Fundamentele Informatica 3

najaar 2016

`http://www.liacs.leidenuniv.nl/~vlietrvan1/fi3/`

Rudy van Vliet
kamer 124 Snellius, tel. 071-527 5777
rvvliet(at)liacs(dot)nl

college 3, 20 september 2016

7. Turing Machines

7.4. Combining Turing Machines

7.5. Multitape Turing Machines

7.4. Combining Turing Machines

Example.

A TM for $f(x) = a^{n_a(x)}$

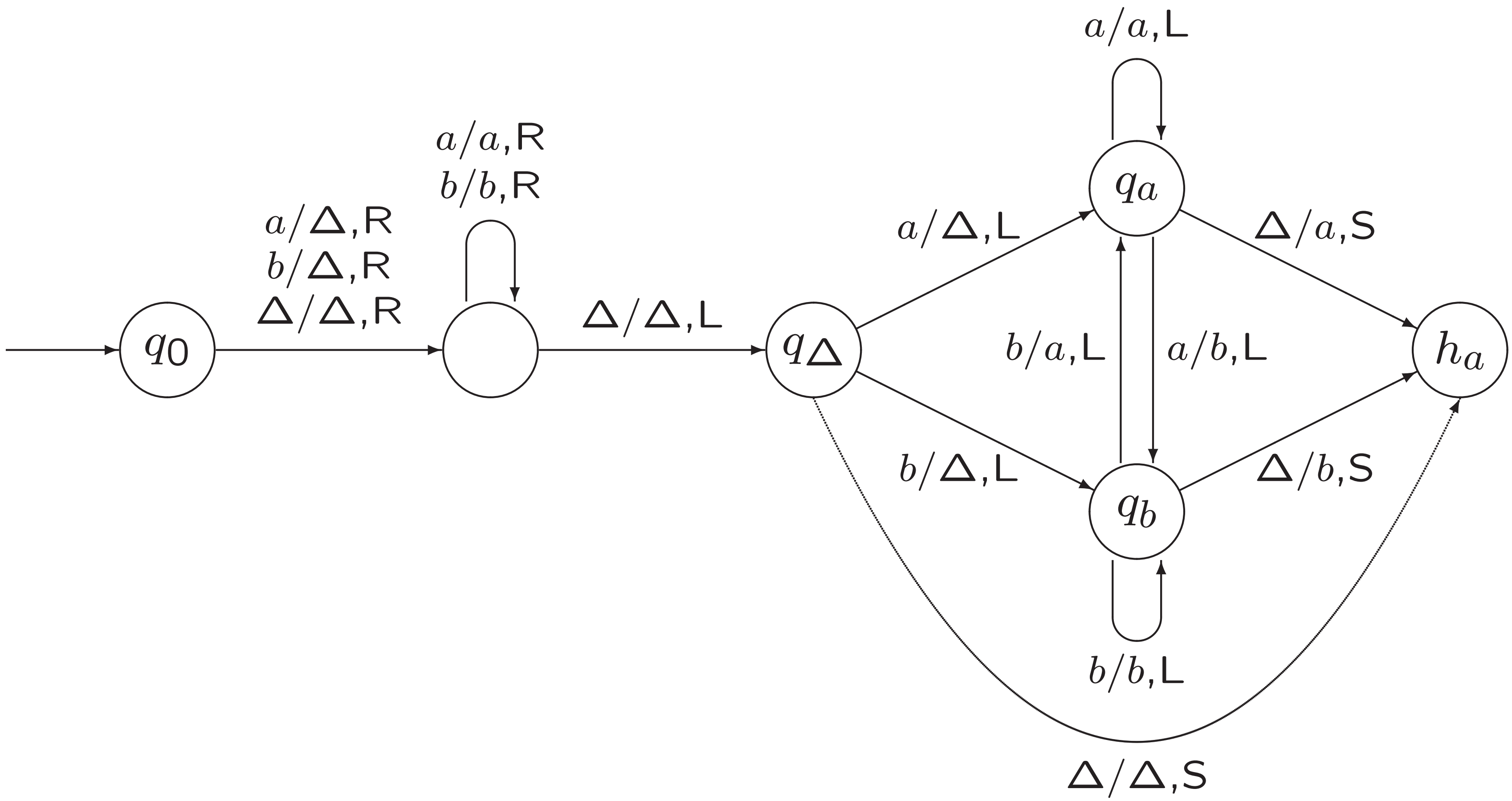
$x = aababba$

Example.

