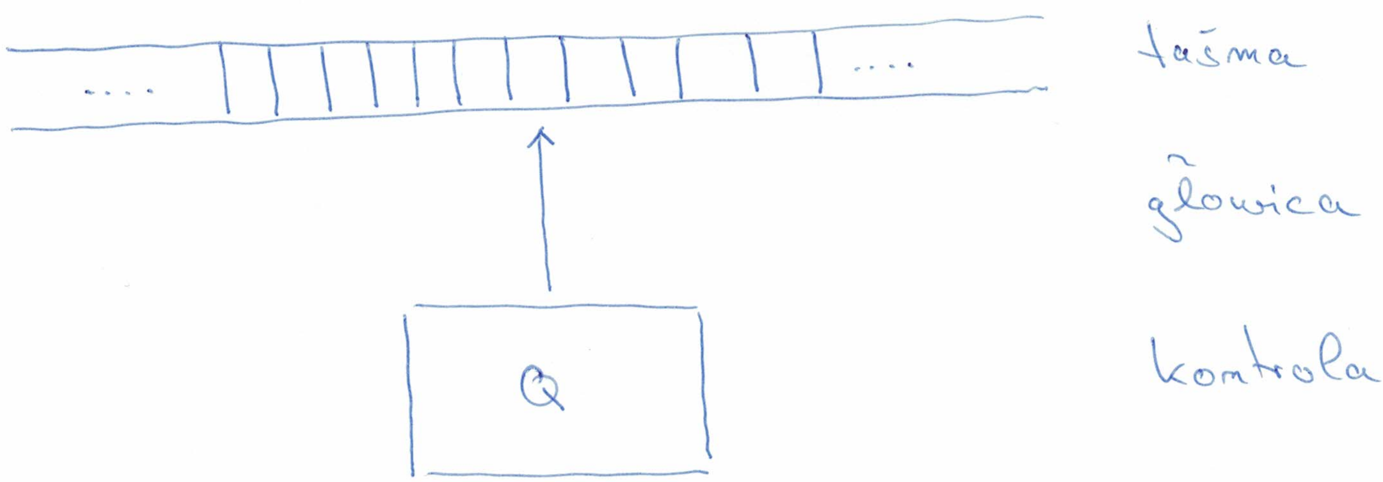
§2 Maszyny Turinga

- Alan M. Turing,
  On Computable Numbers with an Application to the Entscheidungsproblem;
  Proc. London Math. Soc. 42 (1936), 230 – 265
- prosta maszyna, która "manipuluje" litery na taśmie
- "zdefiniuje" jakie funkcje są obliczalne.
  (teza Churcha-Turinga)



na polach taśmy: $a \in \Sigma$ lub b ("blank") $\notin \Sigma$

na prawie wszystkich polach: b

krok obliczenia:

wpisanie $a \in \Sigma \cup \{b\}$ na aktualne pole

lub

przesunięcie (głowicy) o jedno pole na lewo lub prawo czyli na Maszynę Turinga składają się

- alfabet Σ
- skończony zbiór stanów Q
- stan początkowy $q_0 \in Q$
- tablica Turinga T

Tablica Turinga (też: funkcja przejścia) opisuje dla aktualnego stanu i aktualnego pola (pokazuje na niego głowica)

- jaka operacja zostaje wykonana i
- jaki jest kolejny aktualny stan

możliwe operacje, to

a: pisz $a \in \Sigma \cup \{b\}$ na pole aktualne

r: przesuń głowicę o jedno pole na prawo

l: przesuń głowicę o jedno pole na lewo

s: stop

2.1 Definicje

Def

Maszyna Turinga (TM), to 4-elementowa krotka

$$M = (\Sigma, Q, q_0, T)$$

gdzie

Q: skończony zbiór

$q_0 \in Q$

Σ: alfabet, $b, l, r, s \notin \Sigma$

$T : O \times \Sigma_0 \longrightarrow O \times Q$

$\Sigma_0 := \Sigma \cup \{b\}$,

$O := \{r, l, s\} \cup \Sigma_0$

Informacja na taśmie jest skończona (bo prawie wszystkie pole zawierają b). $\longrightarrow$ Aktualna sytuacja obliczenia

$$\ldots |b|b| \quad u \quad |a| \quad v \quad |b|b| \ldots$$

$$\uparrow$$

$$q$$

Def: $M = (\Sigma, Q, q_0, T)$

a) Konfiguracja k (nad M), to słowo $k = uqv$, gdzie $u \in \Sigma_0^*$, $q \in Q$, $v \in \Sigma_0^* \setminus \{\lambda\}$.

