}+3$$

- Let the input of tape 1 be

$$bin(n)\#bin(x_1)\#\ldots\#bin(x_r)$$

and

$$n \in \mathbb{N}_0 \ , \ x = (x_1,\ldots,x_r) \in \mathbb{N}_0^r$$

- with
  $tape\ 2 = hd(tape\ 1);\ tape\ 3 = tail(tape(1));$
  $erase\ tape\ 1;\ tape\ 1 = 0;$
  $tape\ 4 = tape\ 3;\ G(4,\ldots k+3\}$

  we get for the situation of table 4 for i=4

tape	content
1	0
2	$bin(n)$
3	$bin(x_1))\#\ldots\#bin(x_r)$
4	$bin(g(x))$
5	$B\ldots B$
	$\ldots$

Table 4: after copying sequence of $bin(g_i(x))$ on tape 1.

- in the following while loop we maintain for $i = 0;\ldots,n$, that after i passses through the loop we have the situation of table 5. For $i = 0$ this is the case.

tape	content
1	$bin(i)$
2	$bin(n-i)$
3	$bin(x_1))\#\ldots\#bin(x_r)$
4	$bin(f(i,x))$
5	$B\ldots B$
	$\ldots$

Table 5: after executing the loop i times

5.2 primitive recursion

Lemma 5.

If the following functions are computable by regular TM's

$$g : \mathbb{N}_0^r \to \mathbb{N}_0 \ , \ h : \mathbb{N}_0^{r+2} \to \mathbb{N}_0$$

then also f is computable by a regular TM, where

$$f : \mathbb{N}_0^{r+1} \to \mathbb{N}_0$$

$$\begin{aligned} f(0,x) &= g(x) \\ f(n+1,x) &= h(n, f(n,x), x) \end{aligned}$$

- Let g be computed by regular k-tape machine G, and let h be computed by regular k'-tape machine H. Compute f by the following s- tape machine with

$$s = \max\{k,k'\} + 3$$

- Let the input of tape 1 be

$$bin(n)\#bin(x_1)\#\ldots\#bin(x_r)$$

and

$$n \in \mathbb{N}_0 \ , \ x = (x_1,\ldots,x_r) \in \mathbb{N}_0^r$$

- in the following while loop we maintain for $i = 0;\ldots,n$, that after i passses through the loop we have the situation of table 5. For $i = 0$ this is the case.

tape	content
1	$bin(i)$
2	$bin(n-i)$
3	$bin(x_1))\#\ldots\#bin(x_r)$
4	$bin(f(i,x))$
5	$B\ldots B$
	$\ldots$

Table 5: after executing the loop i times

This is achieved by

$while\ tape\ 2 \neq 0\ do$
$\{tape\ 1 = tape\ 1+1;\ tape\ 2 = tape\ 2-1;$
$tape\ 4 = tape\ 1\#tape\ 4;\ tape\ 4 = tape\ 4\#tape\ 3;$
$H(4,\ldots,k'+3)\}$

5.2 primitive recursion

Lemma 5.

If the following functions are computable by regular TM's

$$g : \mathbb{N}_0^r \to \mathbb{N}_0 \ , \ h : \mathbb{N}_0^{r+2} \to \mathbb{N}_0$$

then also f is computable by a regular TM, where

$$f : \mathbb{N}_0^{r+1} \to \mathbb{N}_0$$

$$\begin{aligned} f(0,x) &= g(x) \\ f(n+1,x) &= h(n, f(n,x), x) \end{aligned}$$

- Let g be computed by regular k-tape machine G, and let h be computed by regular k'-tape machine H. Compute f by the following s- tape machine with

$$s = \max\{k, k'\} + 3$$

- Let the input of tape 1 be

$$bin(n)\#bin(x_1)\#\ldots\#bin(x_r)$$

and

$$n \in \mathbb{N}_0 \ , \ x = (x_1, \ldots, x_r) \in \mathbb{N}_0^r$$

- in the following while loop we maintain for $i = 0; \ldots, n$, that after i passses through the loop we have the situation of table 5. For $i = 0$ this is the case.

