Huiswerk

Nakijken huiswerkopgave 3. . .

Huiswerkopgave 4. . .

A slide from lecture 11

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:

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

A slide from lecture 11

Example 7.7. Accepting $L = \{a^i b a^j \mid 0 \leq i < j\}$

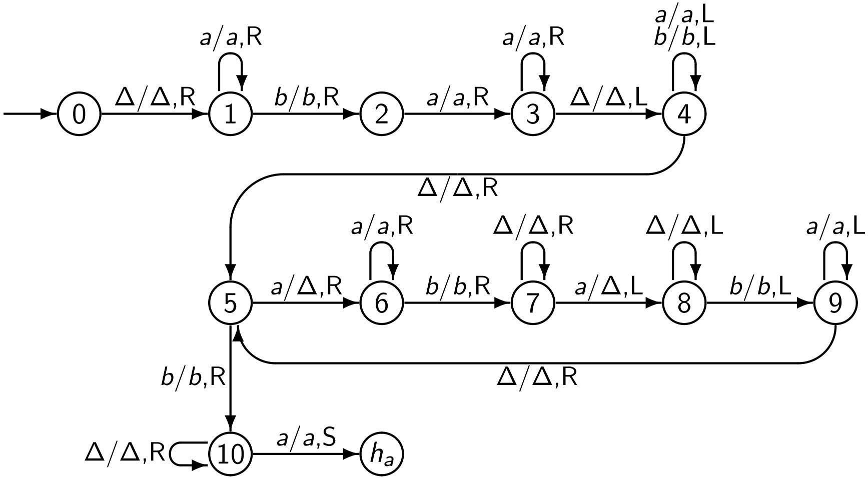
A slide from lecture 11

Example 7.7. Accepting $L = \{a^i b a^j \mid 0 \leq i < j\}$

$$\begin{array}{l}
\underline{\Delta}\, a\ a\ b\ a\ a\ \ a\ \ \ a \\
\Delta\Delta\, a\ b\ a\ a\ \ a\ \ \ a \\
\Delta\Delta\, a\ b\Delta\, a\ \ a\ \ \ a \\
\Delta\Delta\Delta b\Delta\, a\ \ a\ \ \ a \\
\Delta\Delta\Delta b\Delta\Delta\ \ a\ \ \ a
\end{array}$$

A slide from lecture 11

Example 7.7. Accepting $L = \{a^i b a^j \mid 0 \leq i < j\}$



What if $x \notin L$?

A slide from lecture 11

Example 7.7. Accepting $L = \{a^i ba^j \mid 0 \leq i < j\}$

To illustrate that a Turing machine T may run forever for an input that is not in $L(T)$. No problem!

No problem?

Notation:
configuration. . .

Notation:
description of tape contents: $x\underline{\sigma}y$ or $x\underline{y}$
$x\underline{y} = x\underline{y}\Delta = x\underline{y}\Delta\Delta$
if $y = \Lambda$, then $x\underline{\Delta}$

Notation:

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

$x\underline{y} = x\underline{y}\Delta = x\underline{y}\Delta\Delta$

if $y = \Lambda$, then $x\underline{\Delta}$

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

if $y = \Lambda$, then $xq\Delta$

$(Q \cup \{h_a, h_r\}) \ \cap \ (\Gamma \cup \{\Delta\}) = \emptyset$

Notation:

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

$x\underline{y} = x\underline{y}\Delta = x\underline{y}\Delta\Delta$

if $y = \Lambda$, then $x\underline{\Delta}$

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

if $y = \Lambda$, then $xq\Delta$

move: $xqy \vdash_T zrw \quad xqy \vdash_T^* zrw$

$xqy \vdash zrw \quad xqy \vdash^* zrw$

example: configuration $aabqa\Delta a$ and $\delta(q, a) = (r, \Delta, L)$ ...

initial configuration corresponding to input x: ...