A TM for $f(x) = a^{n_a(x)}$

$x = aababba$

$$
\begin{array}{cccccccc}
\underline{\Delta} & a & a & b & a & b & b & a \\
\Delta & a & a & \underline{\Delta} & a & b & b & a \\
\Delta & a & a & \underline{a} & b & b & a & \Delta \\
\Delta & a & a & a & \underline{\Delta} & b & a & \Delta \\
\Delta & a & a & a & \underline{b} & a & \Delta & \Delta \\
\Delta & a & a & a & \underline{\Delta} & a & \Delta & \Delta \\
\Delta & a & a & a & \underline{a} & \Delta & \Delta & \Delta \\
\underline{\Delta} & a & a & a & a & \Delta & \Delta & \Delta
\end{array}
$$



a/a,L
a/a,R
b/b,R
a/Δ,R
b/Δ,R
Δ/Δ,R
Δ/Δ,L
q_0
q_Δ
q_a
q_b
h_a
a/Δ,L
b/Δ,L
b/a,L
a/b,L
Δ/a,S
Δ/b,S
b/b,L
Δ/Δ,S

Example 7.20. Inserting and Deleting a Symbol

Delete: from $y\underline{\sigma}z$ to $y\underline{z}$

Insert(σ): from $y\underline{z}$ to $y\underline{\sigma}z$

N.B.: z does not contain blanks

TM T_1 computes f

TM T_2 computes g

TM T_1T_2 computes . . .