tape	content
1	$bin(i)$
2	$bin(n-i)$
3	$bin(x_1))\#\ldots\#bin(x_r)$
4	$bin(f(i,x))$
5	$B\ldots B$
	$\ldots$

Table 5: after executing the loop i times

This is achieved by

while $tape\ 2 \neq 0$ *do*
$\{tape\ 1 = tape\ 1 + 1;\ tape\ 2 = tape\ 2 - 1;$
$tape\ 4 = tape\ 1\#tape\ 4;\ tape\ 4 = tape\ 4\#tape\ 3;$
$H(4, \ldots, k'+3)\}$

- When the loop exits with $tape\ 2 = 0$ we have $tape\ 1 = bin(n)$ and the result is on tape 4. We copy the result on tape 1 and clean up tapes 2,3 and 4 in order to get a regular machine

$$tape\ 1 \ = tape\ 4;\ erase\ tape\ 2;\ erase\ tape\ 3;\ erase\ tape4$$

5.3 unbounded μ-operator

Lemma 6. *if* $f : \mathbb{N}_0^{r+1} \to \mathbb{N}_0$ *is computable by a regular Turing machine , then also*

$$\mu f : \mathbb{N}_0^{r+1} \to \mathbb{N}_0$$

is computable by a regular Turing machine, where

$$\mu f(n,x) = \begin{cases} \min\{m \ : \ f(m,x) = 0\} & \textit{if it exists} \\ \Omega & \textit{(undefined) otherwise} \end{cases}$$

5.3 unbounded μ-operator

Lemma 6. *if* $f : \mathbb{N}_0^{r+1} \to \mathbb{N}_0$ *is computable by a regular Turing machine , then also*

$$\mu f : \mathbb{N}_0^{r+1} \to \mathbb{N}_0$$

is computable by a regular Turing machine, where

$$\mu f(n,x) = \begin{cases} \min\{m \ : \ f(m,x) = 0\} & \textit{if it exists} \\ \Omega & \textit{(undefined) otherwise} \end{cases}$$

- Let F be computed by regular k-tape machine F. Compute μf by the $(k+2$-tape machine described below.

- With

  $$x = (x_1, \ldots, x_r) \in \mathbb{N}_0^r$$

  the computation starts with

  $$bin(x_1)\# \ldots \# bin(x_r)$$

  on tape 1

5.3 unbounded μ-operator

Lemma 6. *if* $f:\mathbb{N}_0^{r+1}\rightarrow\mathbb{N}_0$ *is computable by a regular Turing machine , then also*

$$\mu f:\mathbb{N}_0^{r+1}\rightarrow\mathbb{N}_0$$

is computable by a regular Turing machine, where

$$\mu f(n,x)=\begin{cases}\min\{m\ :\ f(m,x)=0\} & \textit{if it exists}\\ \Omega & \textit{(undefined) otherwise}\end{cases}$$

- Let F be computed by regular k-tape machine F. Compute μf by the $(k+2$-tape machine described below.

- With

  $$x=(x_1,\dots,x_r)\in\mathbb{N}_0^r$$

  the computation starts with

  $$bin(x_1)\#\dots\#bin(x_r)$$

  on tape 1

- set tape $2=0$; set tape 3 to 0, then append the input; evaluate $f(0,x)$ on tape 3

  $$tape\ 2=0;tape\ 3=tape\ 2;tape\ 3=tape\ 3\#tape\ 1;F(3,\dots,k+2)$$

  For $m=0$ we get the situation of table 6

tape	content
1	$bin(x_1))\#\dots\#bin(x_r)$
2	$bin(m)$
3	$bin(f(m,x))$
4	$B\dots B$
	$\dots$

Table 6: after executing the loop m times

5.3 unbounded μ-operator

Lemma 6. *if $f : \mathbb{N}_0^{r+1} \to \mathbb{N}_0$ is computable by a regular Turing machine , then also*

$$\mu f : \mathbb{N}_0^{r+1} \to \mathbb{N}_0$$

is computable by a regular Turing machine, where

$$\mu f(n,x) = \begin{cases} \min\{m \ : \ f(m,x) = 0\} & \textit{if it exists} \\ \Omega & \textit{(undefined) otherwise} \end{cases}$$

- Let F be computed by regular k-tape machine F. Compute μf by the $(k+2$-tape machine described below.

