Computability

voorjaar 2025

https://liacs.leidenuniv.nl/~vlietrvan1/computability/

college 3, 17 februari 2025

7. Turing Machines
7.3. Turing Machines That Compute Partial Functions
7.4. Combining Turing Machines
7.5. Multitape Turing Machines
7.7. Nondeterministic Turing Machines

Huiswerkopgave

voor 0.4pt

morgen online

individueel

inleveren: dinsdag 4 maart 2025, 23.59 uur

A slide from lecture 2

Example 7.12. The Quotient and Remainder Mod 2

Example.

Dividing by 3

Example.

Accepting $\{xyx \ \mid \ x, y \in \{a, b\}^*$ and $|x| = |y|\}$

Exercise.

Draw a TM that computes the function $f(x, y) = x \bmod y$

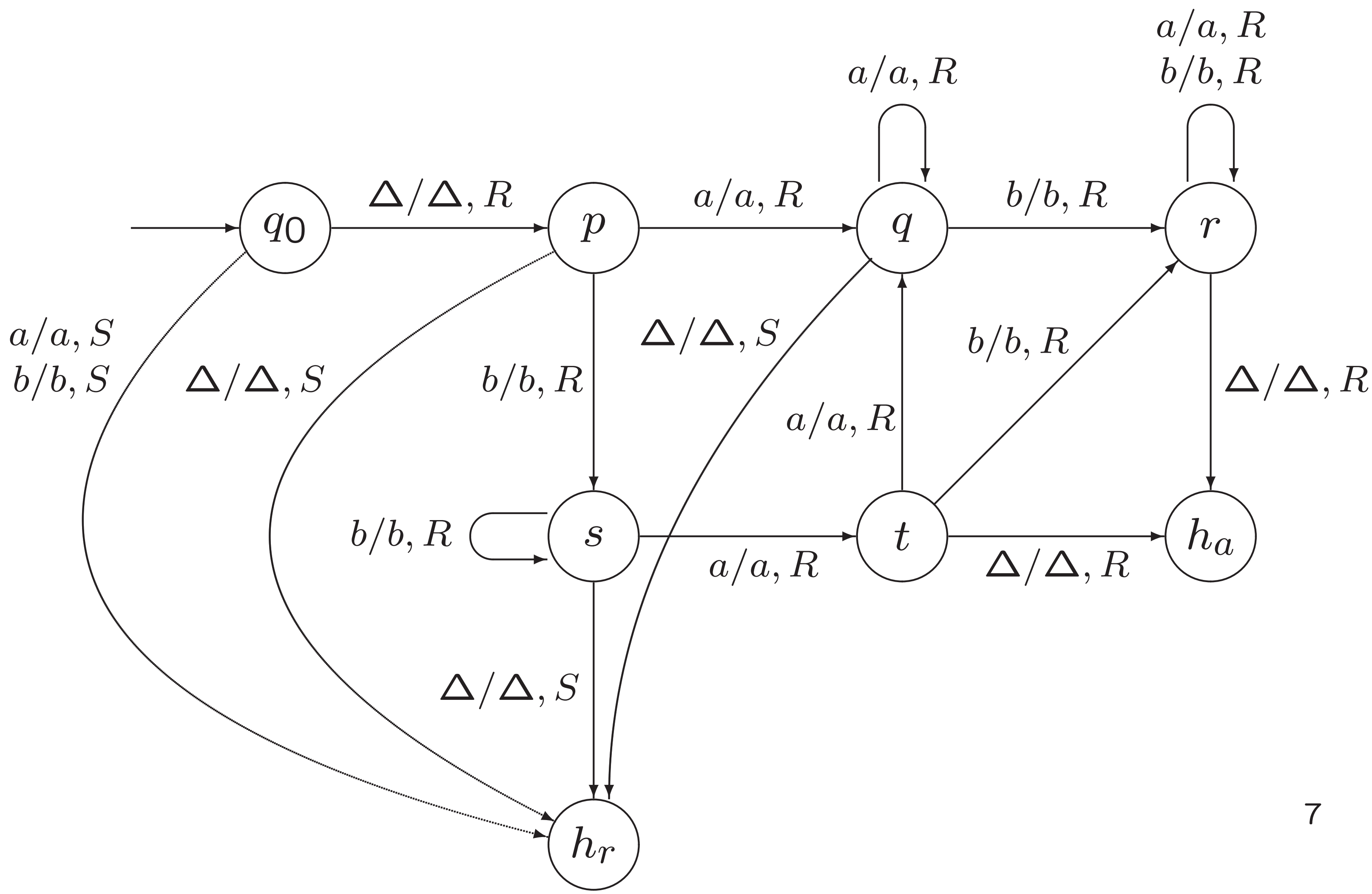
Hint: implement the following algorithm:

```
while (x >= y)
  x = x - y;
```

