Variante de masini Turing

1. Terminologia de descriere a MT
2. MT cu 3 “deplasari”
3. MT cu mai multe benzi
4. MT nedeterminsite
5. Enumeratoare
6. Echivalenta modelelor de calculabilitate
7. Problema a 10-a a lui Hilbert

Variante de masini Turing

O MT poate fi descrisa in 3 moduri:

- formal:

  sunt date complet Q, Σ, Γ, δ;

- la nivelul implementarii MT:

  se utilizeaza limba naturala pt a defini:

  - modul in care se deplaseaza cursorul,
  - modul de memorare a informatiei de pe banda de lucru;

- la nivelul cel mai inalt:

  se utilizeaza limba naturala pt a descrie un algoritm, ignorand complet modul de implementare a acestuia.

Variante de masini Turing

Definim formatul si notatia utilizate pt a descrie o MT:

datele de intrare constau intotdeauna dintr-o secventa de simboluri;

daca intrarea trebuie sa fie un obiect (un graf, un polinom, un automat etc. sau o combinatie a acestora), atunci va trebui mai intai sa-l reprezentam printr-o secventa.

Intrucat codificarile alternative pot fi decodificate unele in altele de catre MT (eficient) => putem alege orice codificare pt obiectul de intrare.

Vom nota

- 1! obiect O, codificat printr-o secventa, prin <O>;
- mai multe obiecte $O_1, O_2, .., O_n$, codificate printr-o unica secventa, prin $< O_1, O_2, .., O_n >$.

Variante de masini Turing

- prima linie a textului va descrie datele de intrare ale MT;
- daca intrarea este:
- chiar o secventa de simboluri w => ea este interpretata ca un cuvant peste un alfabet si notata w;
- codificarea unui obiect <A> => MT incepe automat cu VERIFICAREA CORECTITUDINII CODIFICARII.
- algoritmul este impartit in etape → pasi de calcul numerotati;
- bloc → indentare.

Variante de masini Turing

Problema:

Sa se gaseasca un algoritm care sa verifice daca un graf neorientat oarecare este conex sau nu.

Formalizare:

Fie A = un limbaj care consta din toate secventele care reprezinta grafuri neorientate conexe =>

A = {<G> | G este un graf neorientat, conex};

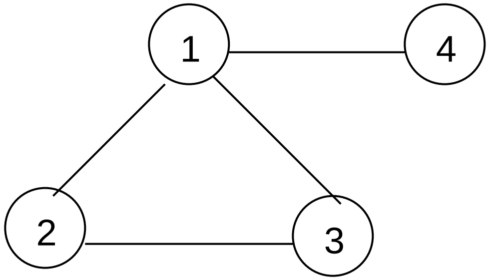
trebuie sa gasim o MT decidenta care sa decida asupra limbajului A.

M = “Fie intrarea <G> (adica: o codificare a grafului G):

1. Se selecteaza “primul” nod din G si se marcheaza.
2. Se reia pasul urmator pana cand nu se mai pot marca noi noduri:
   3. Fie un nod oarecare v din G; daca el este legat printr-o muchie de un nod deja marcat, atunci nodul v trebuie marcat si el.
4. Se scaneaza toate nodurile din G pt a verifica daca sunt toate marcate sau nu. Daca sunt marcate toate nodurile, atunci M accepta; altfel, M respinge.”

Variante de masini Turing

<u>Exemplu</u>



<G>=((1,2,3,4);(1,2),(1,3),(1,4),(2,3)).

Explicam modul de codificare a lui G:

MT incepe prin a verifica corectitudinea codificarii <G>.

Daca testul se incheie cu succes (codificarea e corecta), atunci MT trece la prima etapa a algoritmului.

Variante de masini Turing

1. Terminologia de descriere a MT
2. MT cu 3 “deplasari”
3. MT cu mai multe benzi
4. MT nedeterminsite
5. Enumeratoare
6. Echivalenta modelelor de calculabilitate
7. Problema a 10-a a lui Hilbert

Variante de masini Turing

Exemplu

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

$\delta : Q \times \Gamma \rightarrow Q \times \Gamma \times \{ L, R \}$;

Fie MT' = $(Q', \Sigma, \Gamma, \delta', q'_0, q'_a, q'_r)$ unde:

$\delta' : Q' \times \Gamma \rightarrow Q' \times \Gamma \times \{ L, R, S \}$;

$\delta'(q_i, a) = (q_j, b, S)$

$\delta (q_i, a) = (q_k, b, R)$, $\delta (q_k, b) = (q_j, b, L)$.

