A deterministic two-way multi-head finite automaton can be converted into a reversible one with the same number of heads

Kenichi Morita
Hiroshima University

The Fourth Workshop on Reversible Computation
(RC 2012), Copenhagen, July 2, 2012

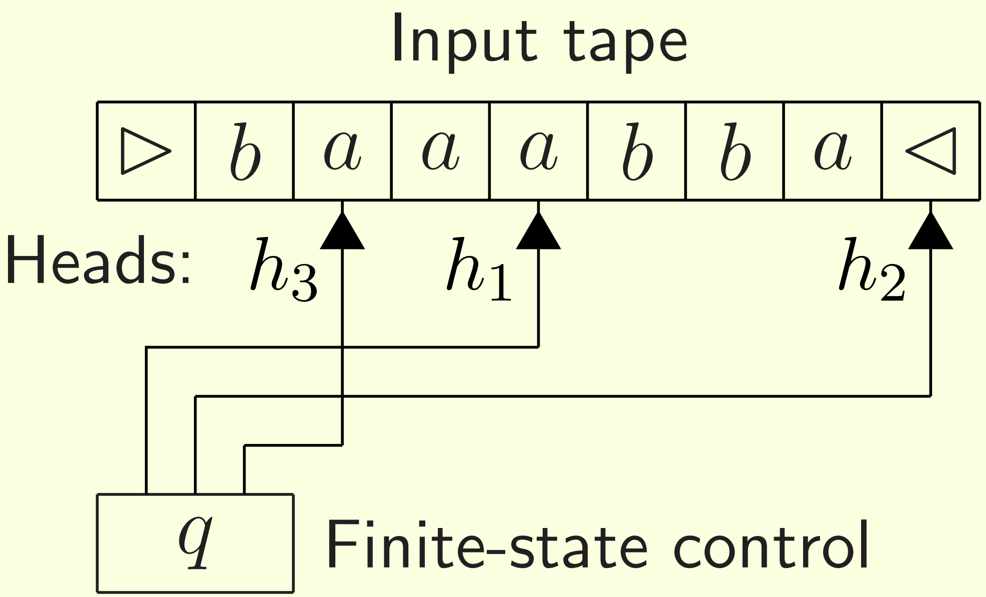
Contents

1. Introduction:
   A multi-head finite automaton (MFA)
2. Converting a deterministic MFA into
   a reversible one
3. Applying the conversion method to
   Turing machines

1. Introduction:
A multi-head finite automaton

Multi-head finite automaton (MFA)

- It is a simple classical model of an acceptor for a formal language.
- It consists of a finite-state control, an input tape, and k read-only heads (MFA(k)).



Past studies on reversible MFAs

- A two-way reversible MFA was introduced, and its basic properties were shown. [Morita, 2011]
- The class of two-way reversible MFAs is exactly characterized by the class of deterministic (and reversible) logarithmic space. [Axelsen, 2012]
- A one-way reversible MFA was studied, and its accepting capability was investigated. [Kutrib, Malcher, 2012]

Formal definition of a two-way MFA(k)

$$M = (Q, \Sigma, k, \delta, \triangleright, \triangleleft, q_0, A, R)$$

Q: a nonempty finite set of states

Σ: a nonempty finite set of input symbols

k: a number of heads ($k \in \{1, 2, \dots\}$)

$\triangleright, \triangleleft$: left and right endmarkers ($\triangleright, \triangleleft \notin \Sigma$)

q_0: the initial state ($q_0 \in Q$)

A: a set of accepting states ($A \subset Q$)

R: a set of rejecting states ($R \subset Q$, $A \cap R = \emptyset$)

δ: a transition relation, which is a subset of
$Q \times ((\Sigma \cup \{\triangleright, \triangleleft\})^k \cup \{-1, 0, +1\}^k) \times Q$

The transition relation δ of an MFA(k)

- δ is a set of “triples” of the form $[p,\ \mathbf{x},\ q]$.

  p : a present state ($p \in Q$)
  $\mathbf{x}$: symbols read ($\mathbf{x} \in (\Sigma \cup \{\triangleright, \triangleleft\})^k$), or
  shift directions ($\mathbf{x} \in \{-1, 0, +1\}^k$)
  q : a next state ($q \in Q$)

- $[p, \mathbf{s}, q]$ is called a read-rule if $\mathbf{s} \in (\Sigma \cup \{\triangleright, \triangleleft\})^k$.
- $[p, \mathbf{d}, q]$ is called a shift-rule if $\mathbf{d} \in \{-1, 0, +1\}^k$.

Note: Quintuple formulation is also used: $[p,\ \mathbf{s},\ \mathbf{d},\ q]$

Determinism of an MFA M

- An MFA M is called a deterministic MFA iff

$$\begin{aligned}&\forall\ r_1 = [p, \mathbf{x}, q] \in \delta,\ \forall\ r_2 = [p', \mathbf{x}', q'] \in \delta\\ &\quad((r_1 \neq r_2\ \wedge\ p = p')\ \Rightarrow\\ &\qquad(\mathbf{x} \notin \{-1, 0, +1\}^k\ \wedge\ \mathbf{x}' \notin \{-1, 0, +1\}^k\\ &\qquad\ \wedge\ \mathbf{x} \neq \mathbf{x}'))\end{aligned}$$

- It means that for every pair of rules r_1 and r_2, if the present states of them are the same, then

(1) r_1 and r_2 must be read-rules, and

(2) the symbols $\mathbf{x}$ and $\mathbf{x}'$ must be different.

Reversibility of an MFA M

- An MFA M is called a reversible MFA iff

$$\begin{aligned}&\forall\ r_1 = [p, \mathbf{x}, q] \in \delta,\ \forall\ r_2 = [p', \mathbf{x}', q'] \in \delta\\ &\quad ((r_1 \neq r_2\ \wedge\ q = q')\ \Rightarrow\\ &\quad\quad (\mathbf{x} \notin \{-1, 0, +1\}^k\ \wedge\ \mathbf{x}' \notin \{-1, 0, +1\}^k\\ &\quad\quad\ \wedge\ \mathbf{x} \neq \mathbf{x}'))\end{aligned}$$

- It means that for every pair of rules r_1 and r_2, if the next states of them are the same, then

(1) r_1 and r_2 must be read-rules, and

(2) the symbols $\mathbf{x}$ and $\mathbf{x}'$ must be different.

Notations for deterministic and reversible MFAs

- DMFA: Irreversible and deterministic MFA.
- RDMFA: Reversible and deterministic MFA.
- DMFA(k): DMFA with k heads.
- RDMFA(k): RDMFA with k heads.

Note: We do not consider a nondeterministic MFA hereafter.

Example: An RMFA(2) that accepts all strings of length 2^m $(m = 0, 1, \ldots)$ [Morita, 2011]

$$M_{2^m} = (\{q_0, q_1, \ldots, q_5, q_a, q_r\}, \{1\}, 2, \delta_{2^m}, \triangleright, \triangleleft, q_0, \{q_a\}, \{q_r\})$$

$$\begin{array}{rll}
\delta_{2^m} = \{ & [q_0, [\triangleright, \triangleright], [0, +], q_1], & \\
& [q_1, [\triangleright, 1], [0, +], q_1], & [q_1, [\triangleright, \triangleleft], [+, -], q_2], \\
& [q_2, [1, 1], [0, -], q_3], & [q_2, [1, \triangleright], [-, +], q_4], \\
& [q_2, [\triangleleft, \triangleright], [0, 0], q_r], & \\
& [q_3, [1, 1], [+, -], q_2], & [q_3, [1, \triangleright], [-, 0], q_5], \\
& [q_4, [1, 1], [-, +], q_4], & [q_4, [\triangleright, 1], [+, -], q_2], \\
& [q_5, [\triangleright, \triangleright], [0, 0], q_a], & [q_5, [1, \triangleright], [0, 0], q_r] \}
\end{array}$$

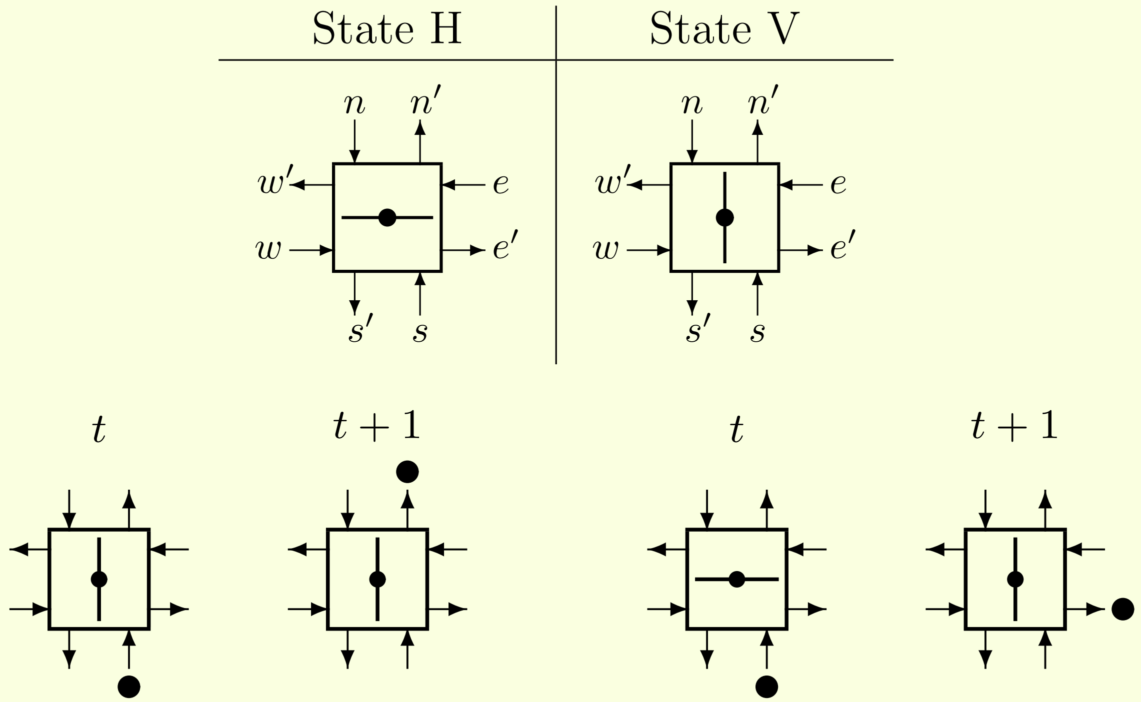
- M_{2^m} divides n by 2 repeatedly and checks if the remainders are all 0s till the dividend becomes 1.