Make this exercise yourself.

Example 7.3. A TM Accepting a Regular Language

$L = \{a, b\}^*\{ab\}\{a, b\}^* \cup \{a, b\}^*\{ba\}$



Example 7.14. The Characteristic Function of a Set

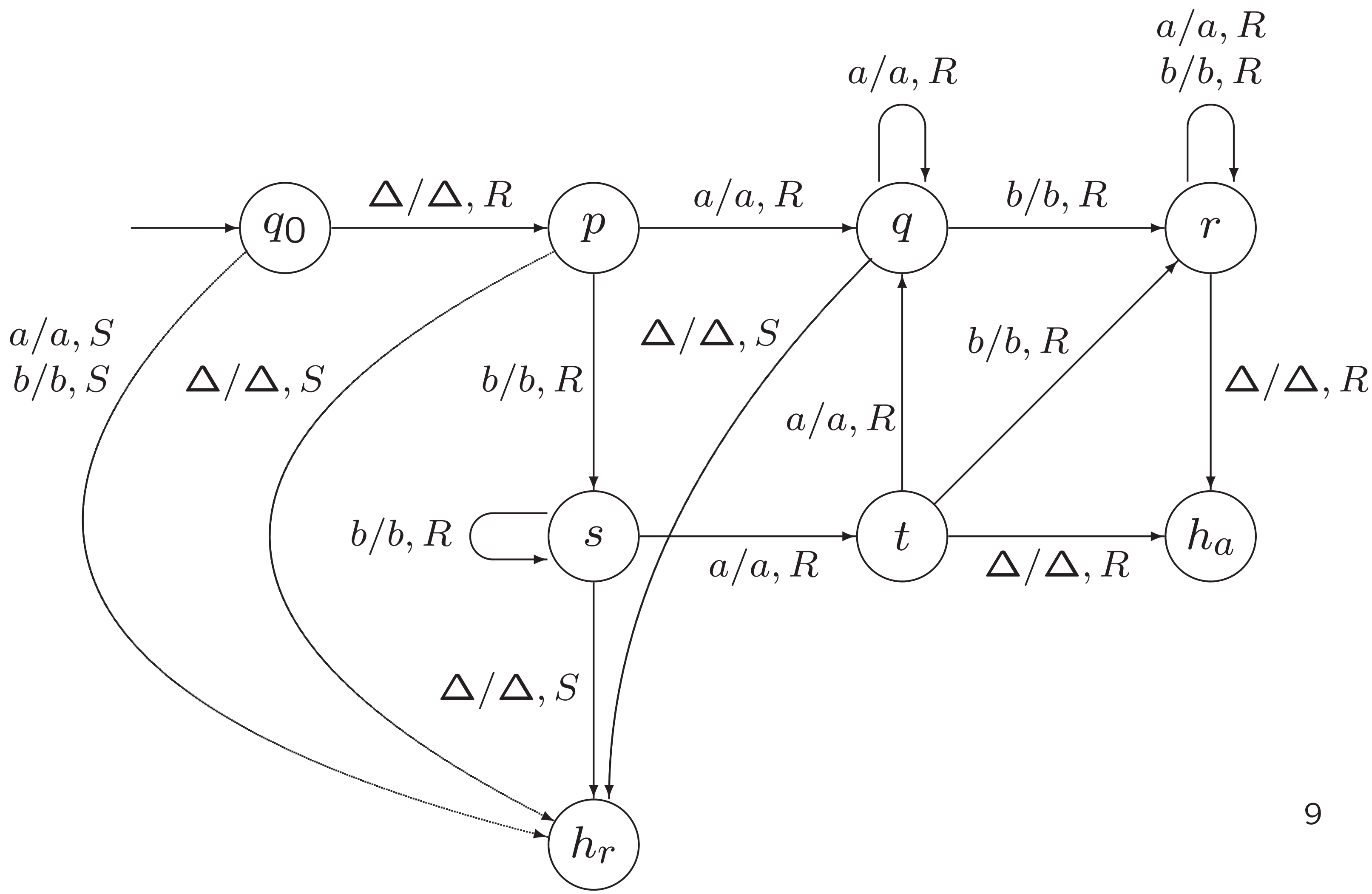
$L = \{a,b\}^*\{ab\}\{a,b\}^* \cup \{a,b\}^*\{ba\}$