b) Dwie konfiguracje k i k' są równoważne, jeżeli istnieją $n, m, p, r \in \mathbb{N}$, takie że

$$b^n k b^m = b^p k' b^r.$$

Obliczenie TM śledzi się z konfiguracjami:

<u>Def</u> $M = (\Sigma, Q, q_0, T)$

k, k' konfiguracje nad M

a) k' jest bezpośrednią kolejną konfiguracją k ($k \vdash k'$ lub $k \to k'$), jeżeli

k	$T(q,a)$	k'	
$uqav$	(a', q')	$uq'a'v$	
$uqav$	(r, q')	$uaq'v$	$(v \neq \lambda)$
uqa	(r, q')	$uaq'b$	
$ua'qav$	(l, q')	$uq'a'av$	
qav	$(l' q')$	$q'bav$	
$uqav$	(s, q')	$uqav$	

b) k' jest kolejną konfiguracją k, jeżeli istnieją konfiguracje $k_0, \dots k_n$, $n \geq 0$, takie że $k = k_0 \vdash k_1 \vdash \dots \vdash k_n = k'$

(piszemy: $k \overset{*}{\vdash} k'$ lub $k \overset{*}{\to} k'$)

c) $k = uqav$ jest konfiguracją końcową, jeżeli $T(q,a) = (s, q')$.

(mówimy: M zatrzyma się w konfiguracji k)

Uwaga: w tym przypadku mamy $k' = k$ dla wszystkich k', takie że $k \overset{*}{\vdash} k'$!

Maszyny Turinga obliczą funkcje
$f: \mathbb{N}^n \rightsquigarrow \mathbb{N}^m$ (lub $f: W^n \rightsquigarrow W^m$ gdzie $W = \Sigma^*$)

<u>Def</u>: $M = (\Sigma, Q, q_0, T)$
$W = \Sigma^*$

a) M obliczy funkcję $f: W^n \rightsquigarrow W^m$, jeżeli
(i) M zatrzyma się startując z konfiguracji
$$k_0 = q_0 b x_1 b x_2 b \dots b x_n b$$
wtw. $(x_1, \dots x_n) \in Def(f)$.

(ii) Jeżeli $f(x_1, \dots x_n) = (y_1, \dots y_m)$,
M zatrzyma się startując z k_0 w konfiguracji końcowym
$$k_e = q_i b y_1 b y_2 b \dots b y_m b$$

b) Funkcja $f: W^n \rightsquigarrow W^m$ jest obliczalna, jeżeli istnieje maszyna Turinga M, która obliczy f.

Funkcję, którą maszyna Turinga M obliczy, oznaczamy jako f_M.

Przykłady: $\Sigma = \{a_1, a_2\}$, $W = \Sigma^*$

(i) TM, która usuwa pierwsze słowo $w \in \Sigma^* \setminus \{\lambda\}$ na prawo od głowicy:

	b	a_1	a_2
q_0	(r, q_0)	(a_1, q_1)	(a_2, q_1)
q_1	(s, q_1)	(b, q_2)	(b, q_2)
q_2	(r, q_1)		

$$q_0 b b a_1 a_2 b a_2 b$$
$$\to b q_0 b a_1 a_2 b a_2 b$$
$$\to b b q_0 a_1 a_2 b a_2 b$$
$$\to b b q_1 a_1 a_2 b a_2 b$$
$$\to b b q_2 b a_2 b a_2 b$$
$$\to b b b q_1 a_2 b a_2 b$$
$$\to b b b q_2 b b a_2 b$$
$$\to b b b b q_1 b a_2 b$$

