

Fundamentele Informatica 3

voorjaar 2020

<http://www.liacs.leidenuniv.nl/~vlietrvan1/fi3/>

Rudy van Vliet

kamer 140 Snellius, tel. 071-527 2876
rvvliet(at)liacs(dot)nl

college 3, 19 februari 2020

7. Turing Machines

7.4. Combining Turing Machines

7.5. Multitape Turing Machines

7.7. Nondeterministic Turing Machines

7.4. Combining Turing Machines

Example.

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

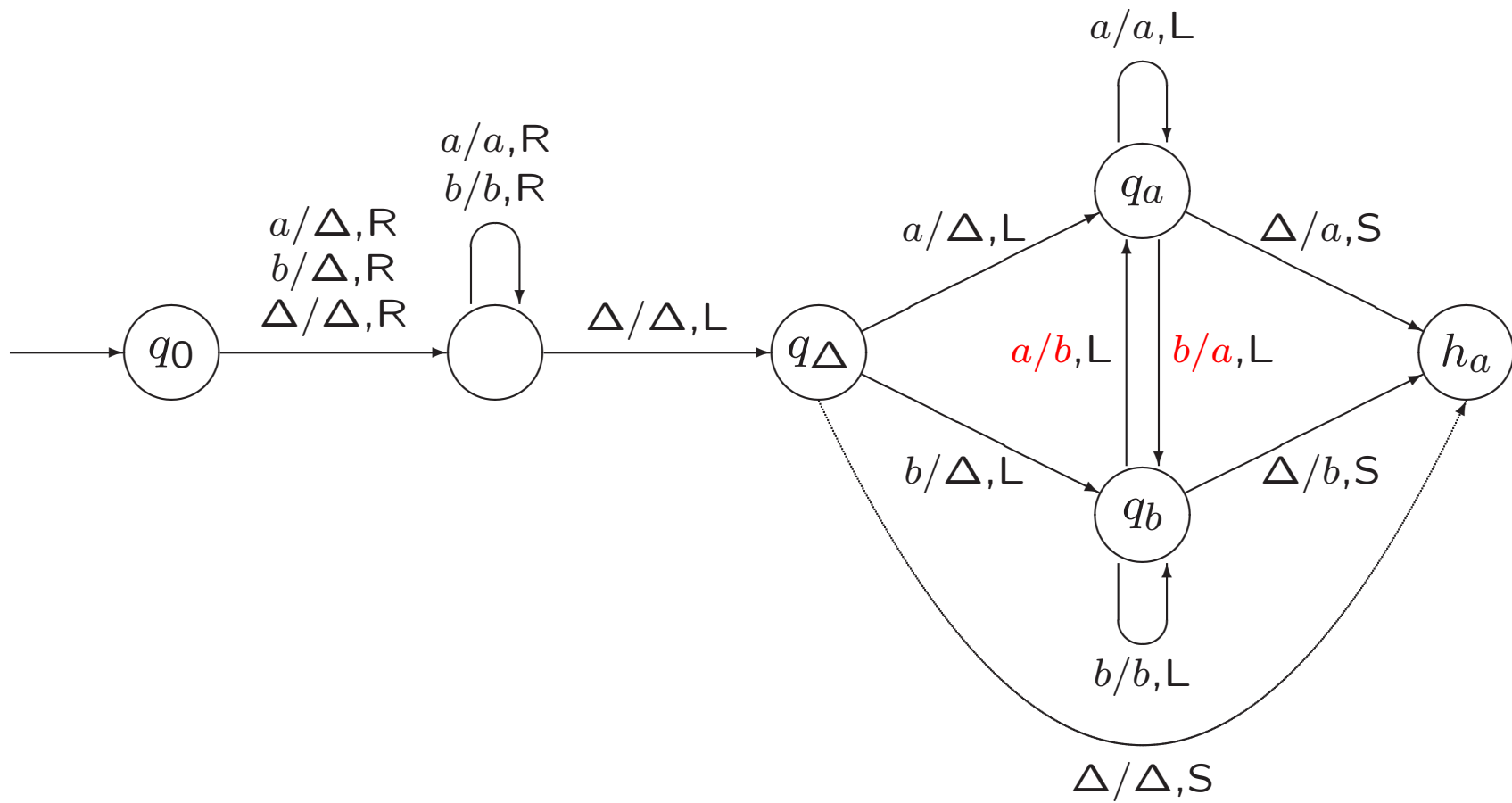
$x = aababba$

Example.

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

$x = aababba$

<u>△</u>	a	a	b	a	b	b	a
△	a	a	<u>△</u>	a	b	b	a
△	a	a	<u>a</u>	b	b	a	△
△	a	a	a	<u>△</u>	b	a	△
△	a	a	a	<u>b</u>	a	△△	
△	a	a	a	<u>△</u>	a	△△△	
△	a	a	a	<u>a</u>	△△△△		
<u>△</u>	a	a	a	a	△△△△		



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 ...