$\chi_L : \{a,b\}^* \to \{0,1\}$, defined by

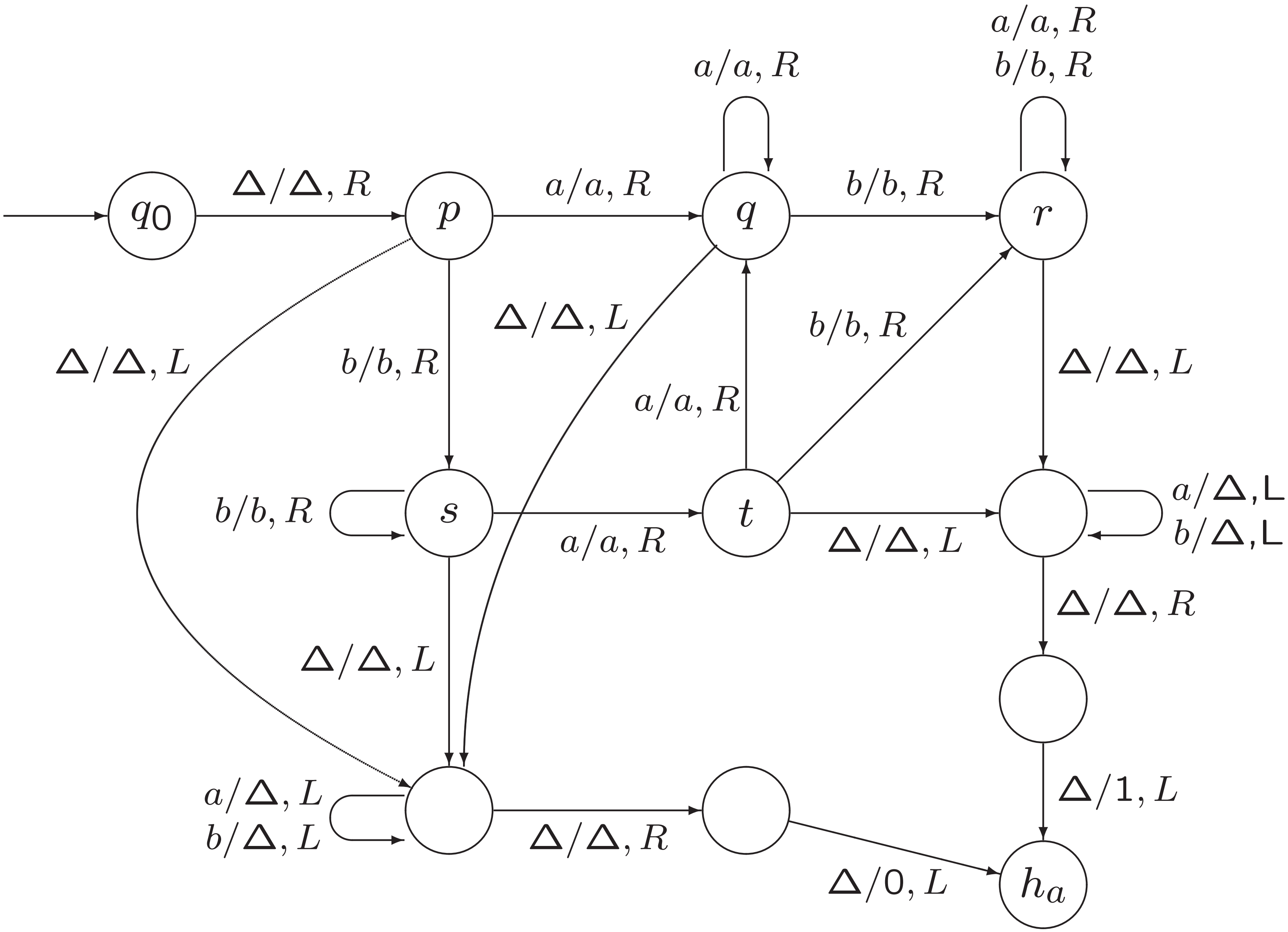
$$\chi_L(x) = \begin{cases} 1 & \text{if } x \in L \\ 0 & \text{if } x \notin L \end{cases}$$