=> Cele 2 modele sunt computational echivalente dar:

- e nevoie de cate o o stare auxiliara suplimentara;
- sunt necesare 2 tranzitii in loc de una.

Variante de masini Turing

1. Terminologia de descriere a MT
2. MT cu 3 “deplasari”
3. MT cu mai multe benzi
4. MT nedeterminsite
5. Enumeratoare
6. Echivalenta modelelor de calculabilitate
7. Problema a 10-a a lui Hilbert

Variante de masini Turing

Definitia 1

O MT cu mai multe benzi este o MT standard

$M = (Q, \Sigma, \Gamma, \delta, q_0, q_a, q_r)$ care:

(i) are $n \geq 1$ benzi de lucru si $n \geq 1$ cursoare corespunzatoare;

(ii) initial prima banda contine secventa de intrare iar celelalte n-1 benzi sunt vide;

(iii) $\delta : Q \times \Gamma^n \rightarrow Q \times \Gamma^n \times \{ L, R, S \}^n$,

$\delta(q_i, a_1, a_2, \ldots, a_n) = (q_j, b_1, b_2, \ldots, b_n, L, R, S, L, \ldots, S)$.

Variante de masini Turing

<u>Teorema 1</u>

$\forall$ M = MT cu mai multe benzi =>

$\exists$ S = MT standard a.i. L(S) = L(M).

<u>Corolar 1</u>

$\forall L \subseteq \Sigma^*$ este Turing-acceptat $\Leftrightarrow$

$\exists$ o MT cu mai multe benzi, M, a.i. L=L(M).

Variante de masini Turing

1. Terminologia de descriere a MT
2. MT cu 3 “deplasari”
3. MT cu mai multe benzi
4. MT nedeterminsite
5. Enumeratoare
6. Echivalenta modelelor de calculabilitate
7. Problema a 10-a a lui Hilbert

Variante de masini Turing

- MTN ≈ AFN, APDN;
- Modelul de calcul: arbore;
- MTN accepta secventa de intrare ⇔

  cel putin una dintre ramurile arborelui are ca eticheta a frunzei o configuratie de acceptare.

Definitia 2

O MT nedeterminista este o MT standard

$M = (Q, \Sigma, \Gamma, \delta, q_0, q_a, q_r)$ unde:

$\delta : Q \times \Gamma \to \mathcal{P}(Q \times \Gamma \times \{L, R\})$.

Variante de masini Turing

Teorema 2

$\forall$ N = MT nedeterminista =>

$\exists$ S = MT standard a.i. L(S) = L(N).

Corolar 2

$\forall L \subseteq \Sigma^*$ este Turing-acceptat $\Leftrightarrow$

$\exists$ o MT nedeterminista, N, a.i. L=L(N).

Variante de masini Turing

Definitie 3

O MT nedeterminista se numeste decidenta $\Leftrightarrow$

$\forall w \in \Sigma^*$, toate ramurile ei de calcul se opresc.

Corolar 3

$\forall L \subseteq \Sigma^*$ este decidabil $\Leftrightarrow$

$\exists$ o MT nedeterminista decidenta, N, a.i. L=L(N).

Variante de masini Turing

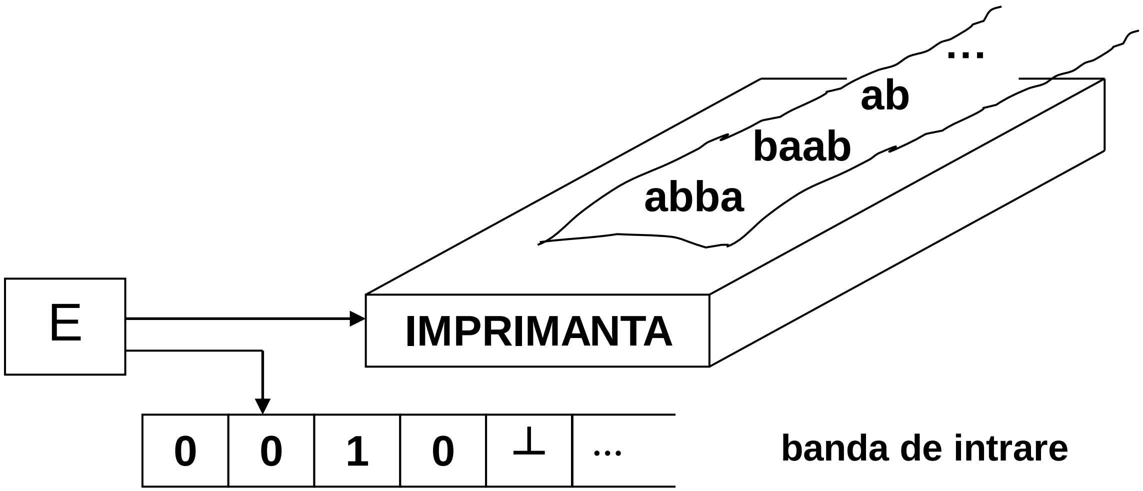
1. Terminologia de descriere a MT
2. MT cu 3 “deplasari”
3. MT cu mai multe benzi
4. MT nedeterminsite
5. Enumeratoare
6. Echivalenta modelelor de calculabilitate
7. Problema a 10-a a lui Hilbert