$$T_1 \longrightarrow T_2$$

Example 7.17. Finding the Next Blank or the Previous Blank

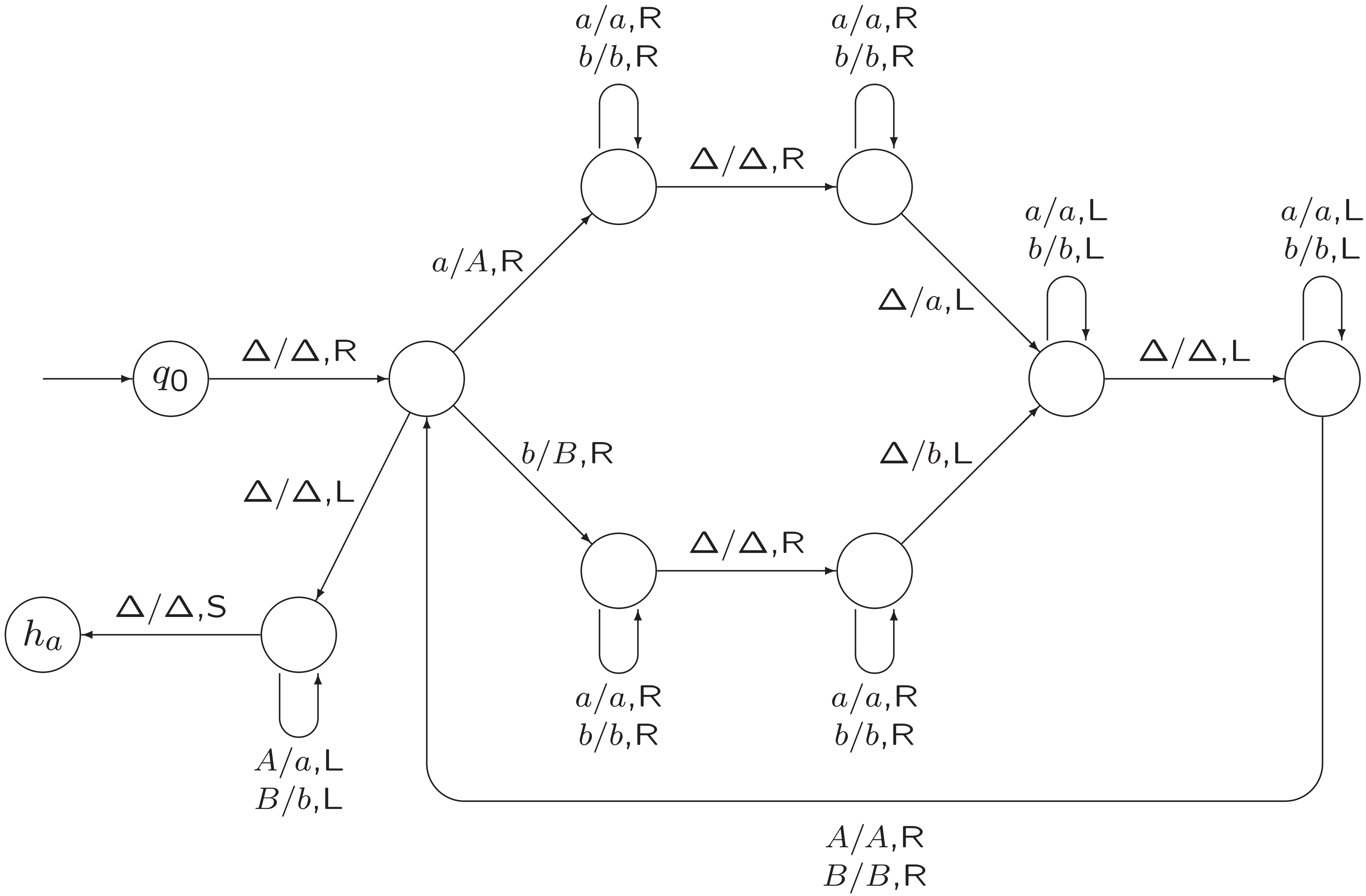
NB

PB

Example 7.18. Copying a String

Copy: from $\underline{\Delta} x$ to $\underline{\Delta} x \Delta x$

$x = abaa$


a/a,R
b/b,R
a/a,R
b/b,R
Δ/Δ,R
a/A,R
a/a,L
b/b,L
a/a,L
b/b,L
Δ/a,L
q_0
Δ/Δ,R
Δ/Δ,L
b/B,R
Δ/b,L
Δ/Δ,L
Δ/Δ,R
Δ/Δ,S
h_a
a/a,R
b/b,R
a/a,R
b/b,R
A/a,L
B/b,L
A/A,R
B/B,R

A slide from lecture 2

Example 7.10. The Reverse of a String

$$\underline{\Delta}\, a\ a\ b\ a\ b$$
$$\Delta A\, a\ b\ a\ b$$
$$\Delta A\, a\ b\ a\, A$$
$$\Delta B\, a\ b\ a\, A$$
$$\Delta BA\, b\ a\, A$$
$$\Delta BA\, b\, AA$$
$$\Delta BA\, b\, AA$$
$$\Delta BABAA$$
$$\underline{\Delta}\, b\ a\ b\ a\ a$$

Example 7.24. Comparing Two Strings

Equal: accept $\underline{\underline{\Delta}} x \Delta y$ if $x = y$,
and reject if $x \neq y$

Exercise 7.17.

For each case below, draw a TM that computes the indicated function.

e. $E : \{a, b\}^* \times \{a, b\}^* \to \{0, 1\}$
defined by $E(x, y) = 1$ if $x = y$, $E(x, y) = 0$ otherwise.

Example 7.25. Accepting the Language of . . .

$$\mathit{Copy} \rightarrow \mathit{NB} \rightarrow R \rightarrow \mathit{PB} \rightarrow \mathit{Equal}$$