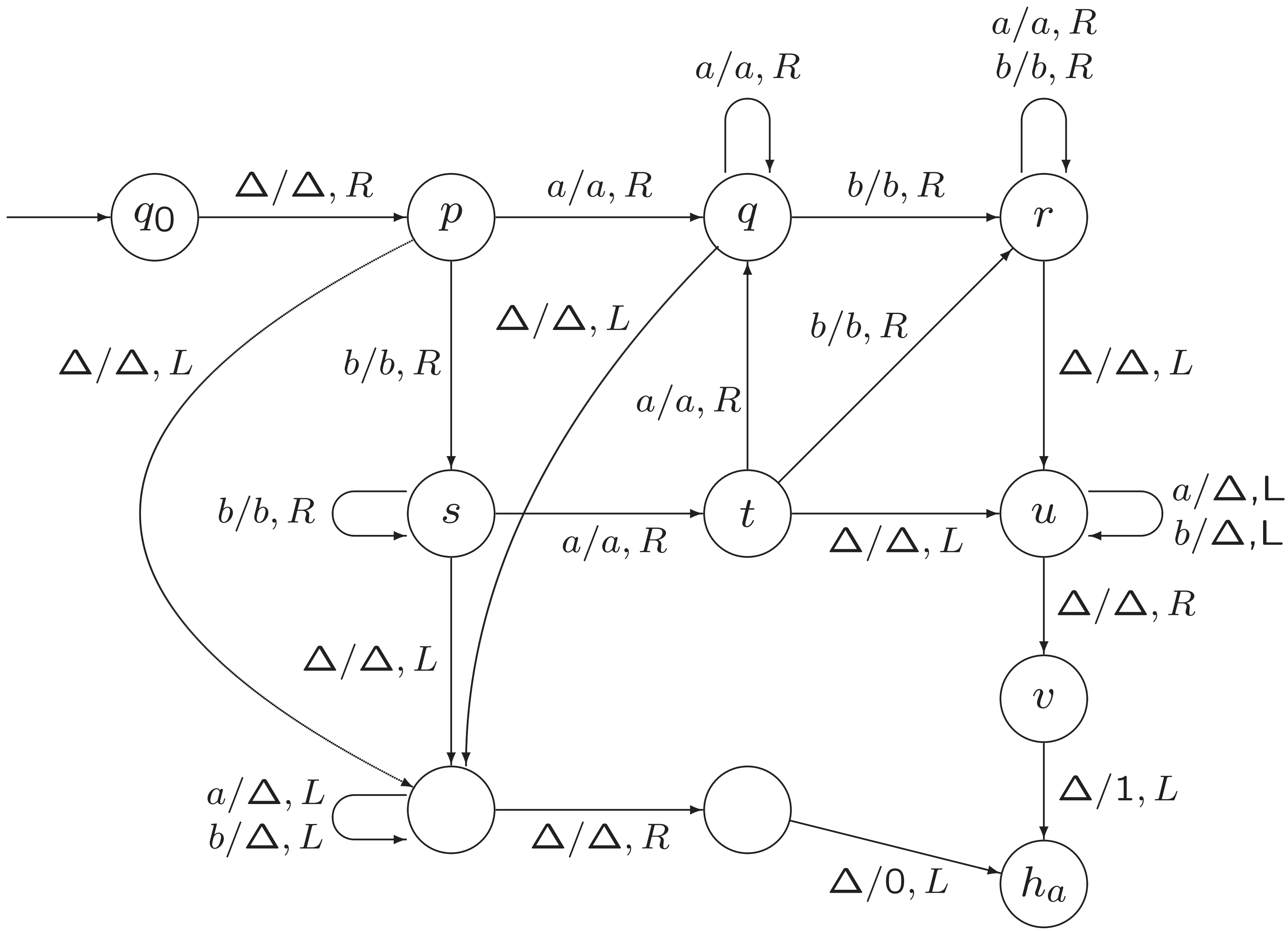
Example 7.3. A TM Accepting a Regular Language

