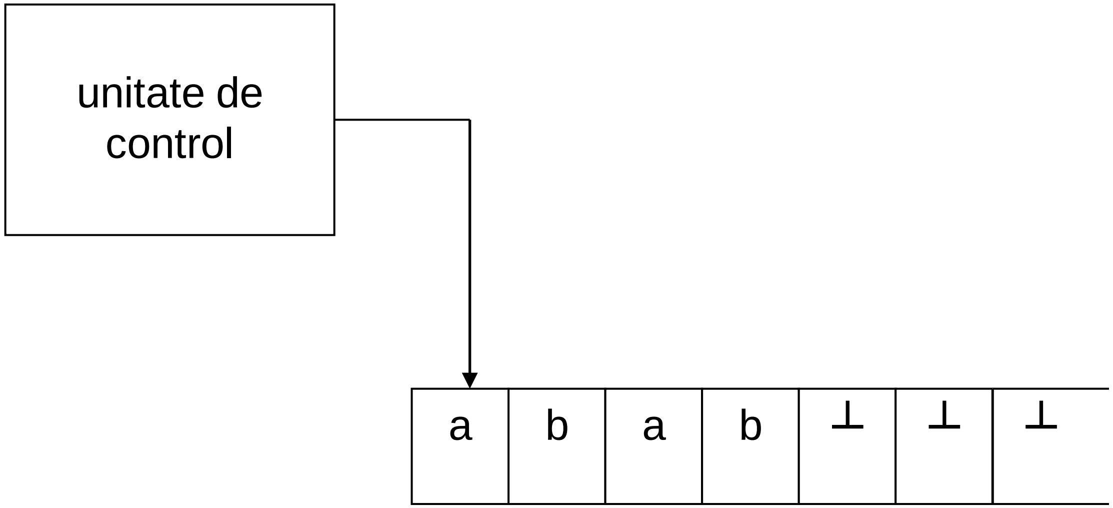
Masini Turing

1. Exemple
2. Definitia formala
3. Limbaje Turing-acceptate si limbaje Turing-decidabile

Masini Turing



Masini Turing

MT poate sa simuleze

- orice calculator real
- orice limbaj de programare.

Masini Turing

Exemplul 1

Fie L = {w#w | w$\in${0,1}*}. Construim o MT care sa testeze apartenenta unei secvente binare la L.

idee:

avem voie sa ne deplasam la stanga si la dreapta in secventa de intrare si putem "marca" un simbol, odata ce l-am examinat.

Cursorul va scana in mod repetat secventa de intrare:

- la fiecare trecere va compara un simbol din stanga cu unul situat in dreapta lui # si, daca coincid, le inlocuieste cu x;
- daca toate simbolurile din secventa au fost inlocuite cu x, atunci MT trece in una dintre starile finale de acceptare; altfel trece in una dintre starile finale de respingere.

Masini Turing

=> *algoritmul:*

(P1) se scaneaza secventa de intrare $w \in \{0,1\}^*$ in cautarea simbolului special # ;

daca simbolul este gasit, atunci se trece la pasul 2; altfel, secventa este respinsa.

(P2) se scaneaza prima pereche de simboluri cele mai din stanga din cele 2 subsecvente;

daca coincid, atunci se inlocuiesc cu x si se trece la pasul 3; altfel, secventa este respinsa.

(P3) se scaneaza urmatoarea pereche de simboluri, pana se epuizeaza simbolurile din stanga lui #;

daca la dreapta lui # mai raman simboluri binare, atunci secventa de intrare w este respinsa; altfel, w este acceptata.

Masini Turing

1. Exemple
2. Definitia formala
3. Limbaje Turing-acceptate si limbaje Turing-decidabile

Masini Turing

Definitia 1

MT = un sistem $(Q, \Sigma, \Gamma, \delta, q_0, q_a, q_r)$ unde:

Q = mulţime finită: mulţimea stărilor;

Σ = mulţime finită: alfabetul de intrare; $\Sigma \cap Q = \emptyset$;

Γ = mulţime finită: alfabetul benzii; $\Sigma \subseteq \Gamma$, $␣ \in \Gamma$, $␣ \notin \Sigma$;

$\delta : Q \times \Gamma \to Q \times \Gamma \times \{ L , R \}$: funcţia de tranziţie;

$q_0 \in Q$: starea iniţială;

$q_a \in Q$: starea finala de acceptare a secventei de intrare;

$q_r \in Q$: starea finala de respingere a secventei de intrare.

Notatie

MT = {M | M este o masina Turing}

Masini Turing

Observatie: Modul de calcul al MT:

1) Initial, MT se afla in starea q_0

si primeste pe banda, in primele n locatii din extr. stg., secventa de intrare $w=w_1w_2\ldots w_n\in\Sigma^*$.

Primul blank $\perp$ care apare pe banda marcheaza sfarsitul secv. de intrare.