t	state	tape
0	q_0	$\triangleright$ 1 1 1 1 1 1 1 1 $\triangleleft$
40	q_a	$\triangleright$ 1 1 1 1 1 1 1 1 $\triangleleft$

t	state	tape
0	q_0	$\triangleright$ 1 1 1 1 1 1 1 1 1 1 $\triangleleft$
36	q_r	$\triangleright$ 1 1 1 1 1 1 1 1 1 1 $\triangleleft$

An RDMFA can be realized as a garbage-less reversible logic circuit

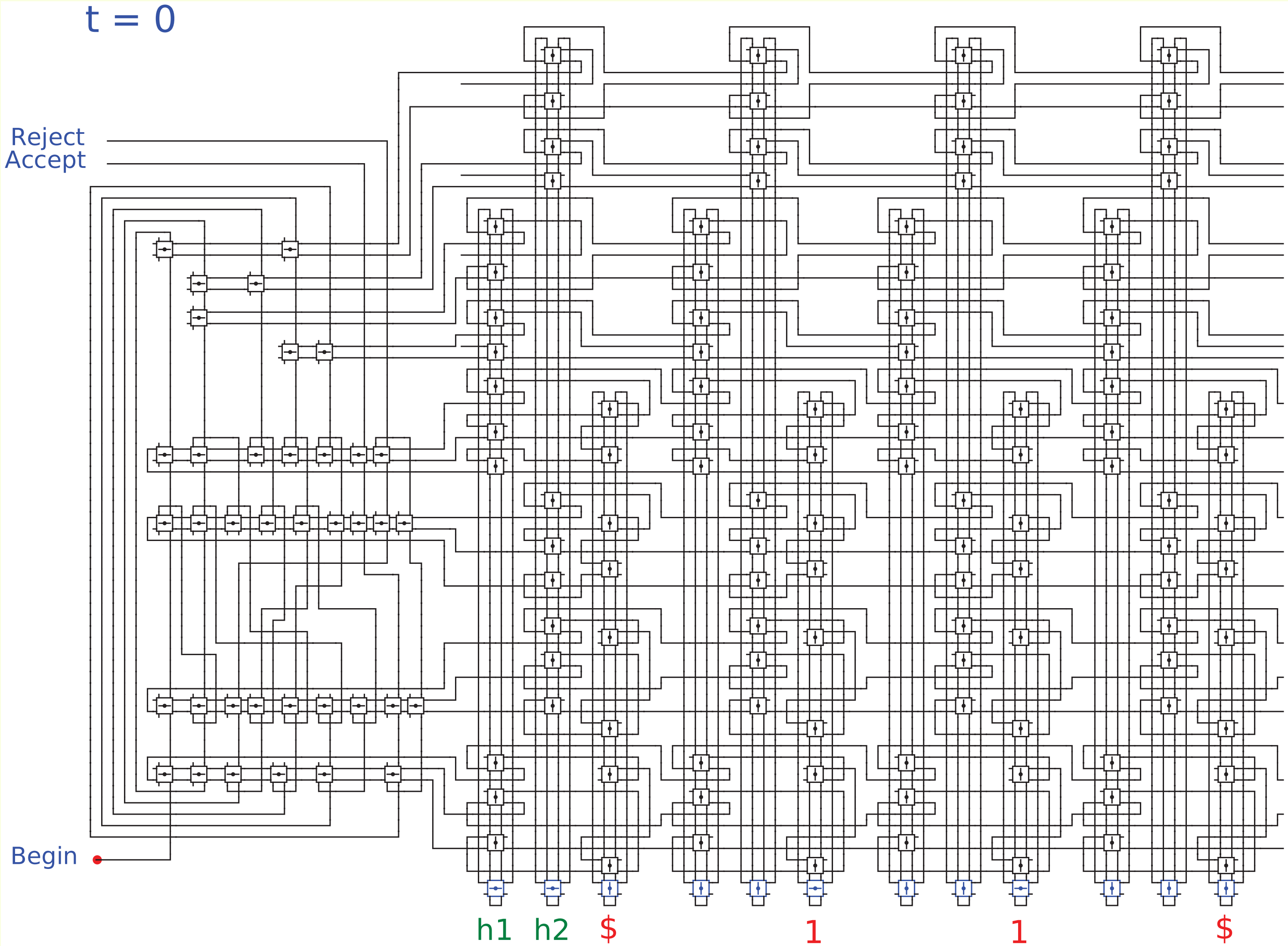
- A rotary element (RE) is a reversible logic element with 2 states and 4 symbols. [Morita, 2001]



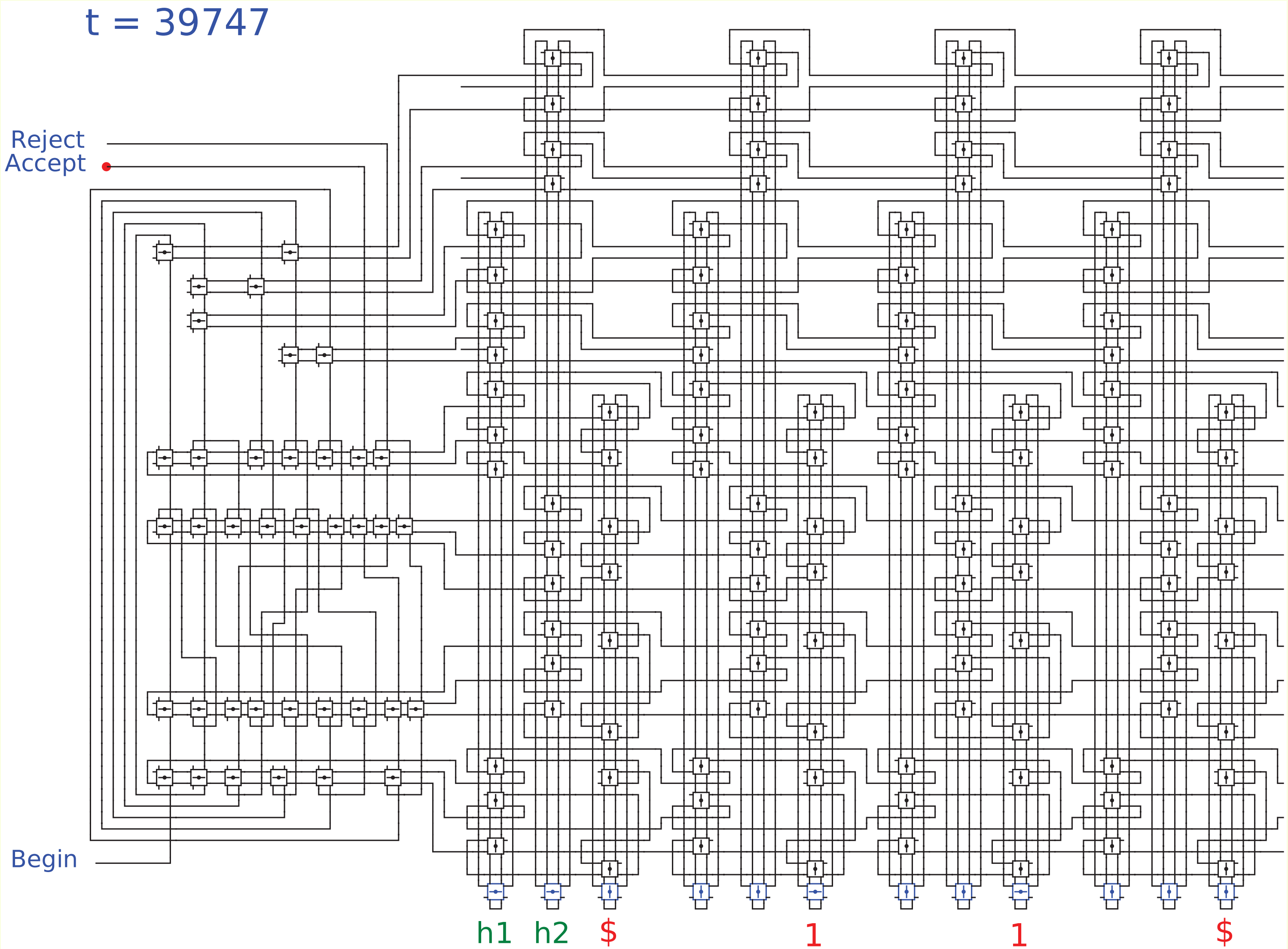
(a) parallel case (b) orthogonal case

Reversible logic circuit realizing M_{2^m}

— Initial configuration for the input $n = 2$ —



Reversible logic circuit realizing M_{2^m}
— Final configuration for the input $n = 2$ —



2. Converting a deterministic MFA into a reversible one

A DMFA(k) is simulated by an RDMFA(k)

Theorem 1 For any DMFA(k) M, we can construct an equivalent RDMFA(k) $M^\dagger$. Hence,

$$\mathcal{L}[\text{RDMFA}(k)] = \mathcal{L}[\text{DMFA}(k)].$$

Remark: $\mathcal{L}[\text{RD1MFA}(k)] \subsetneq \mathcal{L}[\text{D1MFA}(k)]$.