$L = \{a, b\}^*\{ab\}\{a, b\}^* \cup \{a, b\}^*\{ba\}$



Example 7.14. The Characteristic Function of a Set
$L = \{a,b\}^*\{ab\}\{a,b\}^* \cup \{a,b\}^*\{ba\}$





$$q_0\Delta aaba \vdash \Delta paaba \vdash \Delta aqaba \vdash \Delta aaqba \vdash \Delta aabra \vdash \Delta aabar\Delta \vdash$$
$$\Delta aabua \vdash \Delta aaub \vdash \Delta aua \vdash \Delta ua \vdash u\Delta \vdash \Delta v\Delta \vdash h_a\Delta 1$$

Example 7.14. The Characteristic Function of a Set

$$\chi_L(x) = \begin{cases} 1 & \text{if } x \in L \\ 0 & \text{if } x \notin L \end{cases}$$

From computing χ_L to accepting L

From accepting L to computing χ_L

Een Intermezzo

http://www.youtube.com/watch?v=E3keLeMwfHY

7.4. Combining Turing Machines

Example.

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

$x = aababba$

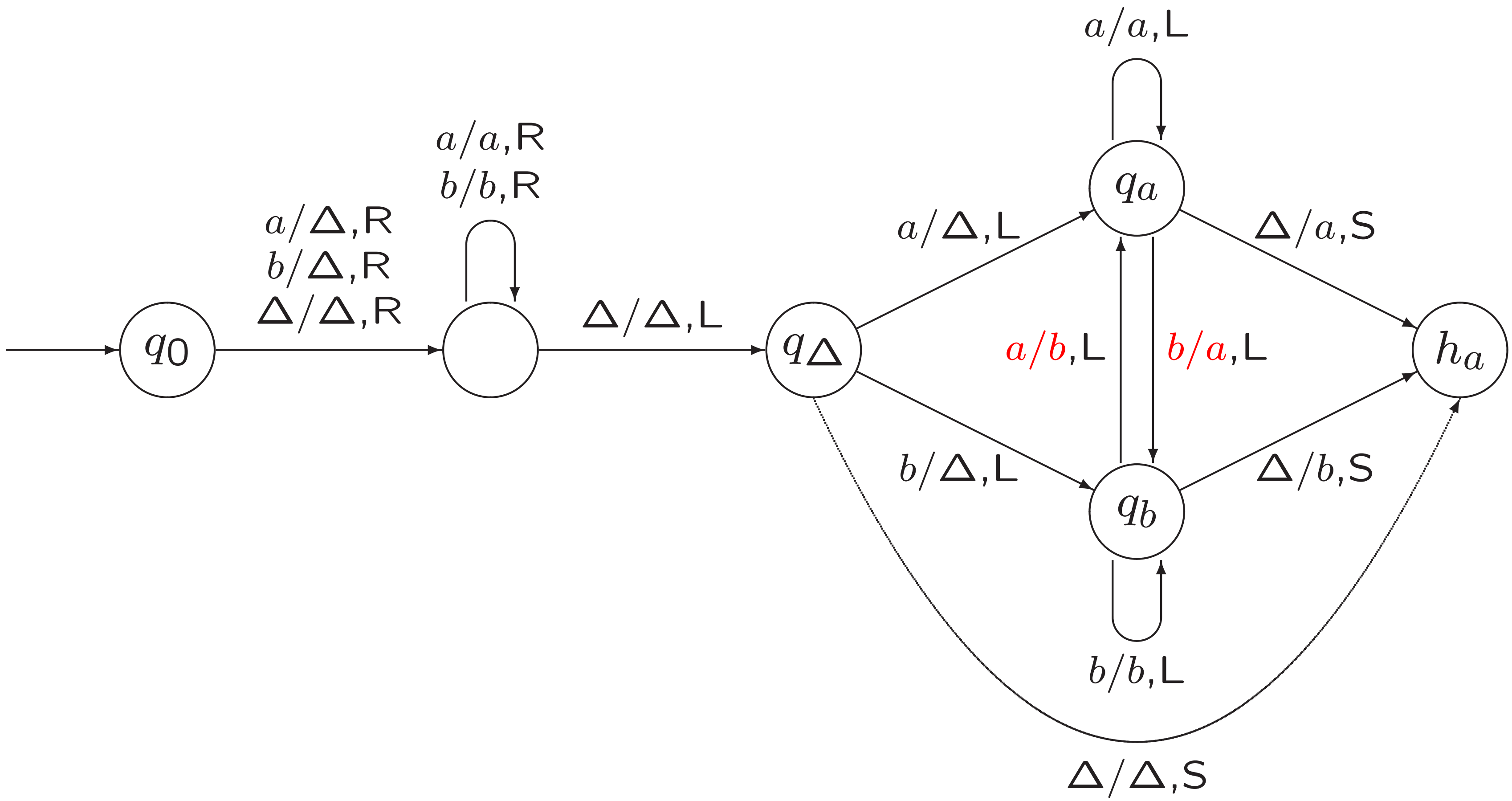
Example.