Notation:

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

$x\underline{y} = x\underline{y}\Delta = x\underline{y}\Delta\Delta$

if $y = \Lambda$, then $x\underline{\Delta}$

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

if $y = \Lambda$, then $xq\Delta$

move: $xqy \vdash_T zrw \quad xqy \vdash_T^* zrw$

$xqy \vdash zrw \quad xqy \vdash^* zrw$

example: configuration $aabqa\Delta a$ and $\delta(q, a) = (r, \Delta, L)$

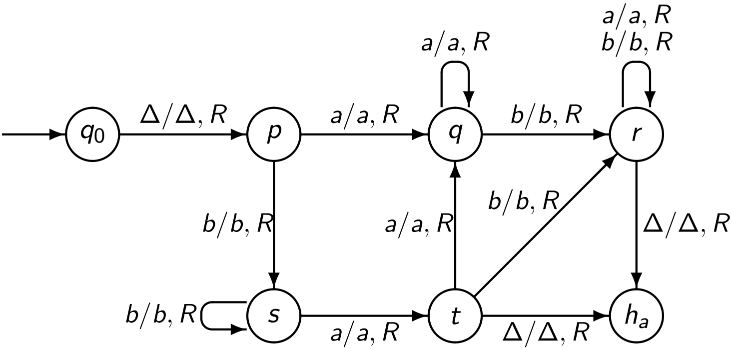
$aabqa\Delta a \vdash aarb\Delta\Delta a$

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

A slide from lecture 11

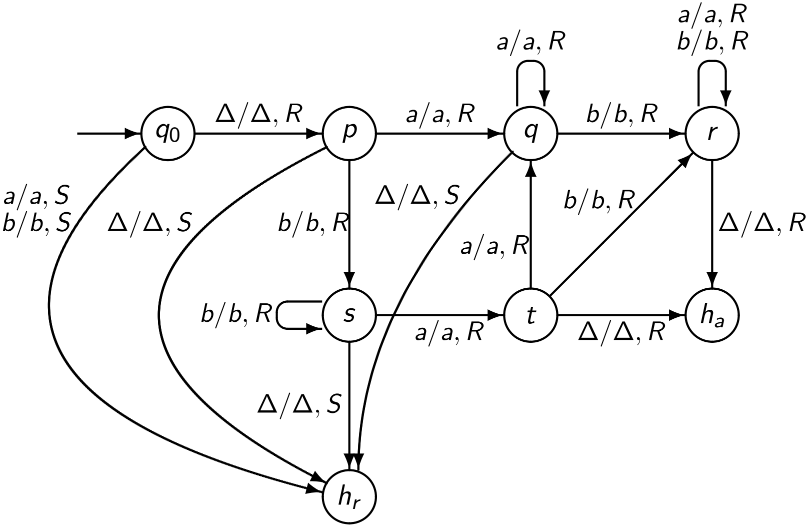
Example 7.3. A TM Accepting a Regular Language