[Kutrib, Malcher, 2012]

Notation:

$\mathcal{L}[\mathcal{M}]$: The class of languages accepted by $\mathcal{M}$.

1MFA: A one-way MFA

Assumptions on DMFA for proving Theorem 1

(M1) The initial state q_0 does not appear as the third component of any rule in δ.

(M2) All the accepting and rejecting states are halting states.

(M3) Every states other than the initial state appears as the third component of some rule in δ.

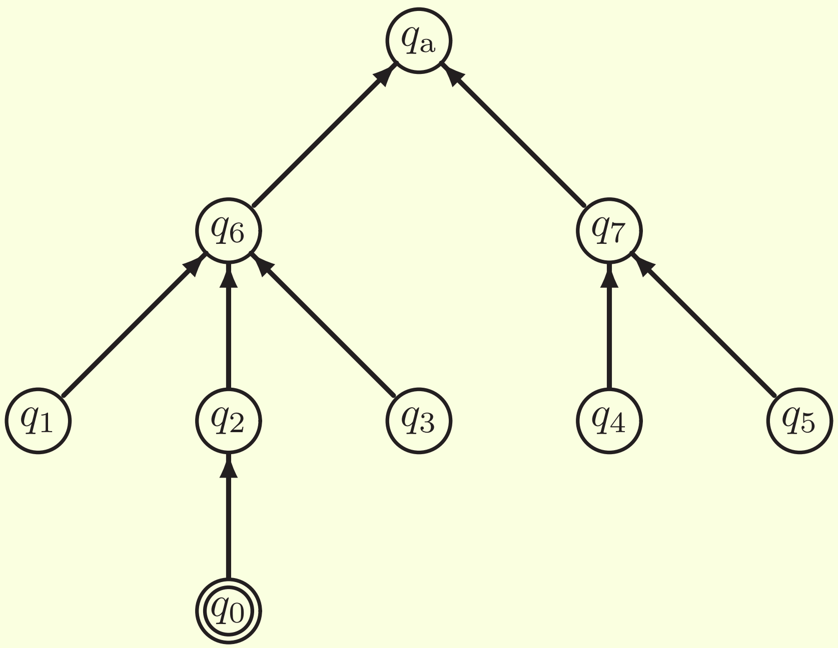
(M4) The DMFA performs read and shift operations alternately.

(M5) Each head must not go beyond the endmarkers both in forward and backward computation.

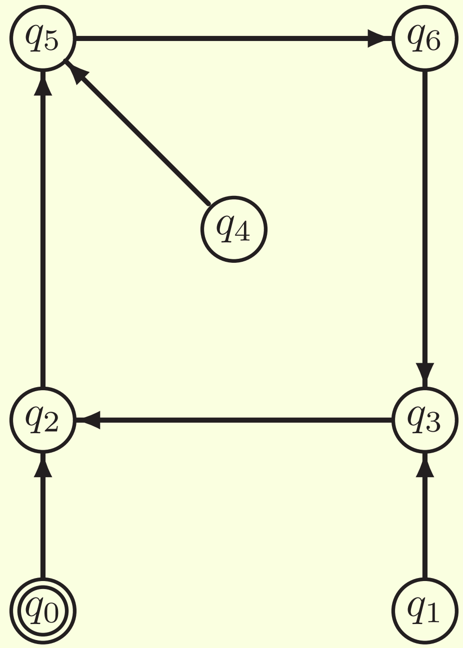
It is easy to modify a DMFA to satisfy them.

Proof outline of Theorem 1

- RDMFA(k) $M^\dagger$ traverses a computation graph of M using additional states.



A case M halts.



A case M loops.

Note: Each node represents a configuration of M. But, here, only a state of the finite-state control is written.

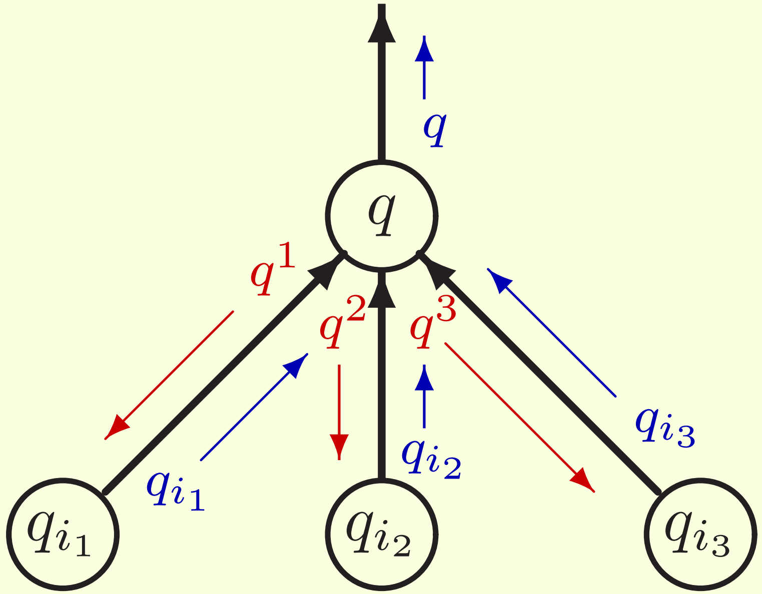
Traversing a computation graph reversibly

- $M^\dagger$ has the following states for each q of M.

  q is for the forward simulation.

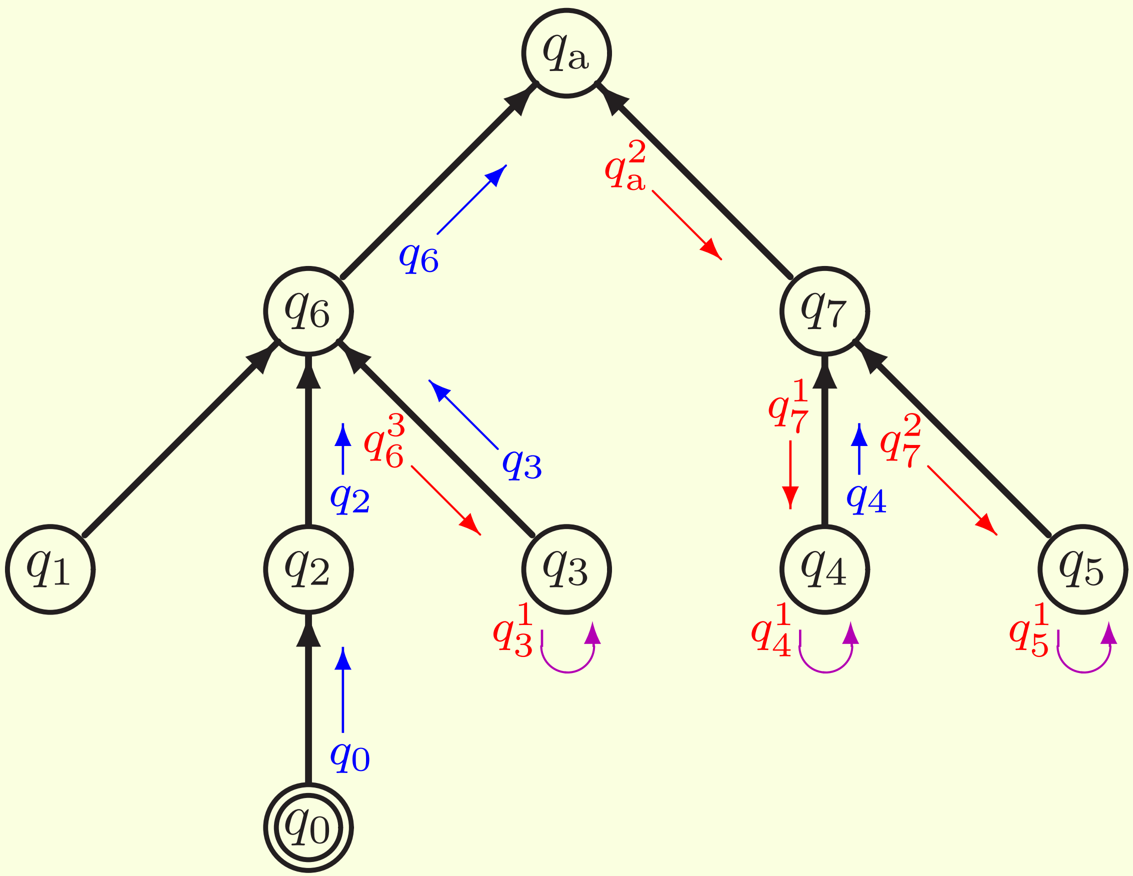
  q^j is for the backward simulation, where

  j is used to distinguish the incoming edges.



