

TC - CURS 14

Maximile Turing au fost inventate în 1936 de Alan Turing. Familia limbajelor acceptate de maximile Turing este L_0 - familia limbajelor recursiv enumerabile.

O mașină Turing (MT, pe scurt) constă dintr-o bandă infinită (de lungime infinită).

Banda este împărțită într-un număr infinit de celule care conțin un simbol dintr-un alfabet dat (alfabetul de intrare) sau un simbol special, numit Blank.

Mașina are un cap de citire-scriere care se poate mișca în ambele direcții (stânga sau dreapta), spre deosebire de automatele finite sau pushdown.

Definiție O mașină Turing are o structură de forma:

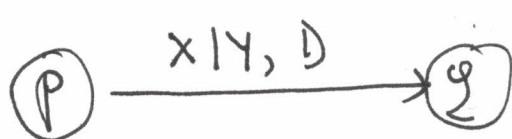
$$T = (Q, T, B, \Sigma, \delta, z_0, F), \text{ unde}$$

- Q este mulțimea (finită) a stărilor
- T este alfabetul benzii (simbolurile ce pot fi scrise pe bandă)
- $B \in T$ este simbolul blank (fiecare celulă a benzii conține inițial pe B , cu excepția

simbolului inițial de intrare).

- $\delta: Q \times T \rightarrow Q \times T \times \{L, R\}$ este funcția de tranziție. (În funcție de starea curentă, de simbolul curent către care indică capul marșului, T își schimbă starea, modifică conținutul celulei către care indică capul și apoi mută capul spre stânga (L) sau spre dreapta (R)).
- $q_0 \in Q$ este starea inițială
- $F \subseteq Q$ este mulțimea stărilor finale. Dacă marșul T ajunge în oricare dintre stările finale, șirul de intrare este acceptat.

În exemple, vom folosi notația grafică



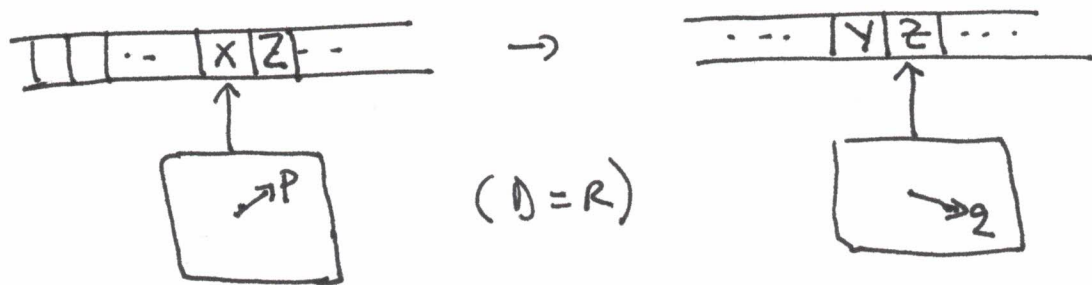
$p, q \in Q$
 $D \in \{L, R\}$
 $x, y \in T$

$$(q, y, D) = \delta(p, x)$$

Aceasta înseamnă că atunci când T este în starea p , capul de citire fixat pe o celulă în care se află x , atunci x este înlocuit cu y (posibil $y=x$), marșul trece în starea q , iar capul se mută sau o celulă la stânga (dacă $D=L$) sau o celulă

$= 3 =$

Spre dreapta (dacă $D = R$).

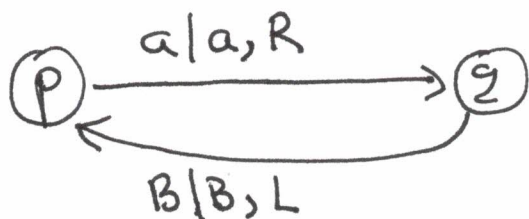


Observații.

- 1) Există o mare varietate de mașini Turing
 - mașini cu o singură bandă infinită la un capăt (mărginită la celălalt)
 - mașini cu mai multe benzi
 - mașini cu mai multe capete de citire/scriere
 - mașini nedeterminate
 - mașini cu mai multe track-uri (fiecare celulă este privită ca un vector cu un număr det de elemente)
 - orice combinație a ^{tipurilor} mașinilor de mai sus.

Toate aceste tipuri de mașini Turing s-au dovedit a fi echivalente.

- 2) Este posibil ca o mașină Turing să nu se oprească niciodată, pentru un anumit înț de intrare, cum este în exemplul următor:



unde la intrare avem doar simbolul a , deci celule din dreapta lui Q conțin B (blank), P, Q nu sunt finale.