Example 7.25. Accepting the Language of Palindromes

$$Copy \to NB \to R \to PB \to Equal$$

Example 7.21. Erasing the Tape

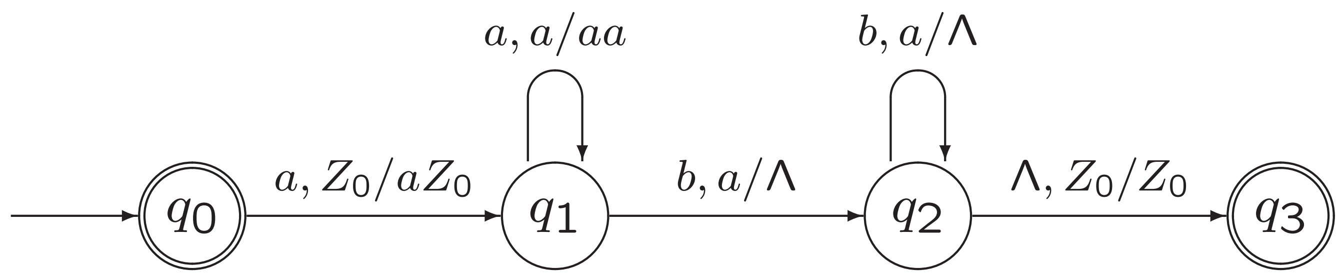
From the current position to the right

Many notations for composition

7.5. Multitape Turing Machines

Example 5.3. A PDA Accepting the Language *AnBn*

$$AnBn = \{a^i b^i \mid i \geq 0\}$$



Exercise 7.4.

For each of the following languages, draw a transition diagram for a Turing machine that accepts that language.

a. *AnBn* $= \{a^i b^i \mid i \geq 0\}$

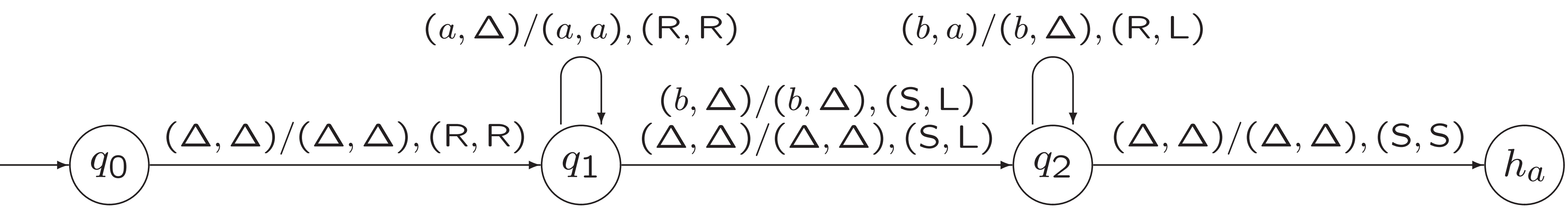
With two tapes. . .

Exercise 7.4.

For each of the following languages, draw a transition diagram for a Turing machine that accepts that language.

a. $AnBn = \{a^i b^i \mid i \geq 0\}$

With two tapes. . .



Exercise 7.4.

For each of the following languages, draw a transition diagram for a Turing machine that accepts that language.

a. *AnBn* $= \{a^i b^i \mid i \geq 0\}$

We could also use the portion of the tape to the right of the input, to simulate the stack of a deterministic pushdown automaton (works for any deterministic PDA!)

Example 7.24. Comparing Two Strings

Equal: accept $\underline{\Delta} x \Delta y$ if $x = y$,
and reject if $x \neq y$

2-tape TM. . .

A slide from lecture 2

Definition 7.1. Turing machines

A Turing machine (TM) is a 5-tuple $T = (Q, \Sigma, \Gamma, q_0, \delta)$, where

Q is a finite set of states. The two *halt* states h_a and h_r are not elements of Q.

Σ, the input alphabet, and Γ, the tape alphabet, are both finite sets, with $\Sigma \subseteq \Gamma$. The *blank* symbol Δ is not an element of Γ.

q_0, the initial state, is an element of Q.

δ is the transition function:

$$\delta : Q \times (\Gamma \cup \{\Delta\}) \to (Q \cup \{h_a, h_r\}) \times (\Gamma \cup \{\Delta\}) \times \{R, L, S\}$$

2-Tape TM $T = (Q, \Sigma, \Gamma, q_0, \delta)$, where

$$\delta : Q \times (\Gamma \cup \{\Delta\})^2 \rightarrow (Q \cup \{h_a, h_r\}) \times (\Gamma \cup \{\Delta\})^2 \times \{R, L, S\}^2$$

Combination of two slides from lecture 2

Notation:

description of tape contents: $x\underline{\sigma}y$ or $x\underline{y}$

configuration $xqy = xqy\Delta = xqy\Delta\Delta$

initial configuration corresponding to input x: $q_0\Delta x$

In the third edition of the book, a configuration is denoted as $(q, x\underline{y})$ or $(q, x\underline{\sigma}y)$ instead of xqy or $xq\sigma y$.
In one case, we still use this old notation.