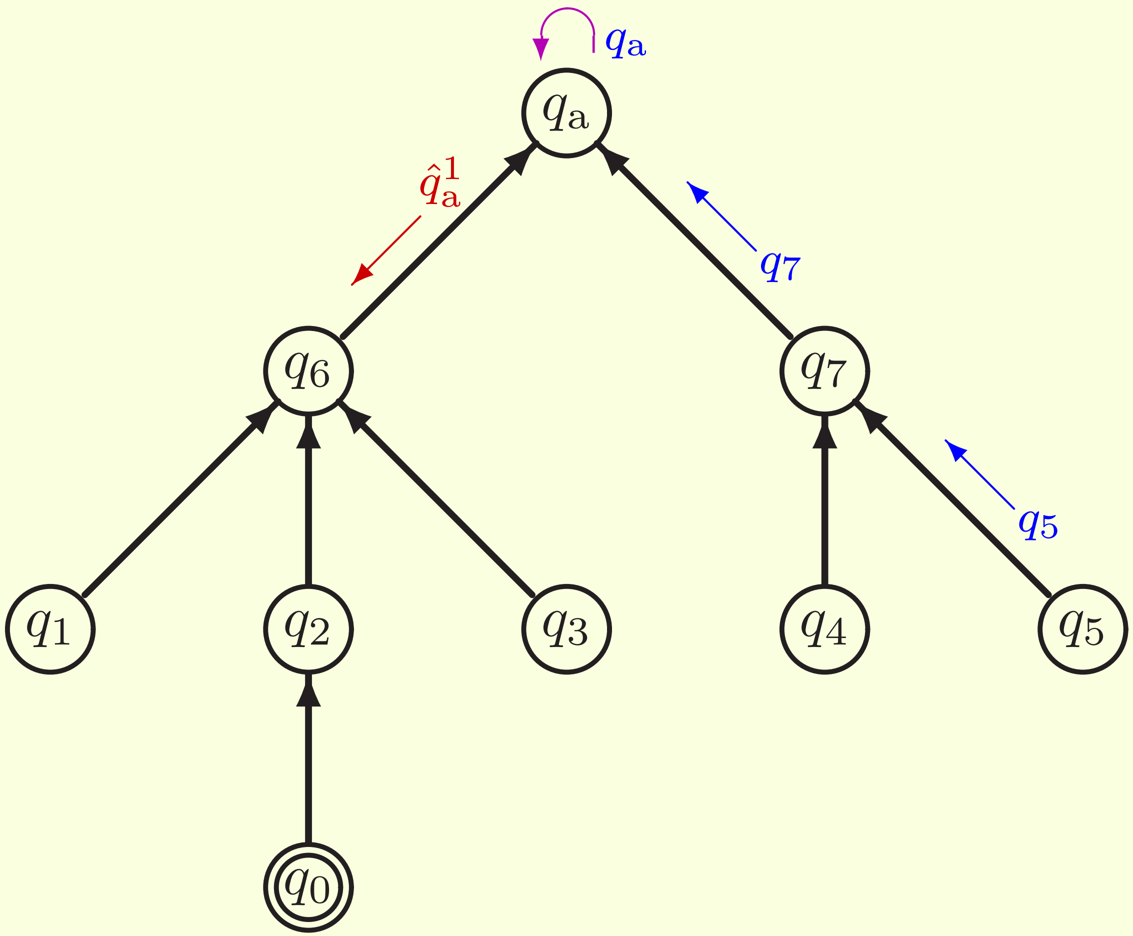
Irreversible transitions of M.

The case M halts in an accepting state (1)



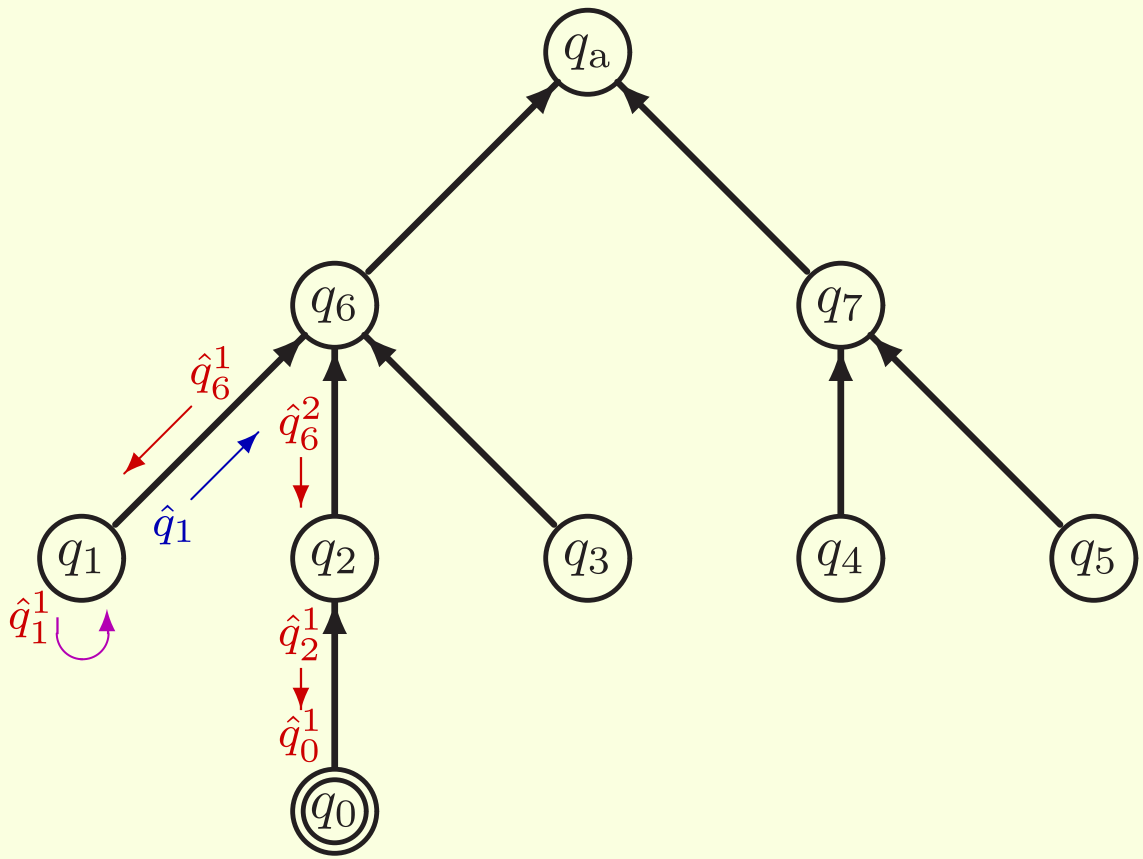
$M^\dagger$ starts to traverse the computation graph from the initial configuration of M.

The case M halts in an accepting state (2)



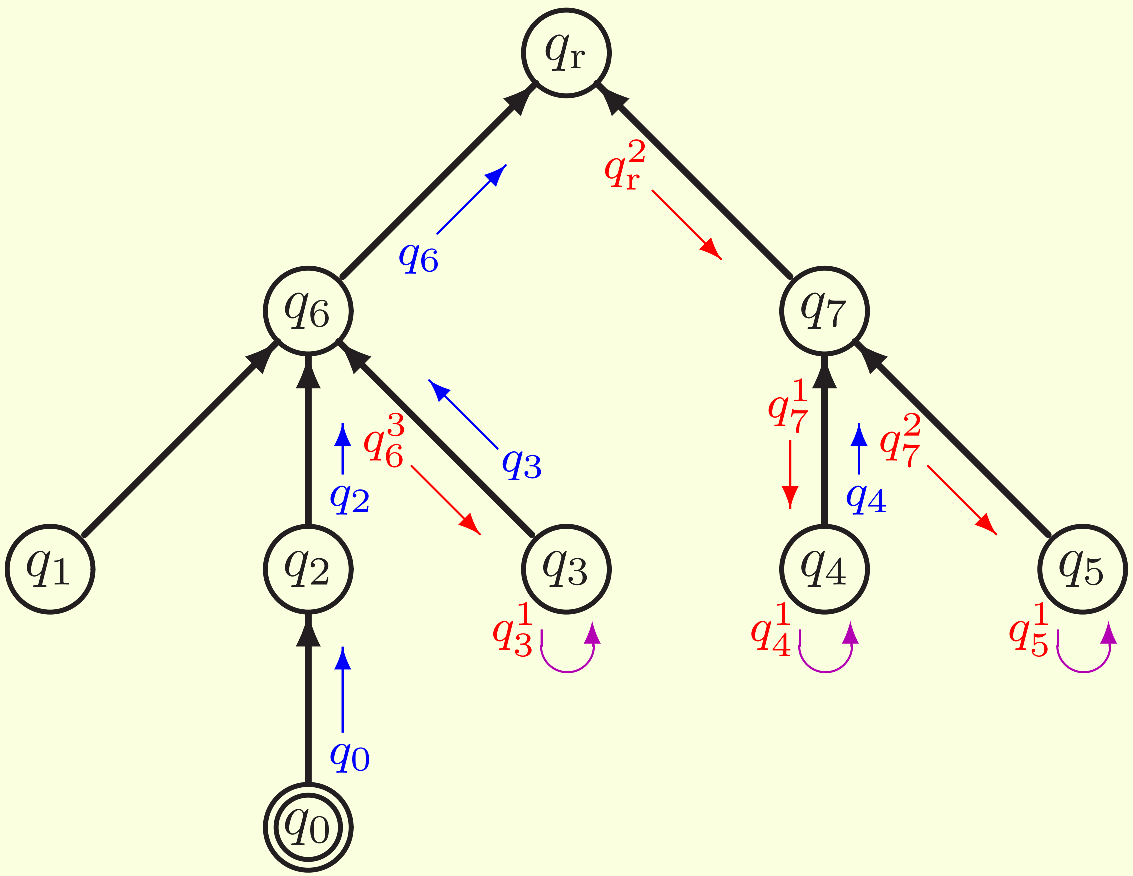
If M enters an accepting state q_{a}, then $M^{\dagger}$ keeps the fact by the states of the form $\widehat{q}$.