- With

  $$x = (x_1, \dots, x_r) \in \mathbb{N}_0^r$$

  the computation starts with

  $$bin(x_1)\#\dots\#bin(x_r)$$

  on tape 1

- set tape $2 = 0$; set tape 3 to 0, then append the input; evaluate $f(0,x)$ on tape 3

  $$tape\ 2 = 0; tape\ 3 = tape\ 2; tape\ 3 = tape\ 3\#tape\ 1; F(3,\dots,k+2)$$

  For $m = 0$ we get the situation of table 6

tape	content
1	$bin(x_1))\#\dots\#bin(x_r)$
2	$bin(m)$
3	$bin(f(m,x))$
4	$B\dots B$
	$\dots$

Table 6: after executing the loop m times

- maintaining the situation of table 6 we compute in the following loop successively $f(m,x)$ for $m = 1,2,\dots$ until we find a solution of the equation $f(m,x) = 0$. If no solution exists, this loop will not terminate.

  $while\ tape\ 3 \neq 0$
  $\{tape\ 2 = tape\ 2 + 1; erase\ tape\ 3$
  $tape\ 3 = tape\ 2;\ tape\ 3 = tape\ 3\#tape\ 1;$
  $F(3,\dots,k+2)\}$

5.3 unbounded μ-operator

Lemma 6. *if* $f : \mathbb{N}_0^{r+1} \to \mathbb{N}_0$ *is computable by a regular Turing machine , then also*

$$\mu f : \mathbb{N}_0^{r+1} \to \mathbb{N}_0$$

is computable by a regular Turing machine, where

$$\mu f(n,x) = \begin{cases} \min\{m \ : \ f(m,x) = 0\} & \textit{if it exists} \\ \Omega & \textit{(undefined) otherwise} \end{cases}$$

- Let F be computed by regular k-tape machine F. Compute μf by the $(k+2$-tape machine described below.
- With

  $$x = (x_1, \ldots, x_r) \in \mathbb{N}_0^r$$

  the computation starts with

  $$bin(x_1)\#\ldots\#bin(x_r)$$

  on tape 1

- set tape $2 = 0$; set tape 3 to 0, then append the input; evaluate $f(0,x)$ on tape 3

  $$tape\ 2 = 0; tape\ 3 = tape\ 2; tape\ 3 = tape\ 3\#tape\ 1; F(3,\ldots,k+2)$$

  For $m = 0$ we get the situation of table 6

tape	content
1	$bin(x_1))\#\ldots\#bin(x_r)$
2	$bin(m)$
3	$bin(f(m,x))$
4	$B\ldots B$
	$\ldots$

Table 6: after executing the loop m times

- maintaining the situation of table 6 we compute in the following loop successively $f(m,x)$ for $m = 1,2,\ldots$ until we find a solution of the equation $f(m,x) = 0$. If no solution exists, this loop will not terminate.

  $while\ tape\ 3 \neq 0$
  $\{tape\ 2 = tape\ 2 + 1; erase\ tape\ 3$
  $tape\ 3 = tape\ 2;\ tape\ 3 = tape\ 3\#tape\ 1;$
  $F(3,\ldots,k+2)\}$

- if a solution m is found the loop terminates with $bin(m)$ on tape 2. In order to make the machine regular we copy it on tape 1 and clean up tapes 2 and 3

  $tape\ 1 = tape\ 2;\ erase\ tape\ 2;\ erase\ tape\ 3$