Configuration of 2-tape TM is

$$(q, x_1\underline{a_1}y_1, x_2\underline{a_2}y_2)$$

Initial configuration corresponding to input string x is

$$(q_0, \underline{\Delta}x, \underline{\Delta})$$

Output will appear on first tape.

Theorem 7.26.
For every 2-tape TM $T = (Q, \Sigma, \Gamma, q_0, \delta)$, there is an ordinary 1-tape TM $T_1 = (Q_1, \Sigma, \Gamma_1, q_1, \delta_1)$ with $\Gamma \subseteq \Gamma_1$, such that

1. For every $x \in \Sigma^*$, T accepts x if and only if T_1 accepts x, and T rejects x if and only if T_1 rejects x.
   (In particular, $L(T) = L(T_1)$.)

2. For every $x \in \Sigma^*$, if

$$(q_0, \underline{\Delta}x, \underline{\Delta}) \vdash_T^* (h_a, y\underline{a}z, u\underline{b}v)$$

for some strings $y, z, u, v \in (\Gamma \cup \{\Delta\})^*$ and symbols $a, b \in \Gamma \cup \{\Delta\}$, then

$$q_1 \Delta x \vdash_{T_1}^* y h_a a z \qquad \text{i.e., } (q_1, \underline{\Delta}x) \vdash_{T_1}^* (h_a, y\underline{a}z)$$

Proof. . .

Simulating two tapes on one

Δ	$\underline{1}$	Δ	0	1	Δ	$\cdots$
0	1	0	$\underline{0}$	Δ	Δ	$\cdots$

If $\delta(p, 1, 0) = (q, \Delta, 1, L, R)\ldots$

Simulating two tape heads

Simulating move of 2-tape TM T by 1-tape TM T_1

1. Move left to \$, right to σ', back to \$

Simulating move of 2-tape TM T by 1-tape TM T_1

1. Move left to \$, right to σ', back to \$

2. Move right to τ'
Let $\delta(p, \sigma, \tau) = (q, \sigma_1, \tau_1, D_1, D_2)$
If $q = h_r$, reject
Otherwise, $\tau' \to \tau_1$ and move D_2

Simulating move of 2-tape TM T by 1-tape TM T_1

1. Move left to \$, right to σ', back to \$

2. Move right to τ'
Let $\delta(p, \sigma, \tau) = (q, \sigma_1, \tau_1, D_1, D_2)$
If $q = h_r$, reject
Otherwise, $\tau' \to \tau_1$ and move D_2

3. If \$, reject
Otherwise, (if #, move #) place ' and back to \$

Simulating move of 2-tape TM T by 1-tape TM T_1

1. Move left to \$, right to σ', back to \$

2. Move right to τ'
Let $\delta(p, \sigma, \tau) = (q, \sigma_1, \tau_1, D_1, D_2)$
If $q = h_r$, reject
Otherwise, $\tau' \to \tau_1$ and move D_2

3. If \$, reject
Otherwise, (if #, move #) place ' and back to \$

4. Move right to σ', $\sigma' \to \sigma_1$ and move D_1

Simulating move of 2-tape TM T by 1-tape TM T_1

1. Move left to \$, right to σ', back to \$

2. Move right to τ'
Let $\delta(p, \sigma, \tau) = (q, \sigma_1, \tau_1, D_1, D_2)$
If $q = h_r$, reject
Otherwise, $\tau' \to \tau_1$ and move D_2

3. If \$, reject
Otherwise, (if #, move #) place ' and back to \$

4. Move right to σ', $\sigma' \to \sigma_1$ and move D_1

5. If \$, reject
Otherwise, (if #, move #) place '

If T accepts, then. . .

6. Delete second track

If T accepts, then...

6. Delete second track

7. Delete \$ and \#

If T accepts, then...

6. Delete second track

7. Delete \$ and $\#$

8. Find σ', unprime, halt in h_a

Corollary 7.27.
Every language that is accepted by a 2-tape TM can be accepted by an ordinary 1-tape TM,
and every function that is computed by a 2-tape TM can be computed by an ordinary TM.

This generalizes to k-tape TMs for $k \geq 3$.