The case M halts in an accepting state (3)



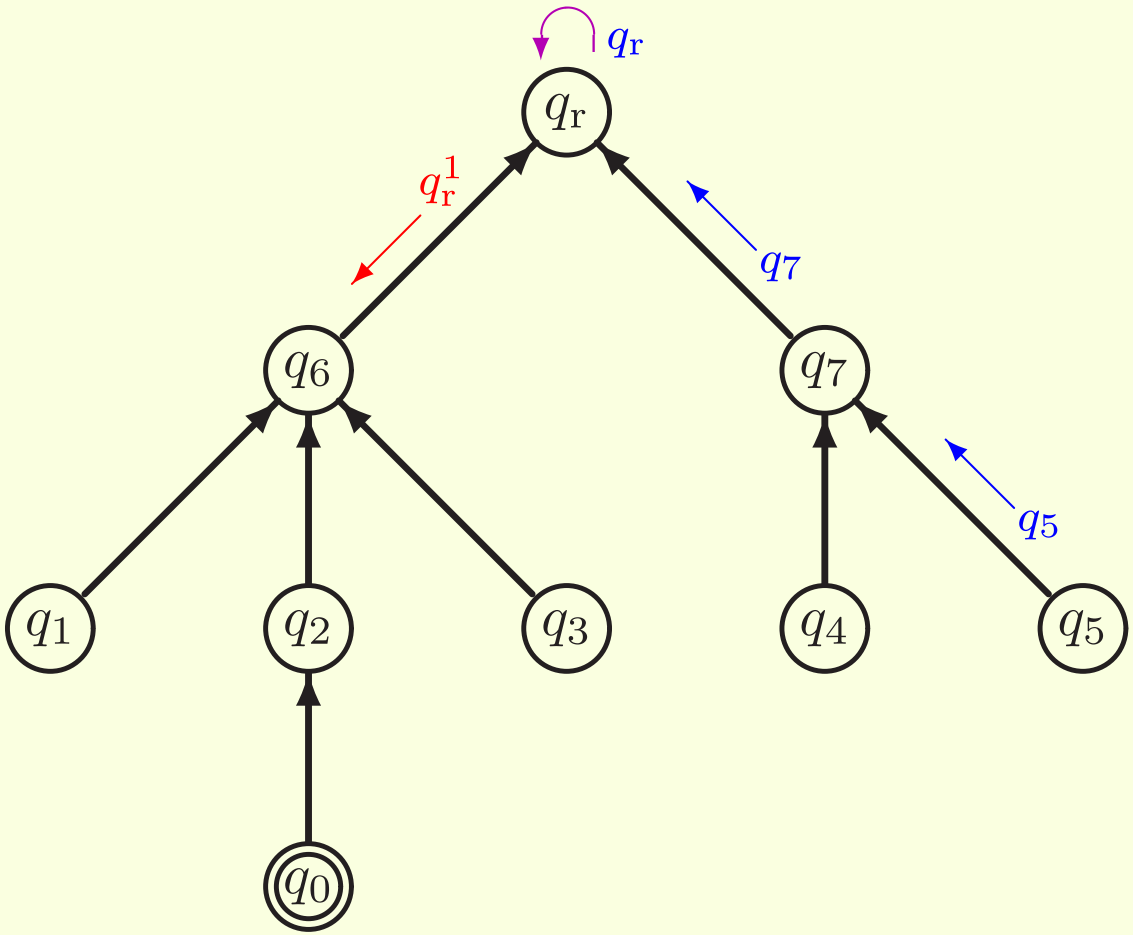
$M^\dagger$ finally goes back to M's initial configuration in the accepting state $\hat{q}_0^1$.

The case M halts in a non-accepting state (1)



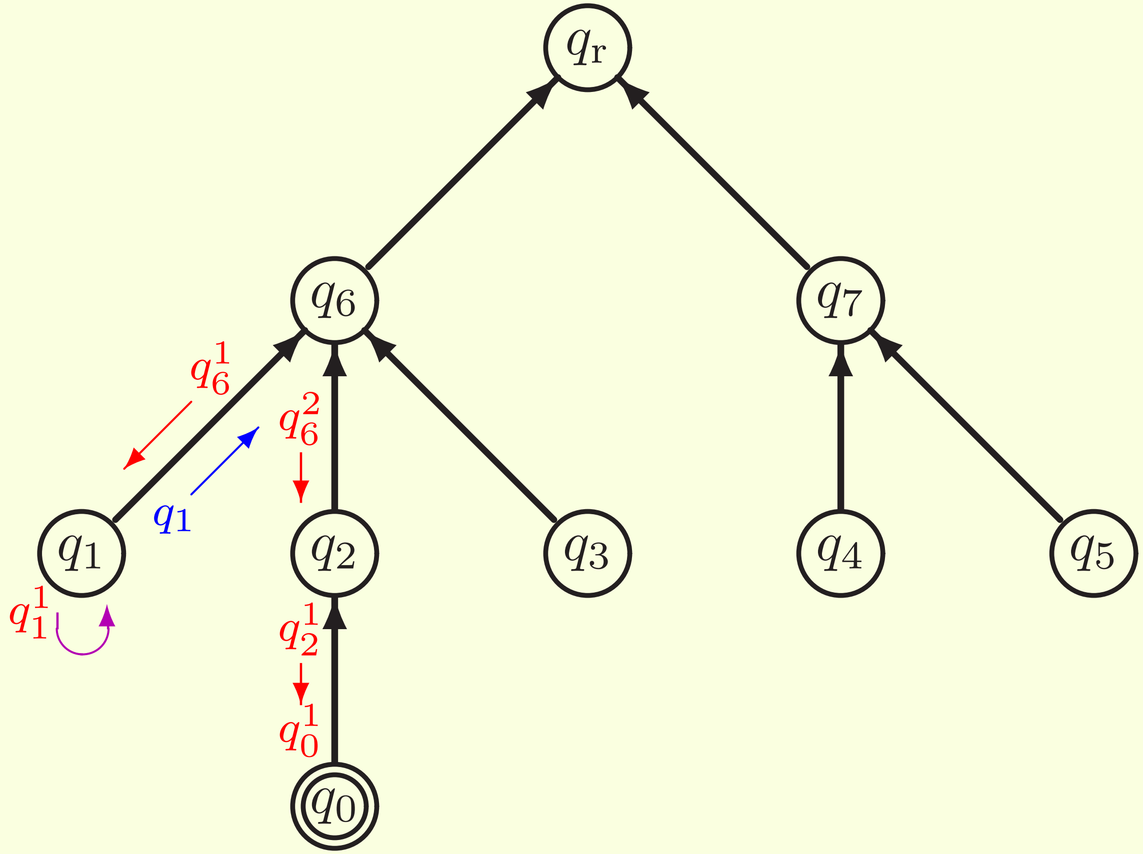
$M^\dagger$ starts to traverse the computation graph from the initial configuration of M.

The case M halts in a non-accepting state (2)



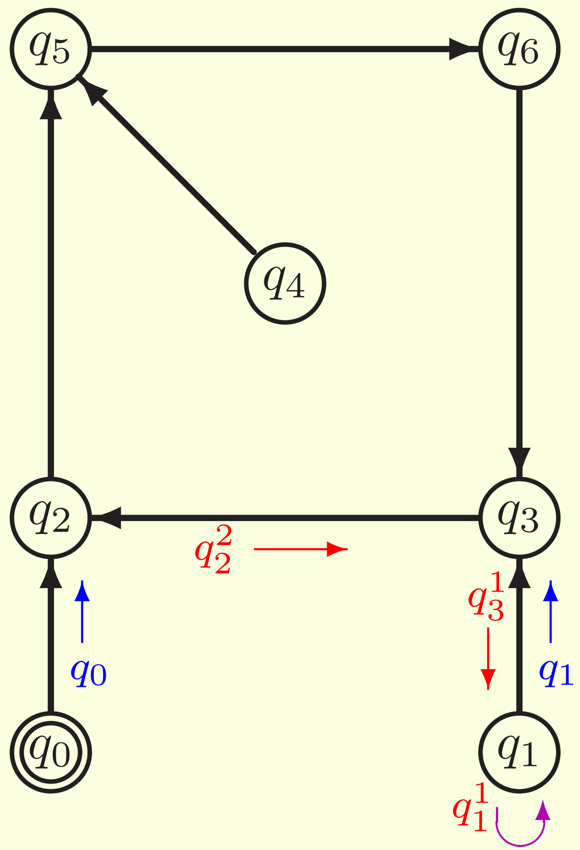
Since M does not enter an accepting state, $M^\dagger$ uses only the states without "$\hat{\ }$".

The case M halts in a non-accepting state (3)



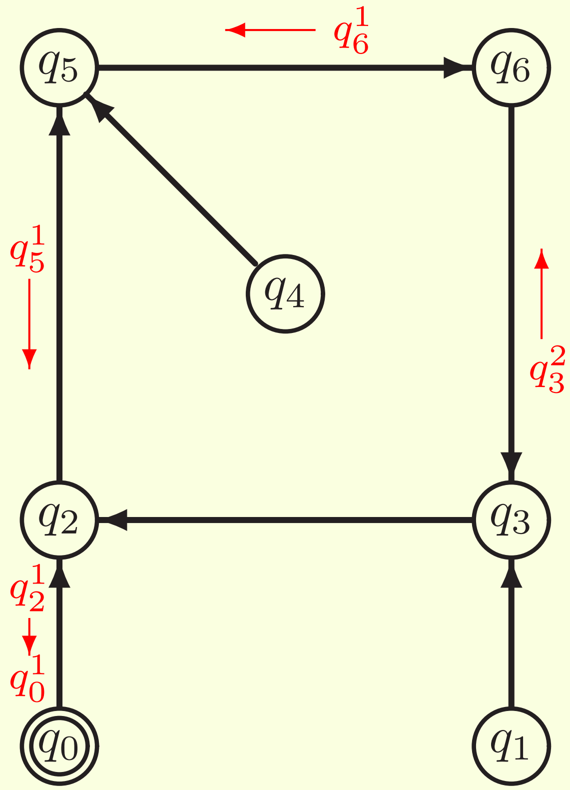
$M^\dagger$ finally goes back to M's initial configuration in the rejecting state q_0^1.