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

$x = aababba$

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

TM. . .



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

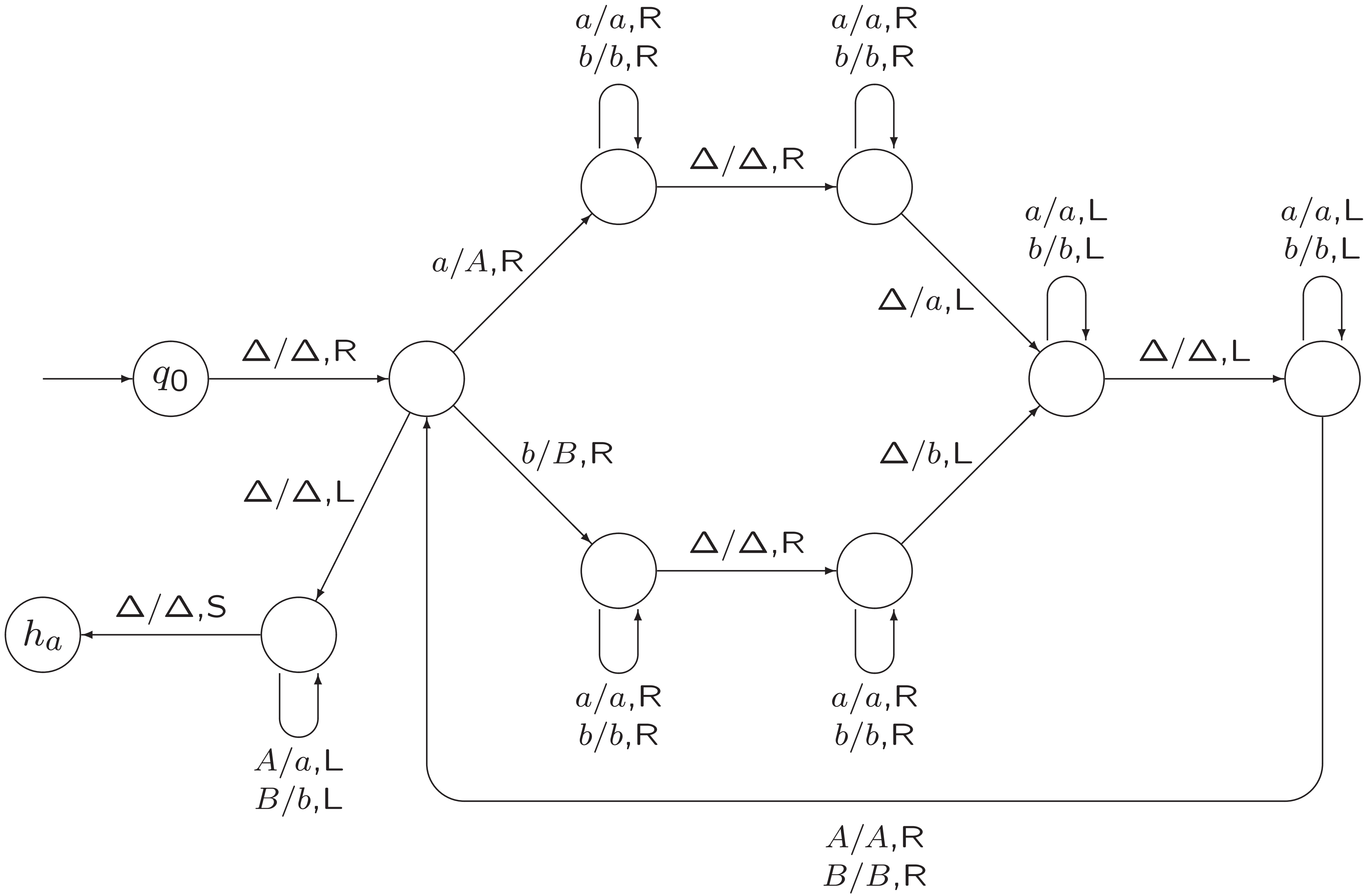
NB

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,L
b/b,L
a/a,L
b/b,L
a/A,R
Δ/a,L
q0
Δ/Δ,R
Δ/Δ,L
b/B,R
Δ/b,L
Δ/Δ,L
Δ/Δ,R
Δ/Δ,S
ha
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 $\underline{\Delta} x \Delta y$ if $x = y$,
and reject if $x \neq y$

An exercise from exercise class 2