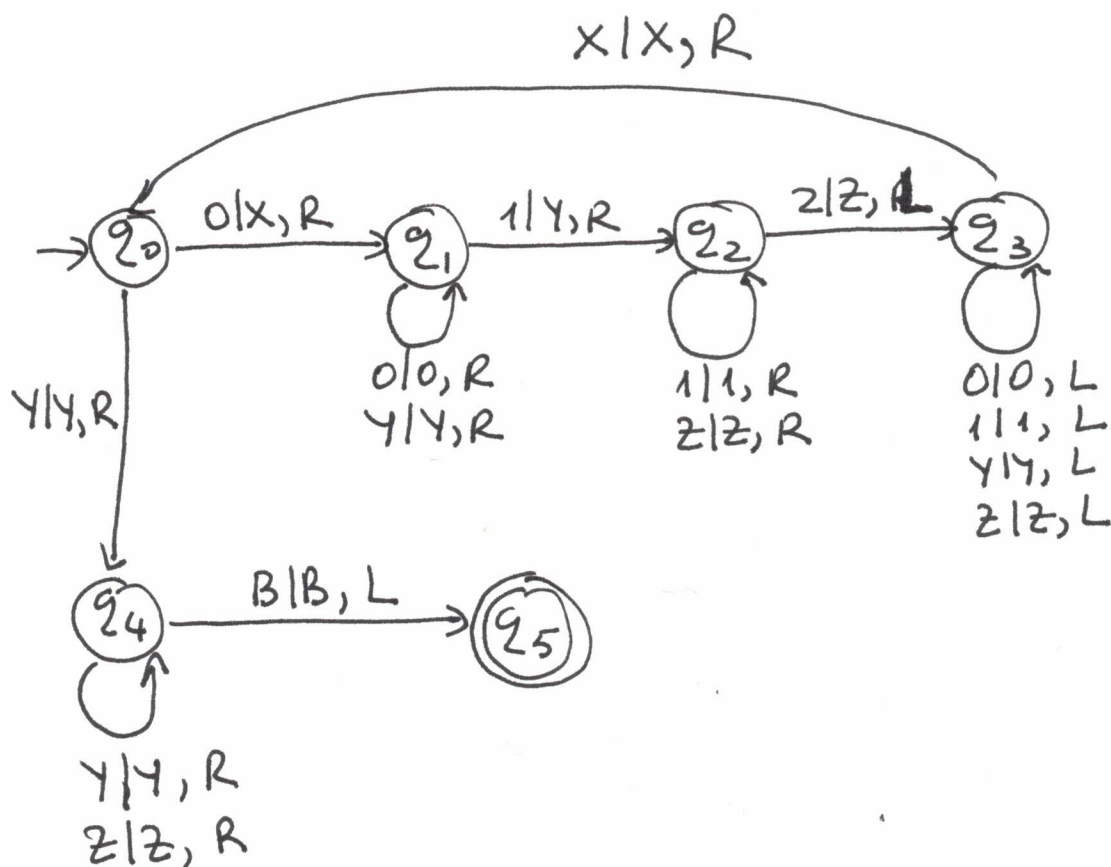
Exemplu MT care acceptă limbajul
 $L = \{ 0^n 1^n 2^n \mid n \geq 1 \}$

Input : 001122

Output : Acceptat

Input : 00011222

Output : Respins



Parul 1 Înlocuiește 0 cu X, mergi dreapta în q_1 .

Parul 2. Înlocuiește 0 cu 0, mergi dreapta, rămâi în aceeași stare.

Înlocuiește Y cu Y, mergi dreapta, rămâi în aceeași stare.

Parul 3 Înlocuiește 1 cu 1 sau 2 cu 2, mergi dreapta, rămâi în aceeași stare.

Înlocuiește 2 cu Z, mergi stânga, treci în starea q_3 .

Parul 4 Înlocuiește 0 cu 0 , 1 cu 1 , 2 cu 2 și
 4 cu 4 , mergi stânga, rămăi în aceeași
 stare.

Când întâlnești X , înlocuiești X cu X ,
 mergi dreapta, revii în q_0 .

Parul 5 Dacă simbolul curent este 0 , goto Parul 1
 Altfel, dacă simbolul curent este 4 ,
 înlocuiești 4 cu 4 , mergi dreapta,
 treci în starea q_4

Parul 6 (verifică dacă simbolul care urmează
 este de forma 4^+2^+)

Înlocuiești 4 cu 4 , 2 cu 2 , mergi
 dreapta, rămăi în q_4 .

Dacă întâlnești simbolul blank, B ,
 înlocuiești B cu B , mergi stânga,
 treci în starea finală q_5 . Simbolul
 este acceptat.

Dacă mașina se oprește în oricare dintre
 stările q_0, q_1, q_2, q_3, q_4 fără să mai
 poată face nicio mișcare, simbolul este
 respins.