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

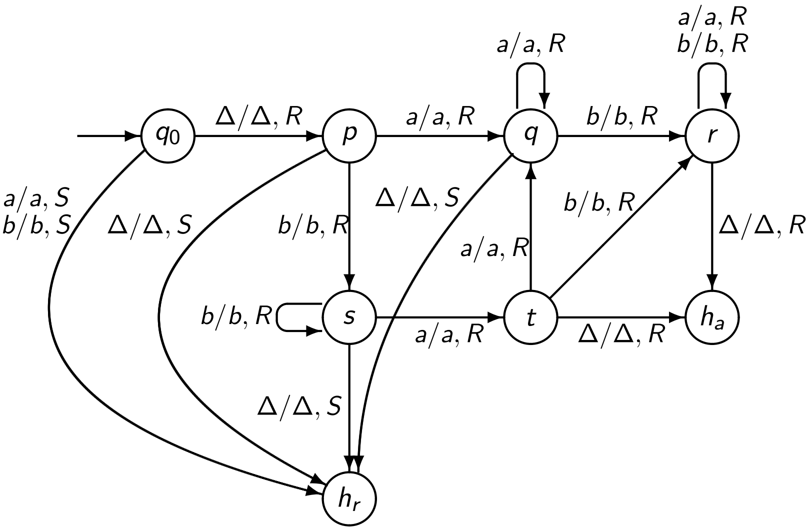


Example 7.3. A TM Accepting a Regular Language

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

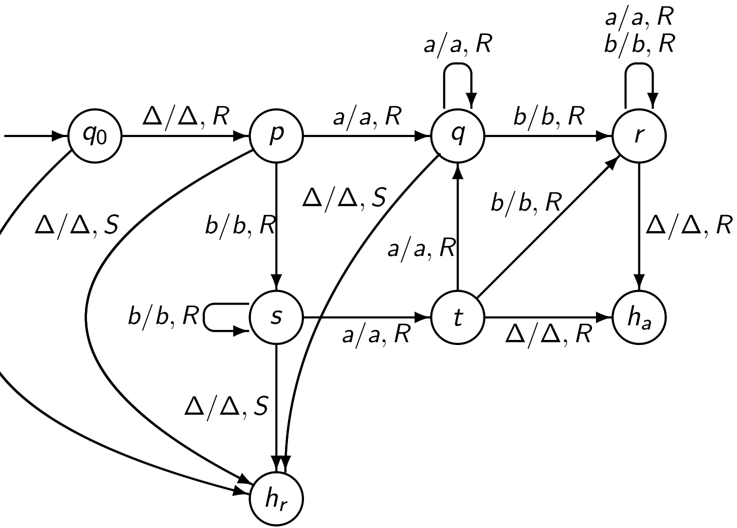


Example 7.3. A TM Accepting a Regular Language



$$q_0\Delta aaba \vdash \Delta paaba \vdash \Delta aqaba \vdash \Delta aaqba \vdash \Delta aabra \vdash \Delta aabar\Delta \vdash \Delta aaba\Delta h_a\Delta$$

Example 7.3. A TM Accepting a Regular Language



$$q_0\Delta bbaa \vdash \Delta pbbaa \vdash \Delta bsbaa \vdash \Delta bbsaa \vdash \Delta bbata \vdash \Delta bbaaq\Delta \vdash \Delta bbaah_r\Delta$$

Properties of notation xqy

- all information about configuration in one string
- move TM yields only local change

Useful, when ...

Definition 7.2. Acceptance by a TM
If $T = (Q, \Sigma, \Gamma, q_0, \delta)$ is a TM and $x \in \Sigma^*$,
x is accepted by T if

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

for some strings $w, y \in (\Gamma \cup \{\Delta\})^*$
(i.e., if, starting in the initial configuration corresponding to input x, T eventually halts in the accepting state).
N.B.: sequence of moves leading to h_a is unique
A language $L \subseteq \Sigma^*$ is accepted by T if $L = L(T)$, where

$$L(T) = \{x \in \Sigma^* \mid \ x \text{ is accepted by } T \ \}$$

If $x \notin L(T)$, ...

7.3. Turing Machines That Compute Partial Functions

$$f(x, y) = x/y$$

with real numbers
Partial function:
$f(x, y)$ is not defined if $y = 0$,
i.e., if $y = 0$, then (x, y) is not in domain of f

Example 7.10. The Reverse of a String

$$\underline{\Delta}aabab$$

Example 7.10. The Reverse of a String

$$\begin{array}{l}
\underline{\Delta}\, a\, a\, b\, a\, b \\
\Delta A\, a\, b\, a\, b \\
\Delta A\, a\, b\, a A \\
\Delta B\, a\, b\, a A \\
\Delta B A\, b\, a A \\
\Delta B A\, b A A \\
\Delta B A\, b A A \\
\Delta B A B A A \\
\underline{\Delta}\, b\, a\, b\, a\, a
\end{array}$$

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:

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

Simple version of:

Definition 7.9. A Turing Machine Computing a Function

Let $T = (Q, \Sigma, \Gamma, q_0, \delta)$ be a Turing machine, and f a partial function on Σ^* with values in Γ^*. We say that T computes f if for every x in the domain of f,

$$q_0 \Delta x \vdash_T^* h_a \Delta f(x)$$

and no other input string is accepted by T.