M obliczy funkcję $f: W^n \to W^{n-1}$

$$f(x_1, \ldots x_n) = (x_2, \ldots x_n)$$

(ii) TM, która usuwa ostatnią literę pierwszego słowa na prawo (i "wraca") (16)

<u>struktura:</u>

w q_0: szukaj pierwszego $a \neq b$ na prawo i do q_1

q_1: jeżeli $w \neq 1$ then do q_2 else do q_4

q_2: szukaj pierwszego b na prawo i do q_3

q_3: pisz b i do q_4

q_4: szukaj pierwszego b na lewo i stop

	b	a_1	a_2
q_0	(r, q_0)	(a_1, q_1)	(a_2, q_2)
q_1	(l, q_4)	(r, q_2)	(r, q_2)
q_2	(l, q_3)	(r, q_2)	(r, q_2)
q_3	(l, q_4)	(b, q_3)	(b, q_3)
q_4	(s, q_4)	(l, q_4)	(l, q_4)

$q_0 b a_1 a_2 b b$

$\to b q_0 a_1 a_2 b b$

$\to b q_1 a_1 a_2 b b$

$\to b a_1 q_2 a_2 b b$

$\to b a_1 a_2 q_2 b b$

$\to b a_1 q_3 a_2 b b$

$\to b a_1 q_3 b b b$

$\to b q_4 a_1 b b b$

$\to q_4 b a_1 b b b$

M obliczy funkcję $f: W \to W$

$$f(x) = \begin{cases} \lambda; & x = \lambda \\ a_1 \dots a_{n-1}; & x = a_1 \dots a_n \end{cases}$$

lub – bo $|\Sigma| = 2$

$f: \mathbb{N} \to \mathbb{N}$, $f(x) = x \bmod 2$

2.2 Uniwersalna Maszyna Turinga

Czy istnieje maszyna Turinga M_0 taka, że jest w stanie "symulować" wszystkie maszyny Turinga?

M_0 dostaje wejścia "opisowanie" maszyny Turinga M oraz $w \in \Sigma^*$ i ma obliczyć $f_M(w)$.

Kodowanie c maszyny Turinga

$$M = (\Sigma, Q, q_0, T)$$

$$b, \#, :, + \notin \Sigma \;, \quad \Delta := \Sigma_0 \cup \{\#, :, +\} \text{ alfabet } M_0$$

$T(a, q_i) = (o, q_j)$ kodujemy przez

$$\# a \# 1^i \# o \# 1^j \#$$

a całą funkcję T – czyli M – przez

$$c(T) := + \# a \# 1^i \# o \# 1^j \# \# a' \# 1^{i'} \# o' \# 1^{j'} \# \ldots\ldots \# +$$

konfigurację $u q_i v$ kodujemy przez

$$\# u \# 1^i \# v \#$$

Maszyna M_0 startuje z taśmą

$$\ldots \,|\, b \,|\, c(T) \,|\, b \,|\, w \,|\, b \,|\, \ldots$$

którą zmieni na

$$\ldots \,|\, b \,|\, c(T) \,|\, b \,|\, + \,|\, + \,|\, \#\# 1^0 \# w \# \,|\, b \,|\, \ldots$$

M_0 symuluje M krok po kroku, tzn.
dla aktualnego stanu q_i i aktualnej głowicy a
maszyny M, maszyna M_0 znajduje

$$\#\#a\#1^i\#o\#1^j\#\#$$

na taśmie
i wykonuje o (na konfiguracje), n.p.
zmieni

$$++\#u\#1^i\#v\# \quad \text{na} \quad ++\#u'\#1^j\#av\#$$

jeżeli $l=o$ i $u=u'a$.
Potem znajduje $\#\#a\#1^j\#o'\#1^j\#\#$ itd.

Wynik

Istnieje maszyna Turinga M_0 taka, że dla dowolnej maszyny Turinga M

(i) M_0 zatrzyma się z wejściami $c(M)$ i $w \in \Sigma^*$ wtw. M zatrzyma się z wejściem w.