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 $x = y$, $\quad E(x, y) = 0$ otherwise.

Example 7.25. Accepting the Language of . . .

$$Copy \rightarrow NB \rightarrow R \rightarrow PB \rightarrow Equal$$

Example 7.25. Accepting the Language of Palindromes

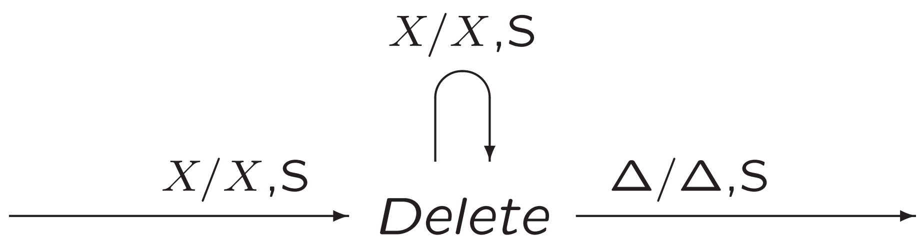
$$Copy \to NB \to R \to PB \to Equal$$

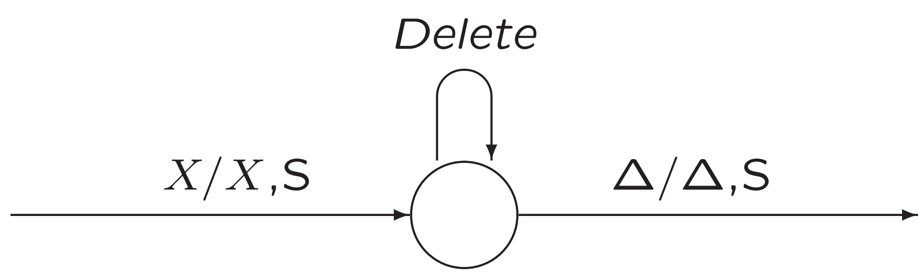
Example 7.21. Erasing the Tape

From the current position to the right

Many notations for composition

Incorrect notations for DeleteString:



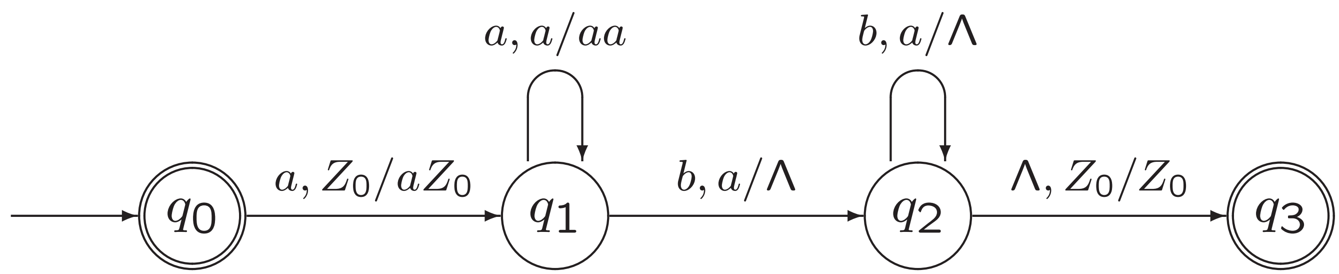


Not preferred notation:

7.5. Multitape Turing Machines

Example 5.3. A PDA Accepting the Language *AnBn*

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



Part of a slide from exercise class 1

Exercise 7.4.

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

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

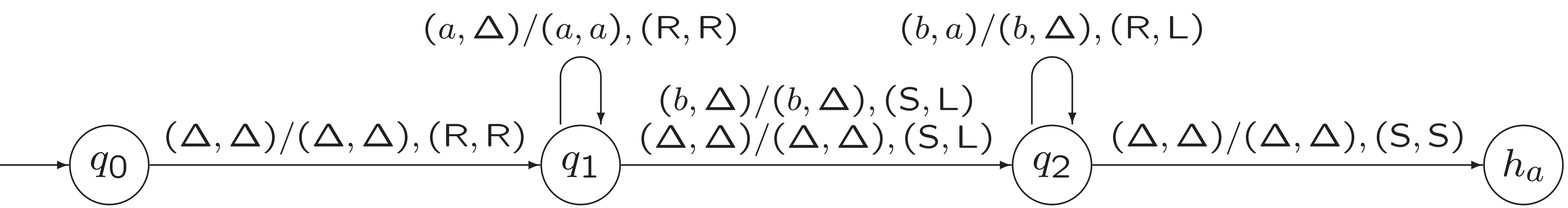
With two tapes. . .

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:



A slide from exercise class 1

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. . .

Multi-tape TM

- initial configuration. . .
- to compute a function. . .

Theorem 7.26. (informal)
For every 2-tape TM T, there is an ordinary 1-tape TM T_1, which for every input x,

- simulates the computation of T for x,
- accepts (rejects) x, if and only if T accepts (rejects) x,
- on acceptance, leaves the same output on its tape as T leaves on its first tape.

Proof idea, for x = aabb...

The proof of this result does not have to be known for the exam.

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 $k \geq 3$.

7.7. Nondeterministic Turing Machines

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:

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

Nondeterministic Turing machine.

There may be more than one move for a state-symbol pair.

Same notation:

$$wpax \vdash_T yqbz \qquad wpax \vdash_T^* yqbz$$

A string x is accepted by T if

$$q_0 \Delta x \vdash_T^* w h_a y$$

for some strings $w, y \in (\Gamma \cup \{\Delta\})^*$.

NTM useful for accepting languages, for producing output, but not for computing function.

Example 7.28. The Set of Composite Natural Numbers.

Use *G2*

Example 7.28. The Set of Composite Natural Numbers.

$$\mathit{NB} \rightarrow \mathit{G2} \rightarrow \mathit{NB} \rightarrow \mathit{G2} \rightarrow \mathit{PB} \rightarrow \mathit{M} \rightarrow \mathit{PB} \rightarrow \mathit{Equal}$$

Take $x = 1^{15}$