The case M loops (1)



$M^\dagger$ starts to traverse the computation graph from the initial configuration of M.

The case M loops (2)



Though it is not a tree, $M^\dagger$ finally goes back to M's initial configuration in the rejecting state q_0^1. This is because an RDMFA *always halts*.

An RDMFA always halts

Lemma 1 [Morita, 2011] If M is an RDMFA, then M eventually halts for any input w.

Note: Here, we assume the condition (M1) that the initial state q_0 of M does not appear as the third component of any rule of M (i.e., q_0 has no predecessor state).

Example of an irreversible DMFA(3)

The following irreversible DMFA(3) M_p accepts all strings whose length is a prime number.

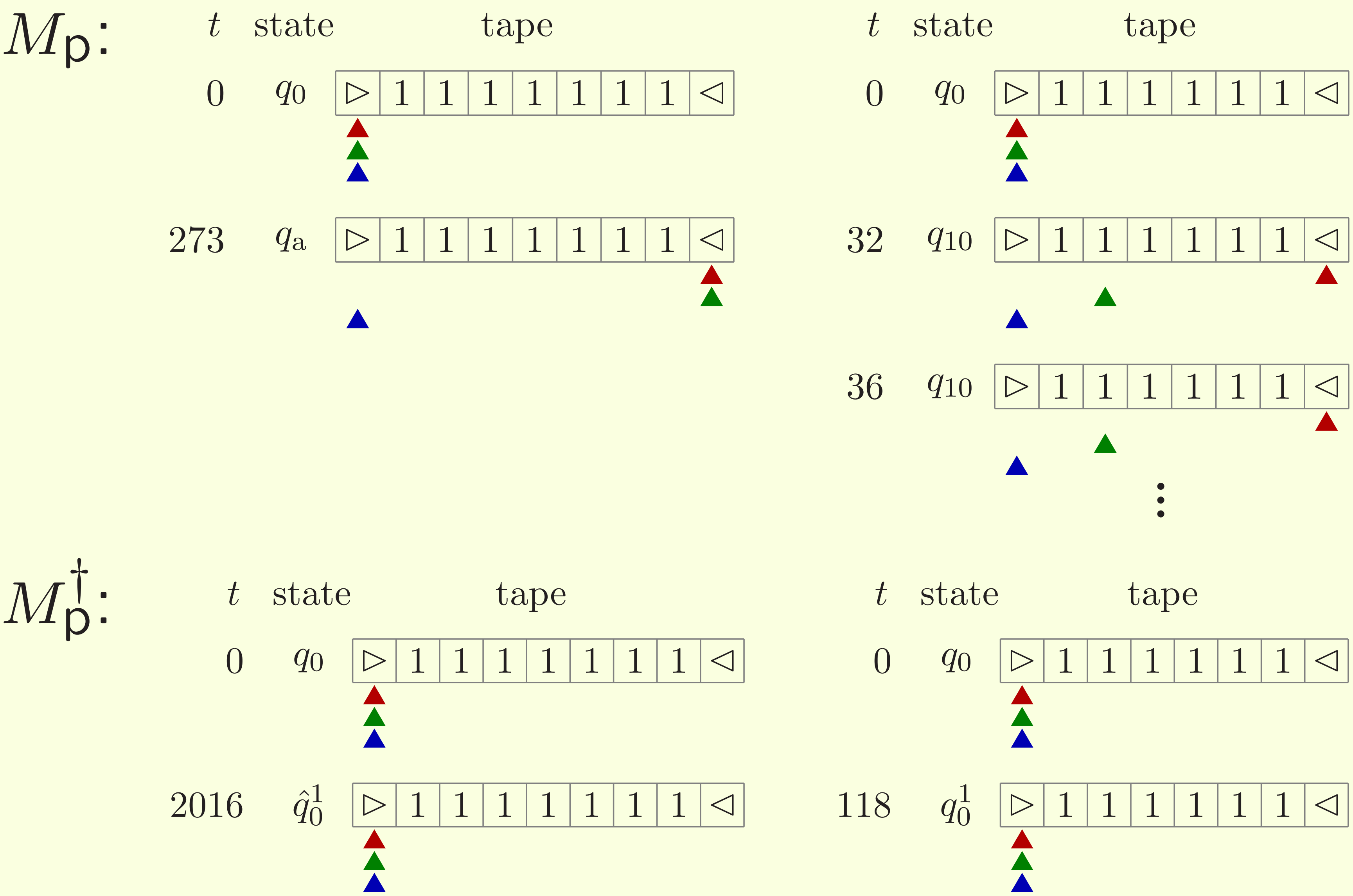
$$
\begin{array}{rcl}
M_\mathrm{p} & = & (Q, \{1\}, 3, \delta, \triangleright, \triangleleft, q_0, \{q_\mathrm{a}\}, \{\ \}) \\
Q & = & \{q_0, q_1, \ldots, q_{16}, q_\mathrm{a}\} \\
\delta & = & \{\ [q_0, [\triangleright, \triangleright, \triangleright], q_1],\ \ [q_1, [0, +, 0], q_2],\ \ [q_2, [\triangleright, 1, \triangleright], q_3],\ \ [q_3, [0, +, 0], q_4], \\
& & [q_4, [\triangleright, 1, \triangleright], q_5],\ \ [q_5, [+, -, +], q_6],\ \ [q_6, [1, 1, 1], q_5],\ \ [q_6, [1, \triangleright, 1], q_7], \\
& & [q_6, [\triangleleft, 1, 1], q_9],\ \ [q_6, [\triangleleft, \triangleright, 1], q_9],\ \ [q_7, [0, +, -], q_8],\ \ [q_8, [1, 1, 1], q_7], \\
& & [q_8, [1, 1, \triangleright], q_5],\ \ [q_9, [0, +, -], q_{10}],\ \ [q_{10}, [\triangleleft, 1, \triangleright], q_{14}],\ \ [q_{10}, [\triangleleft, 1, 1], q_{11}], \\
& & [q_{11}, [-, +, -], q_{13}],\ \ [q_{12}, [-, 0, 0], q_{13}],\ \ [q_{13}, [1, 1, 1], q_{11}],\ \ [q_{13}, [1, 1, \triangleright], q_{12}], \\
& & [q_{13}, [\triangleright, 1, \triangleright], q_3],\ \ [q_{14}, [0, +, 0], q_{15}],\ \ [q_{15}, [\triangleleft, \triangleleft, \triangleright], q_\mathrm{a}],\ \ [q_{15}, [\triangleleft, 1, \triangleright], q_{16}], \\
& & [q_{16}, [0, -, 0], q_{10}]\ \}
\end{array}
$$

t	state	tape								
0	q_0	$\triangleright$	1	1	1	1	1	1	1	$\triangleleft$
273	q_a	$\triangleright$	1	1	1	1	1	1	1	$\triangleleft$

An RDMFA(3) $M_{\mathsf{p}}^{\dagger}$ that simulates M_{p}

$$
\begin{array}{rcl}
M_{\mathsf{p}}^{\dagger} &=& (Q^{\dagger}, \{1\}, 3, \delta^{\dagger}, \rhd, \lhd, q_0, \{\hat{q}_0^1\}, \{q_0^1\}) \\
Q^{\dagger} &=& \{q, \hat{q}, q^1, \hat{q}^1 \mid q \in Q\} \cup \{\ q_3^2, q_{10}^2, q_{13}^2, \hat{q}_3^2, \hat{q}_{10}^2, \hat{q}_{13}^2\ \} \\
\delta^{\dagger} &=& \delta_1 \cup \cdots \cup \delta_6 \cup \hat{\delta}_1 \cup \cdots \cup \hat{\delta}_5 \cup \delta_{\mathsf{a}} \cup \delta_{\mathsf{r}} \\
\delta_1 &=& \{\, [q_1, [0,+,0], q_2],\ \ [q_3, [0,+,0], q_4],\ \ [q_5, [+,-,+], q_6],\ \ [q_7, [0,+,-], q_8], \\
&& \ [q_9, [0,+,-], q_{10}^2],\ [q_{11}, [-,+,-], q_{13}^2],\ [q_{12}, [-,0,0], q_{13}],\ [q_{14}, [0,+,0], q_{15}], \\
&& \ [q_{16}, [0,-,0], q_{10}]\ \} \\
\delta_2 &=& \{\, [q_0, [\rhd,\rhd,\rhd], q_1],\ \ [q_2, [\rhd,1,\rhd], q_3^2],\ \ [q_4, [\rhd,1,\rhd], q_5],\ \ [q_6, [1,1,1], q_5], \\
&& \ [q_6, [1,\rhd,1], q_7],\ \ [q_6, [\lhd,1,1], q_9],\ \ [q_6, [\lhd,\rhd,1], q_9],\ \ [q_8, [1,1,1], q_7], \\
&& \ [q_8, [1,1,\rhd], q_5],\ \ [q_{10}, [\lhd,1,\rhd], q_{14}],\ [q_{10}, [\lhd,1,1], q_{11}],\ [q_{13}, [1,1,1], q_{11}], \\
&& \ [q_{13}, [1,1,\rhd], q_{12}],\ [q_{13}, [\rhd,1,\rhd], q_3],\ \ [q_{15}, [\lhd,\lhd,\rhd], q_{\mathsf{a}}],\ [q_{15}, [\lhd,1,\rhd], q_{16}]\, \} \\
\delta_3 &=& \{\, [q_2^1, [0,-,0], q_1^1],\ \ [q_4^1, [0,-,0], q_3^1],\ \ [q_6^1, [-,+,-], q_5^1],\ \ [q_8^1, [0,-,+], q_7^1], \\
&& \ [q_{10}^1, [0,-,+], q_9^1],\ [q_{10}^2, [0,+,0], q_{16}^1],\ [q_{13}^1, [+,-,+], q_{11}^1],\ [q_{13}^2, [+,0,0], q_{12}^1], \\
&& \ [q_{15}^1, [0,-,0], q_{14}^1]\ \} \\
\delta_4 &=& \{\, [q_1^1, [\rhd,\rhd,\rhd], q_0^1],\ \ [q_3^1, [\rhd,1,\rhd], q_2^1],\ \ [q_3^2, [\rhd,1,\rhd], q_{13}^1],\ \ [q_5^1, [\rhd,1,\rhd], q_4^1], \\
&& \ [q_5^1, [1,1,1], q_6^1],\ \ [q_5^1, [1,1,\rhd], q_8^1],\ \ [q_7^1, [1,\rhd,1], q_6^1],\ \ [q_7^1, [1,1,1], q_8^1], \\
&& \ [q_9^1, [\lhd,1,1], q_6^1],\ \ [q_9^1, [\lhd,\rhd,1], q_6^1],\ \ [q_{11}^1, [\lhd,1,1], q_{10}^1],\ [q_{11}^1, [1,1,1], q_{13}^1], \\
&& \ [q_{12}^1, [1,1,\rhd], q_{13}^1],\ [q_{14}^1, [\lhd,1,\rhd], q_{10}^1],\ [q_{16}^1, [\lhd,1,\rhd], q_{15}^1],\ [q_{\mathsf{a}}^1, [\lhd,\lhd,\rhd], q_{15}^1]\, \} \\
\delta_5 &=& \{\, [q_1^1, [\rhd,\rhd,1], q_1],\ [q_1^1, [\rhd,\rhd,\lhd], q_1],\ \ \ldots\ \ , [q_{16}^1, [\lhd,\lhd,\lhd], q_{16}]\, \} \\
\hat{\delta}_i &=& \{\, [\hat{p}, \mathbf{x}, \hat{q}] \mid [p, \mathbf{x}, q] \in \delta_i \,\}\ (i = 1, \ldots, 5) \\
\delta_6 &=& \{\, [q_2, [\rhd,\rhd,\rhd], q_2^1],\ [q_2, [\rhd,\rhd,1], q_2^1],\ \ \ldots\ \ , [q_{15}, [\lhd,\lhd,\lhd], q_{15}^1]\, \} \\
\delta_{\mathsf{a}} &=& \{\, [q_{\mathsf{a}}, [0,0,0], \hat{q}_{\mathsf{a}}^1]\, \} \\
\delta_{\mathsf{r}} &=& \{\ \}
\end{array}
$$