(ii) M_0 zatrzyma się w konfiguracji uq_sv wtw. M zatrzyma się w konfiguracji uq_sv.

tzn.

$$f_{M_0}(c(M), w) = f_M(w)$$

dla wszystkich maszyny Turinga M i wszystkich $w \in \Sigma^*$

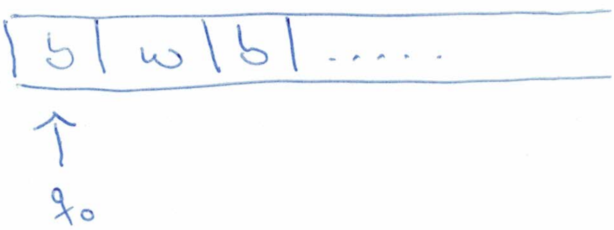
Uwaga: f_{M_0} jest obliczalna.

2.3 Inne modely

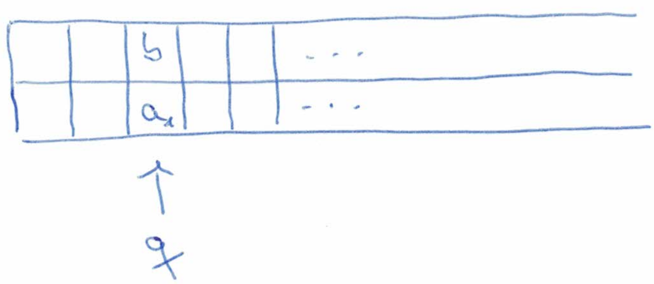
(i) Maszyna akceptująca

zamiast operacji "stop" maszyna zatrzyma się w pewnych oznaczonych stanach ("stany akceptujące")

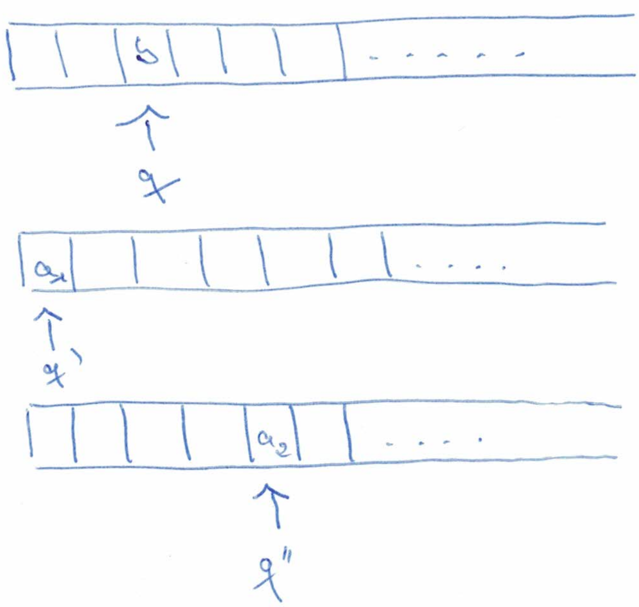
(ii) Maszyna z jednostronnie nieskończoną taśmą



(iii) Maszyna z taśmą wielościeżką



(iv) k-taśmowa maszyna



(v) Niedeterministyczna maszyna

zamiast $T : \Sigma \times Q \longrightarrow O \times Q$

posiada

$$T : \Sigma \times Q \longrightarrow \mathcal{P}(O \times Q)$$

2.4 Tesa Turinga-Chwcha

jest dużo innych modeli obliczenia (programowania), n.p.

- imperatywne (Java, C#, TM, ...)
- funkcyjne (Haskell, Lisp, funkcje μ-rekursywne
- logiczne (Prolog, FOL)
- maszyny rejestrowe (Assembler)

Okazuje się, że funkcje obliczalne w każdym modelu są dokładnie te same. Stąd

Tesa

Funkcja f jest (intuicyjnie) obliczalna wtw. istnieje maszyna Turinga M taka, że $f_M = f$.

Uwaga: Tesa nie jest twierdzeniem, bo nie ma definicji obliczalności.