Definition 7.9. A Turing Machine Computing a Function
Let $T = (Q, \Sigma, \Gamma, q_0, \delta)$ be a Turing machine, k a natural number, and f a partial function on $(\Sigma^*)^k$ with values in Γ^*. We say that T computes f if for every $(x_1, x_2, \dots, x_k)$ in the domain of f,

$$q_0 \Delta x_1 \Delta x_2 \Delta \dots \Delta x_k \vdash_T^* h_a \Delta f(x_1, x_2, \dots, x_k)$$

and no other input that is a k-tuple of strings is accepted by T.
A partial function $f : (\Sigma^*)^k \to \Gamma^*$ is Turing-computable, or simply computable, if there is a TM that computes f.

k can be 0...

Functions on natural numbers...

Example 7.12. The Quotient and Remainder Mod 2

Example.
Dividing by 3

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

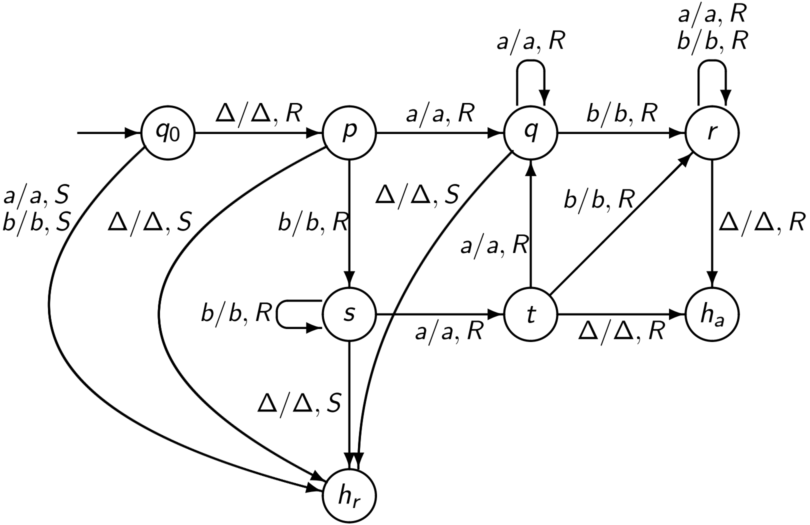
Exercise.
Draw a TM that computes the function $f(x, y) = x \bmod y$

Hint: implement the following algorithm:

while $(x \geq y)$
  $x = x - y;$

Make this exercise yourself.

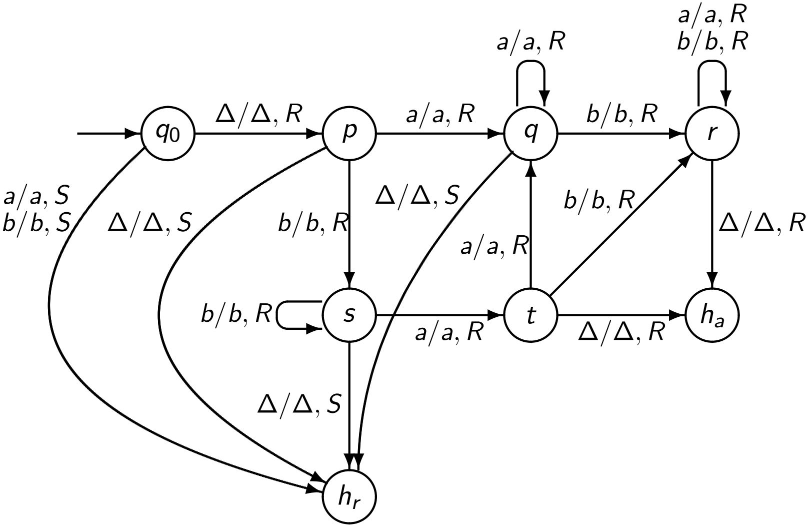
Example 7.3. A TM Accepting a Regular Language
$L = \{a, b\}^*\{ab\}\{a, b\}^* \cup \{a, b\}^*\{ba\}$