AUTOMATE LINIAR MĂRGINITE

(Linear Bounded Automata - LBA)

Un automat liniar mărginit este o MT nedeterministă care satisface condițiile:

- Alfabetul benzii include două simboluri speciale, $\$$ și $\$$, care marchează limita din stânga, respectiv limita din dreapta a benzii.
- Automatul nu poate rescrie simbolurile $\$$ și $\$$.
- Automatul nu se poate deplasa la stânga lui $\$$ sau la dreapta lui $\$$.

Familia limbajelor acceptate de LBA este exact L_1 , adică familia limbajelor dependente de context.

O PROBLEMĂ (ÎNCĂ) DESCHISĂ, CELEBRĂ ÎN TEORIA AUTOMATELOR:

Familia limbajelor acceptate de LBA (adică L_1) este egală cu familia limbajelor acceptate de LBA deterministe?

= 7 =

RECAPITULARE AUTOMATE

- Puterea diferitelor tipuri de automate în recunoașterea limbajelor. Urmatorea secvență redă în ordine crescătoare puterea de acceptare a automatoelor:

$$AF < DAPD < APD < LBA < MT$$

AF - Automate finite

DAPD - Automate pushdown deterministe

APD - Automate pushdown (nedeterministe)

LBA - Automate liniear marginite

MT - Mașini Turing.

În termenii familiilor de limbaje aceste încreșnări:

$$L_{AF} \subsetneq L_{DAPD} \subsetneq L_{APD} \subsetneq L_{LBA} \subsetneq L_{MT}$$

unde $L_{AF} = L_3$

$$L_{APD} = L_2$$

$$L_{LBA} = L_1$$

$$L_{MT} = L_0$$

L_3, L_2, L_1, L_0 din ierarhia lui Chomsky

• Ecchivalente între diferite tipuri de automate

- (i) Automatele finite (AF) sunt echivalente cu:
 - \equiv APD cu stivă finită
 - \equiv MT cu bandă finită
 - \equiv MT cu bandă read only
 - \equiv MT cu bandă unidirectională
- (ii) Automatele pushdown (APD) sunt echivalente cu:
 - \equiv Automate finite cu stivă
- (iii) Mașinile Turing (MT) sunt echivalente cu:
 - \equiv APD cu stivă adițională
 - \equiv AF cu două stive

• Aplicații ale diferitelor tipuri de automate

1. Automatele finite (AF)

- pentru implementarea analizatorilor lexicali în cazul limbajelor de programare
- pentru recunoașterea pattern-urilor descrise cu ajutorul expresiilor regulate.
- pentru implementarea circuitelor combinate n, n-ventuale utilizând mașinile (automatele) Mealy n, Moore.
- folosite de editoarele de texte
- pentru implementarea spell-checker.

2. Automatele pushdown (APD)

- pentru implementarea fazei de parsing (analiză sintactică) a unui compilator
- pentru implementarea aplicațiilor stivă
- pentru evaluarea expresiilor aritmetice
- pentru rezolvarea problemei Termului din Hanoi.

3. Automatele liniar mărginite (LBA)

- pentru implementarea programării genetice
- pentru construcția arborilor sintactici în faza de analiză semantică a unui compilator

4. Mașinile Turing (MT)

- pentru rezolvarea problemelor recursive enumerabile
- în teoria complexității
- în implementarea rețelelor neuronale
- pentru implementarea unor aplicații din robotică
- pentru implementarea inteligenței artificiale.

GRAMATICI ȘI LIMBAJE FORMALE - RECAPITULARE

1) Gramaticile de tipul 0 (fără restricții)

- generoso familia limbajelor recursive enumerabile, notată cu L_0 în ierarhia Chomsky
- sunt cele mai generale gramatici
- sunt echivalente cu Mașinile Turing

2) Gramaticile de tipul 1

- generoso familia limbajelor dependente de context, notată cu L_1 în ierarhia Chomsky
- sunt echivalente cu automatele liniare mărginite (LBA)
- există mai multe tipuri de gramatici (dependente de context, semidependente de context, monotone) care generează L_1
- gramaticile atributate sunt cazuri particulare de gramatici dependente de context, utilizate în descrierea restricțiilor semantice ale unui limbaj de programare

3) Gramaticile independente de context (de tipul 2)

- generează familia limbajelor independente de context (CF), notată cu L_2 în ierarhia lui Chomsky
- sunt utilizate în descrierea sintaxei limbajelor de programare
- sunt echivalente cu automatele pushdown

4) Gramaticile regulate (de tip 3)

- generează familia limbajelor regulate, notată cu L_3 în ierarhia Chomsky
- sunt echivalente cu automatele finite
- sunt echivalente cu expresiile regulate, utilizate în descrierea lexicului unui limbaj de programare.