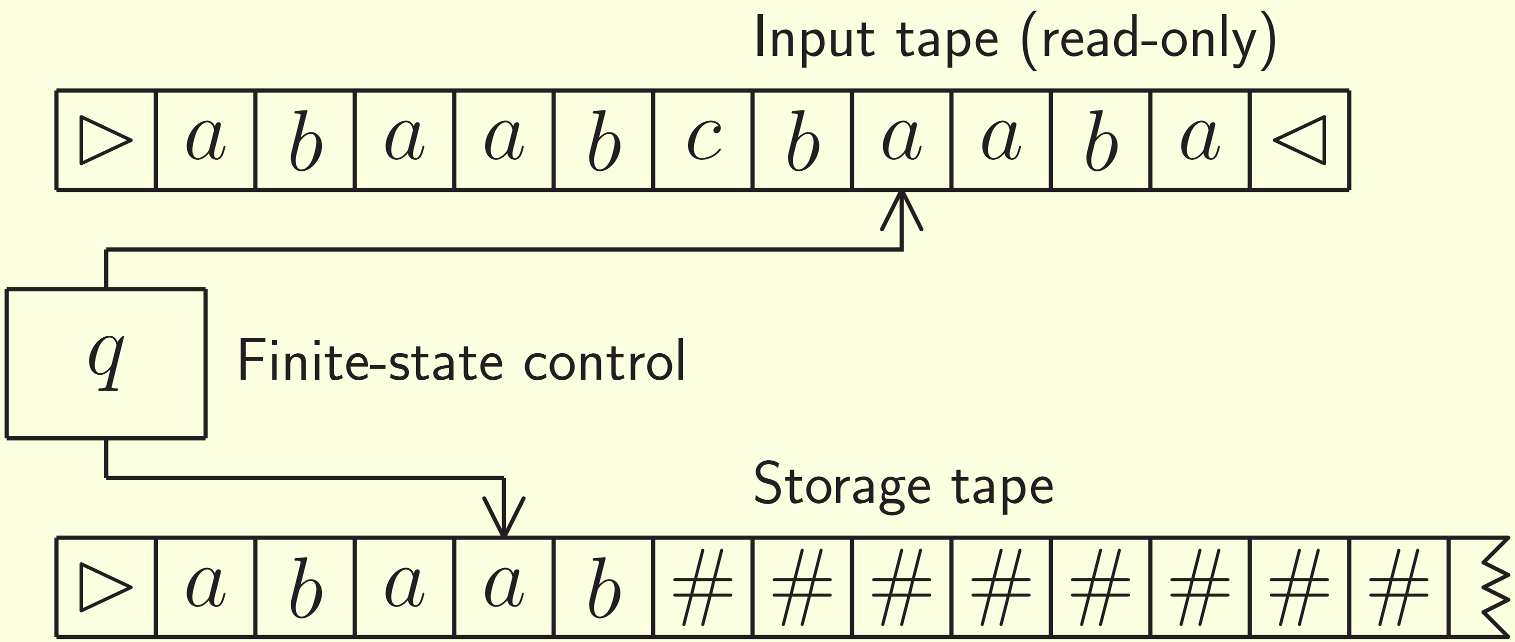
Simulating the DMFA M_{p} by the RDMFA $M_{\mathsf{p}}^{\dagger}$



3. Applying the conversion method to Turing machines

Two-tape Turing machine (TM)

- A model suited for studying space complexity.
- It consists of a finite-state control, a read-only input tape, a storage tape, and two heads.



Relation between DSPACE($s(n)$) and RDSPACE($s(n)$)

Proposition [Lange, McKenzie, Tapp, 2000]

$$\mathrm{DSPACE}(s(n)) = \mathrm{RDSPACE}(s(n))$$

- (R)DSPACE($s(n)$) : The class of languages accepted by an $s(n)$ space-bounded (R)DTM.
  n is the length of the input, and $s(n)$ is a space function.
- But, the simulation method given by them is rather complex.

The method of converting DMFAs to RDMFAs can be applied to DTMs simply

Theorem 2 For any DTM T, we can construct an equivalent RDTM $T^\dagger$ such that the following holds.

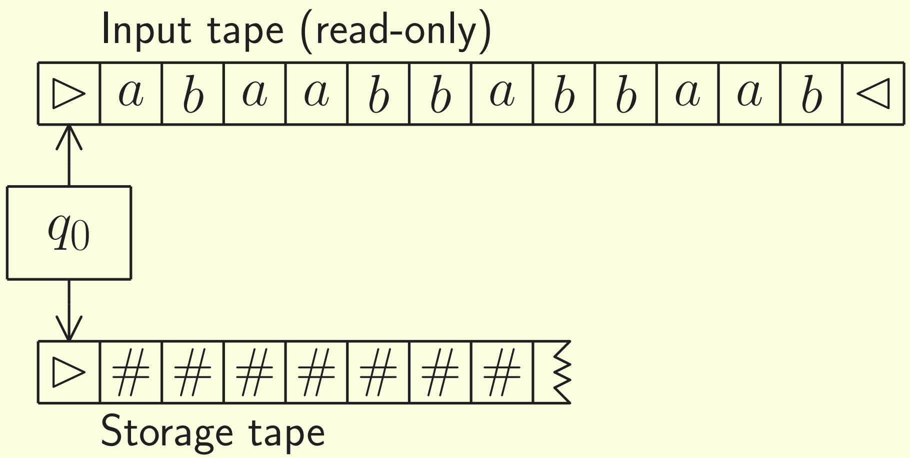
1. $T^\dagger$ uses exactly the same numbers of storage tape squares and tape symbols as T. (Thus, it is a bit stronger result than that of Lange et al.)
2. $T^\dagger$ with $w \in \Sigma^*$ always halts, provided that T with w uses finitely many storage squares. (We need not know T's space function $s(n)$.)

Example of an irreversible DTM

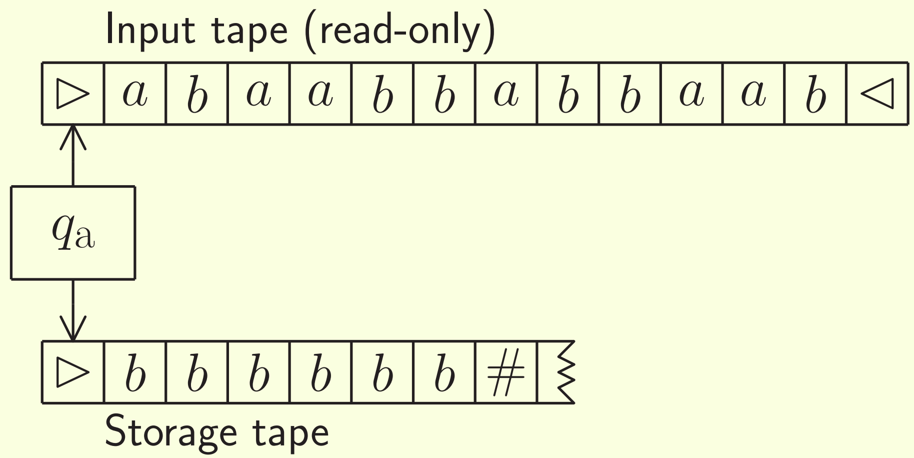
The DTM T_{eq} accepts all strings over $\{a, b\}^*$ such that the number of a's is equal to that of b's.