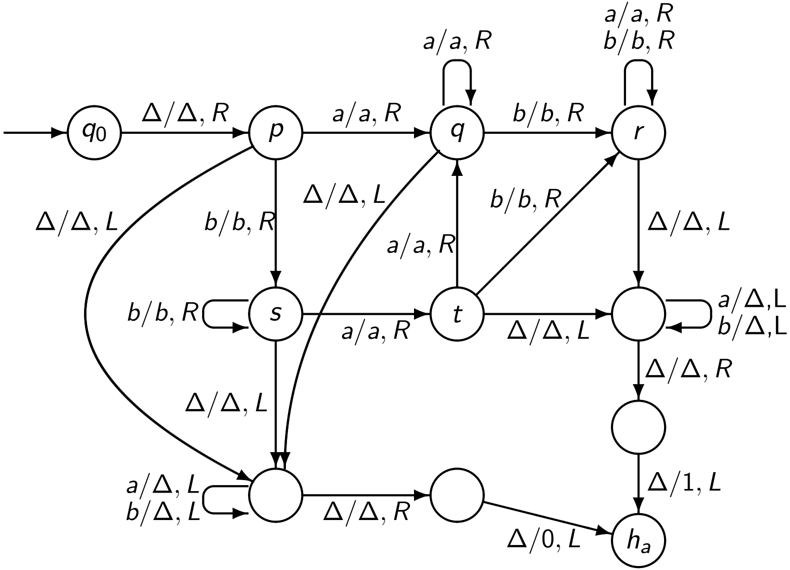
Example 7.14. The Characteristic Function of a Set
$L = \{a, b\}^*\{ab\}\{a, b\}^* \cup \{a, b\}^*\{ba\}$
$\chi_L : \{a, b\}^* \to \{0, 1\}$, defined by

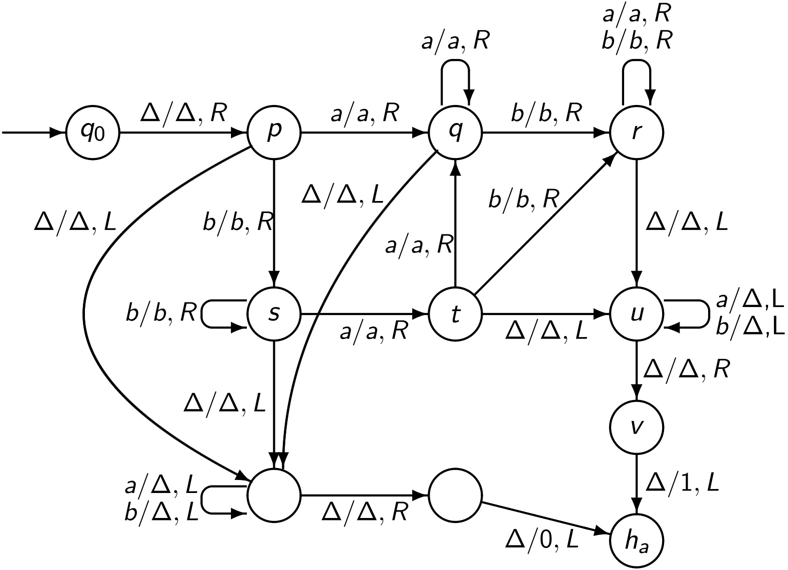
$$\chi_L(x) = \begin{cases} 1 & \text{if } x \in L \\ 0 & \text{if } x \notin L \end{cases}$$

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