Example 7.17. Finding the Next Blank or the Previous Blank

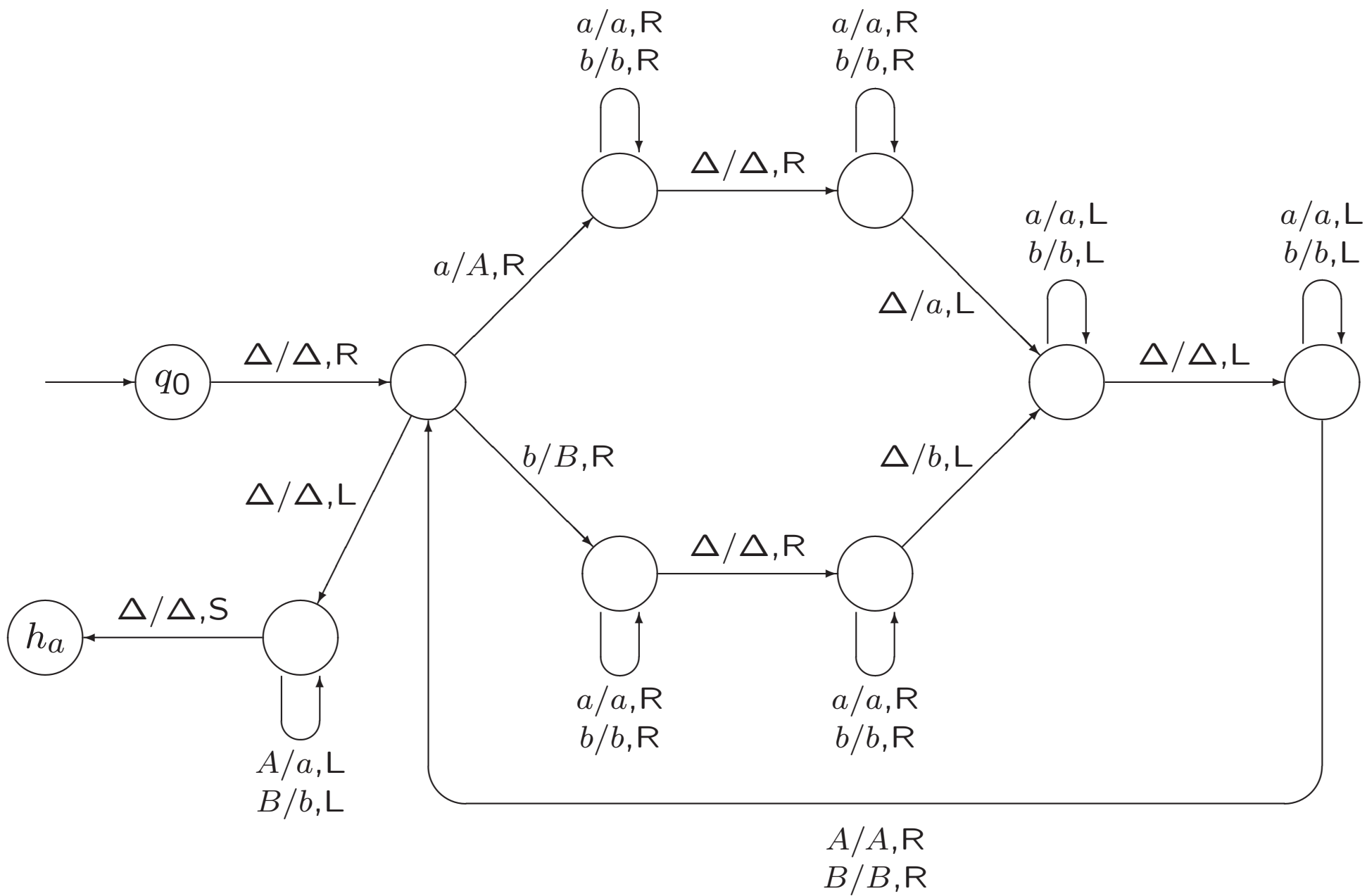
NB

PB

Example 7.18. Copying a String

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

$x = abaa$



A slide from lecture 2

Example 7.10. The Reverse of a String

Δ a a b a b
Δ A a b a b
Δ A a b a A
Δ B a b a A
Δ B A b a A
Δ B A b A A
Δ B A b A A
Δ B A B A A
Δ 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. $E : \{a, b\}^* \times \{a, b\}^* \rightarrow \{0, 1\}$
defined by $E(x, y) = 1$ if $x = y$, $E(x, y) = 0$ 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