2) Cursorul se afla in extremitatea stanga a benzii (in prima locatie).

$\delta(q,a)= \delta(p,b,R)$

$\Leftrightarrow$ MT, aflata in starea q, citeste pe banda de intrare simbolul a =>

- MT trece in starea p,
- inlocuieste simbolul a cu simbolul b in celula examinata si
- deplaseaza cursorul cu o celula la dreapta celulei examinate.

3) Daca MT incearca sa deplaseze cursorul dincolo de extremitatea stanga a benzii, acesta ramane in dreptul primei locatii din extremitatea stanga.

4) Calculul continua pana MT ajunge in q_a sau q_r si se opreste. Altfel, cicleaza nedefinit.

Masini Turing

Definitia 2

Configuratie a unei $M \in MT$ = un triplet format din:

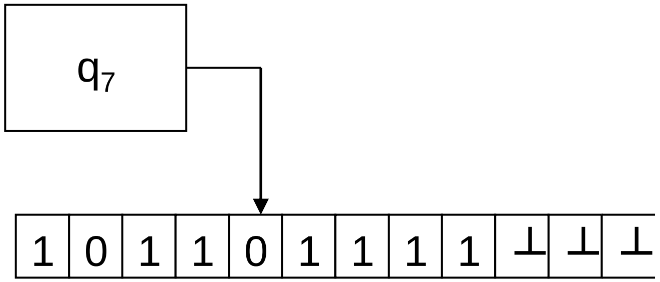
- starea curenta a M, q;
- continutul curent al benzii, v w;
- pozitia curenta a cursorului.

Notatie

vqw, $v,w \in \Gamma^*$, $q \in Q$.

Exemplu

Configuratia $1011q_701111$ inseamna



Masini Turing

Definitia 3

Configuratia C_1 produce configuratia $C_2 \Leftrightarrow$

MT trece – corect – din C_1 in C_2 intr-un singur pas $\Leftrightarrow$

Fie a,b,c $\in \Gamma$,

v,w $\in \Gamma^*$,

q_i, $q_j \in Q$

Spunem ca o configuratie vaq_ibw produce configuratia vq_jacw daca $\delta(q_i,b)=(q_j,c,L)$;

Analog: o configuratie vaq_ibw produce configuratia $vacq_jw$ daca $\delta(q_i,b)=(q_j,c,R)$.

Masini Turing

<u>Cazuri particulare de configuratii</u>:

1) Configuratia initiala: q_0w ;
2) Configuratii de oprire:
- configuratia de acceptare : $q=q_a$,
- configuratia de respingere : $q=q_r$;

3) Cursorul este in extr. stanga a benzii => config. curenta q_ibw produce:
- q_jcw daca cursorul ramane pe loc,
- cq_jw daca cursorul se deplaseaza la dreapta;

4) Cursorul este in extr. dreapta a benzii =>
config. curenta vaq_i este echivalenta cu $vaq_i\perp$.

Masini Turing

1. Exemple
2. Definitia formala
3. Limbaje Turing-acceptate si limbaje Turing-decidabile

Masini Turing

Definitia 4

$M \in MT$ M accepta secventa de intrare $w \in \Sigma^* \Leftrightarrow$

$\exists$ o succesiune (finita) de configuratii $C_1, C_2, \ldots, C_s$ astfel incat:

1) C_1 este configuratia initiala a lui M pentru intrarea w,
2) $\forall 1 \le i \le s-1$: $C_i \rightarrow C_{i+1}$,
3) C_s este o configuratie de acceptare.

Definitia 5

Fie $M \in MT$: $L(M)$ = limbajul masinii Turing M = $\{ w \in \Sigma^* \mid M \text{ accepta } w \}$.

Masini Turing

<u>Definitia 6</u>

Limbajul $L \subseteq \Sigma^*$ se numeste Turing-acceptat = recursiv enumerabil $\Leftrightarrow$

$\exists$ $M \in MT$: $L=L(M)$.

<u>Definitia 7</u>

1) $M \in MT$ se numeste decidenta $\Leftrightarrow$

M se opreste indiferent ce secventa primeste la intrare.

2) Fie $M \in MT$ si $L \subseteq \Sigma^*$; Spunem ca M decide asupra limbajului L $\Leftrightarrow$

(i) $L=L(M)$,

(ii) M este decidenta.

Masini Turing

Definitia 8

Limbajul $L \subseteq \Sigma^*$ se numeste [Turing-]decidabil = recursiv $\Leftrightarrow$

$\exists$ $M \in MT$ decidenta: $L=L(M)$.

Observatie

$\forall$ L Turing-decidabil => L este Turing-acceptat dar

(rec.) (r.e.)

$<\neq$

Masini Turing

1. Exemple
2. Definitia formala
3. Limbaje Turing-acceptate si limbaje Turing-decidabile