$$
\begin{array}{rcl}
T_{\mathrm{eq}} & = & (Q, \{a, b\}, \{a, b\}, \delta, \triangleright, \triangleleft, q_0, \#, \{q_{\mathrm{a}}\}, \{\ \}) \\
Q & = & \{q_0, q_1, \ldots, q_6, q_{\mathrm{a}}\} \\
\delta & = & \{\, [q_0, \triangleright, [\triangleright, \triangleright], q_1],\ [q_1, +, +, q_2],\ [q_2, a, [\#, a], q_1], \\
 & & [q_2, b, [\#, \#], q_3],\ [q_2, \triangleleft, [\#, \#], q_4],\ [q_3, +, 0, q_2], \\
 & & [q_4, -, -, q_5],\ [q_5, b, [a, b], q_4],\ [q_5, a, [a, a], q_6], \\
 & & [q_5, \triangleright, [\triangleright, \triangleright], q_a],\ [q_5, a, [\triangleright, \triangleright], q_6],\ [q_6, -, 0, q_5] \ \}
\end{array}
$$

$t = 0$



$t = 53$

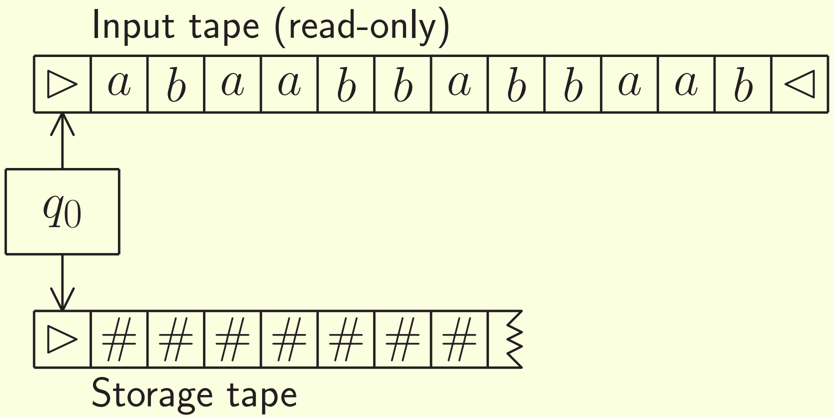


An RDTM $T^{\dagger}_{\mathsf{eq}}$ that simulates T_{eq}

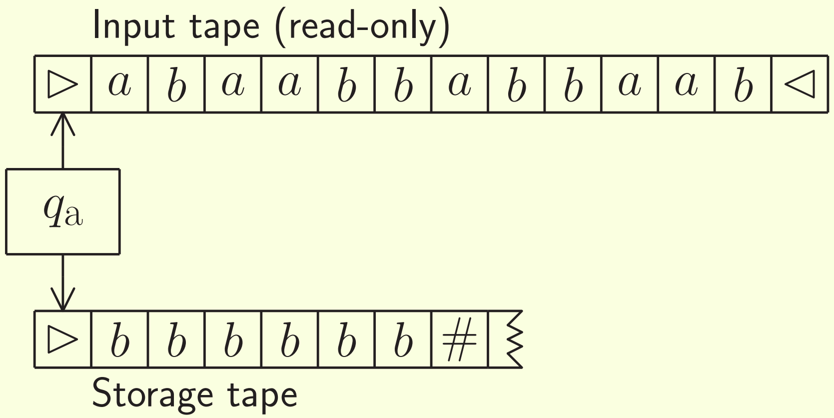
$$
\begin{array}{rcl}
T^{\dagger}_{\mathsf{eq}} & = & (Q^{\dagger}, \{a,b\}, \{a,b\}, \delta^{\dagger}, \rhd, \lhd, q_0, \#, \{\hat{q}^1_0\}, \{q^1_0\}) \\
Q^{\dagger} & = & \{q, \hat{q}, q^1, \hat{q}^1 \mid q \in Q\} \cup \{\ q^2_2, q^2_5\ \} \\
\delta^{\dagger} & = & \delta_1 \cup \cdots \cup \delta_6 \cup \hat{\delta}_1 \cup \cdots \cup \hat{\delta}_5 \cup \delta_{\mathsf{a}} \cup \delta_{\mathsf{r}} \\
\delta_1 & = & \{\, [q_1, +, +, q^2_2], [q_3, +, 0, q_2], [q_4, -, -, q^2_5], [q_6, -, 0, q_5]\ \} \\
\delta_2 & = & \{\, [q_0, \rhd, [\rhd, \rhd], q_1],\ \ [q_2, a, [\#, a], q_1], [q_2, b, [\#, \#], q_3], \\
 & & \ \ [q_2, \lhd, [\#, \#], q_4], [q_5, b, [a, b], q_4],\ \ [q_5, a, [a, a], q_6], \\
 & & \ \ [q_5, \rhd, [\rhd, \rhd], q_a],\ \ [q_5, a, [\rhd, \rhd], q_6]\ \ \} \\
\delta_3 & = & \{\, [q^1_2, -, -, q^1_1], [q^2_2, -, 0, q^1_3], [q^1_5, +, +, q^1_4], [q^2_5, +, 0, q^1_6]\ \} \\
\delta_4 & = & \{\, [q^1_1, \rhd, [\rhd, \rhd], q^1_0],\ \ [q^1_1, a, [a, \#], q^1_2],\ \ [q^1_3, b, [\#, \#], q^1_2], \\
 & & \ \ [q^1_4, \lhd, [\#, \#], q^1_2], [q^1_4, b, [b, a], q^1_5],\ \ [q^1_6, a, [a, a], q^1_5], \\
 & & \ \ [q^1_a, \rhd, [\rhd, \rhd], q^1_5],\ \ [q^1_6, a, [\rhd, \rhd], q^1_5]\ \} \\
\delta_5 & = & \{\, [q^1_1, \rhd, [\#, \#], q_1], [q^1_1, \rhd, [a, a], q_1],\ \ \ldots\ \ , [q^1_6, \lhd, [b, b], q_6]\ \} \\
\hat{\delta}_i & = & \{\, [\hat{p}, \mathbf{x}, \hat{q}] \mid [p, \mathbf{x}, q] \in \delta_i\,\}\ (i = 1, \ldots, 5) \\
\delta_6 & = & \{\, [q_2, \rhd, [\rhd, \rhd], q^1_2], [q_2, \rhd, [\#, \#], q^1_2],\ \ \ldots\ \ , [q_5, \lhd, [b, b], q^1_5]\ \} \\
\delta_{\mathsf{a}} & = & \{\, [q_a, 0, 0, \hat{q}^1_a]\ \} \\
\delta_{\mathsf{r}} & = & \{\ \}
\end{array}
$$

Simulating the DTM T_{eq} by the RDMFA $T_{\mathsf{eq}}^{\dagger}$

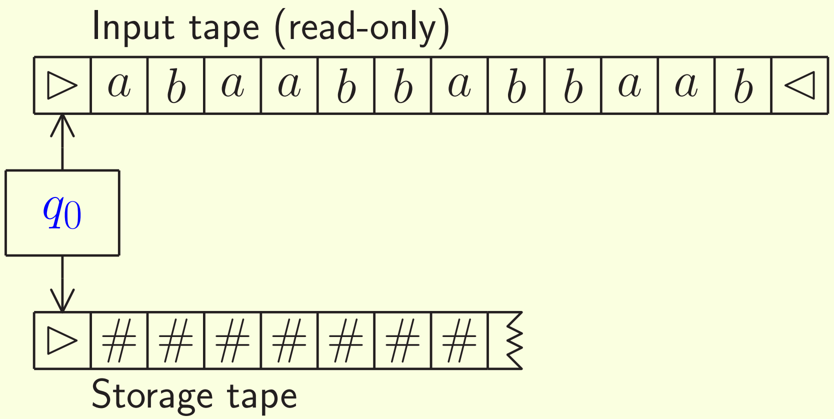
T_{eq}: $t = 0$



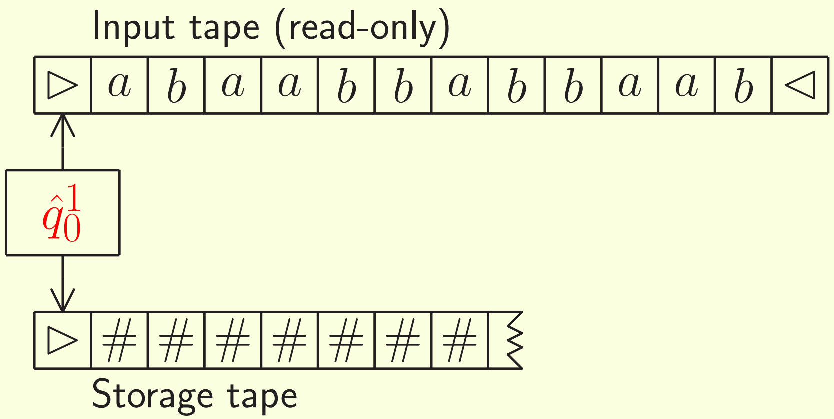
$t = 53$



$T_{\mathsf{eq}}^{\dagger}$: $t = 0$



$t = 185$



Concluding remarks

The following relations are proved.

- $\mathcal{L}[\mathsf{RDMFA}(k)] = \mathcal{L}[\mathsf{DMFA}(k)] \quad (k = 1, 2, \ldots)$.
- $\mathcal{L}[\mathsf{RDTM}(s(n))] = \mathcal{L}[\mathsf{DTM}(s(n))]$.

  The constructed RDTM is garbage-less, and uses the same number of storage tape symbols.

The proposed converting method can be used to many other memory-bounded computing models, e.g. a marker automaton, a space-bounded Turing transducer, etc.