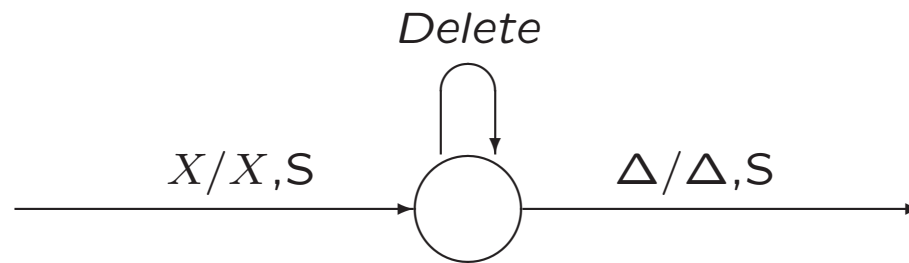
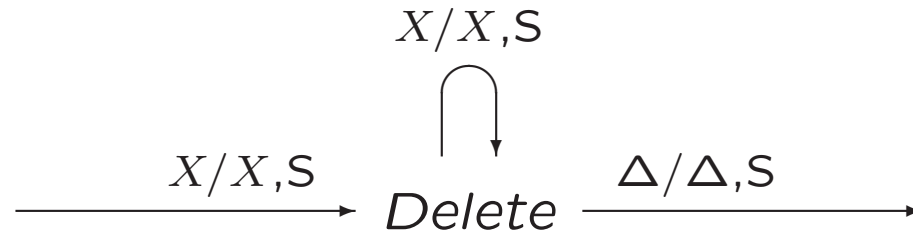
Copy \rightarrow *NB* \rightarrow *R* \rightarrow *PB* \rightarrow *Equal*

Example 7.21. Erasing the Tape

From the current position to the right

Many notations for composition

Incorrect notations for DeleteString:



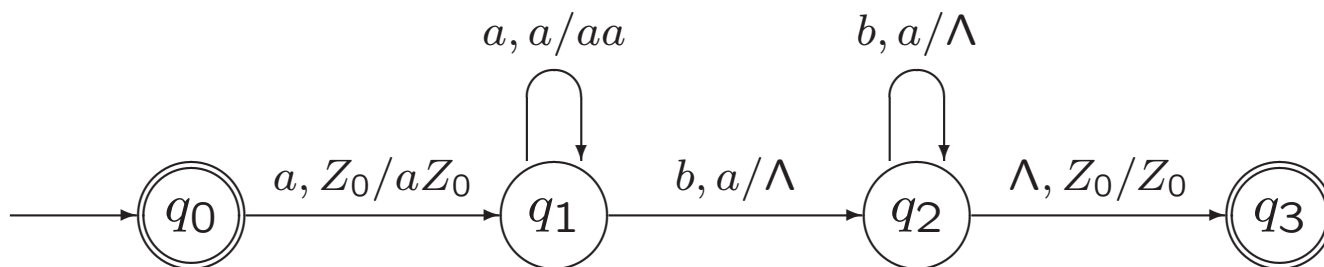
Not preferred notation:



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

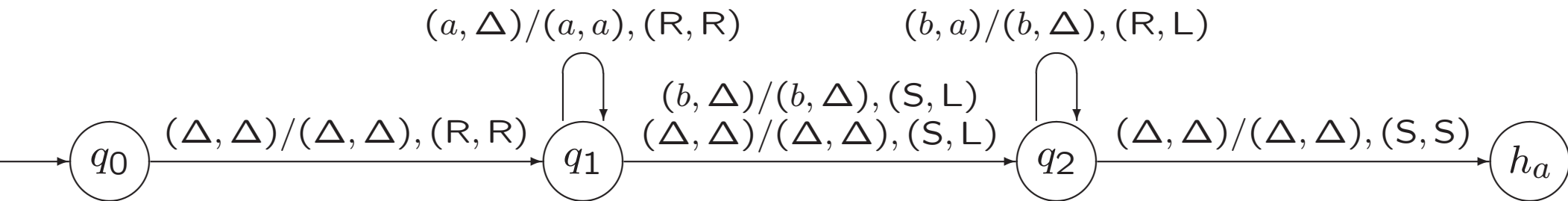
With two tapes...

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 $x = aabb\dots$

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**:

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

Nondeterministic Turing machine.

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

Same notation:

$$wpax \vdash_T yqbz \quad wpax \vdash_T^* yqbz$$

A string x is accepted by T if

$$q_0 \Delta x \vdash_T^* wh_a y$$

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.

$$NB \rightarrow G2 \rightarrow NB \rightarrow G2 \rightarrow PB \rightarrow M \rightarrow PB \rightarrow Equal$$

Take $x = 1^{15}$

Example 7.30. The Language of Prefixes of Elements of L .

Let $L = L(T)$. Then

$$P(L) = \{x \in \Sigma^* \mid xy \in L \text{ for some } y \in \Sigma^*\}$$

Example 7.30. The Language of Prefixes of Elements of L .

Let $L = L(T)$. Then

$$P(L) = \{x \in \Sigma^* \mid xy \in L \text{ for some } y \in \Sigma^*\}$$

Deterministic TM accepting $P(L)$ may execute following algorithm for input x :

$y = \Lambda$;

while (T does not accept xy)

y is next string in Σ^* (in canonical order);

accept;

but...

Example 7.30. The Language of Prefixes of Elements of L .

Let $L = L(T)$. Then

$$P(L) = \{x \in \Sigma^* \mid xy \in L \text{ for some } y \in \Sigma^*\}$$

$NB \rightarrow G \rightarrow Delete \rightarrow PB \rightarrow T$