Variante de masini Turing

Limbajele Turing acceptate = limbaje recursiv enumerabile

Enumerator

Imprimanta



Modul de lucru:

- initial: banda de intrare este vida;
- limbajul enumerat de E: multimea tuturor secventelor de simboluri printate de imprimanta;
- situatia de ciclare: tiparirea unei liste infinite de cuvinte.

Variante de masini Turing

Teorema 3

Fie $L \subseteq \Sigma^*$; L este Turing-acceptat $\Leftrightarrow$
$\exists$ $E \in MT$ a.i. $L=L(E)$.

Variante de masini Turing

<u>Observatii</u>

1. Pentru a construi E a fost nevoie se o listare a cuvintelor din Σ: orice listare (fara repetitii) este acceptabila.
2. Evident, daca M accepta un cuvant oarecare $s \in \Sigma^*$, acesta apare la un moment dat in lista tiparita de E. De fapt, el va fi tiparit de o infinitate de ori deoarece M reia calculul de la pasul 1 pt fiecare **i**, oridecateori lista $s_1, s_2, \ldots, s_i$ se lungeste cu inca un cuvant.
3. Procedura de mai sus simuleaza rularea lui M in paralel pe toate cuvintele de intrare posibile.

Variante de masini Turing

1. Terminologia de descriere a MT
2. MT cu 3 “deplasari”
3. MT cu mai multe benzi
4. MT nedeterminsite
5. Enumeratoare
6. Echivalenta modelelor de calculabilitate
7. Problema a 10-a a lui Hilbert

Variante de masini Turing

Am definit mai multe variante de **MT** şi am demonstrat echivalenţa lor .

Există şi alte modele de calculabilitate:

- funcţiile λ-calculabile: Alonzo CHURCH, Stephen Cole KLEENE,1934;
- funcţiile general recursive: Kurt Godel, 1935
- sistemele (masinile) Post: Emil POST, 1936
- algoritmii normali Markov: A.A.MARKOV, 1954.

Toate au aceeaşi caracteristică: acces nerestricţionat la o memorie nelimitată.

În plus, toate sunt echivalente cu **MT** (şi deci unele cu altele)

=> clasa algoritmilor pe care o descriu este unică şi naturală.

vezi si clasa limbajelor de programare.

Variante de masini Turing

1. Terminologia de descriere a MT
2. MT cu 3 “deplasari”
3. MT cu mai multe benzi
4. MT nedeterminsite
5. Enumeratoare
6. Echivalenta modelelor de calculabilitate
7. Problema a 10-a a lui Hilbert

Variante de masini Turing

Algoritm: o notiune primara

Problema a 10-a a lui Hilbert

Variante de masini Turing

Congresul Internaţional al matematicienilor, 1900, Paris;

Conferinta lui David HILBERT: lista celor 23 probleme

să se găsească un algoritm care să verifice dacă un polinom în oricâte variabile admite o rădăcină în **Z**.

<u>Observatii</u>

- "… un proces cu ajutorul căruia, după un număr finit de operaţii, să se determine …."
- Hilbert presupunea că algoritmul există si trebuie doar descoperit.
- Lipsa unei definitii formale pentru notiunea de algoritm → solutii pentru cazuri particulare ale problemelor dar nu pentru o clasa intreaga.

Variante de masini Turing

Teza Church-Turing

Clasa algoritmilor (functiilor intuitiv calculabile) coincide cu clasa functiilor λ-calculabile (functiilor calculabile cu MT).

Problema a 10a a lui Hilbert.

1970, Yuri MATIJASEVIČ, (Martin DAVIS, Hilary PUTNAM, Julia ROBINSON):

nu există nici un algoritm care să verifice dacă un polinom oarecare are radacini intregi $\Leftrightarrow$

Problema a 10a a lui Hilbert este nerezolvabilă algoritmic.

Variante de masini Turing

Fie D = {p | p este un polinom care admite cel putin o radacina in Z}

D este decidabil ?

D este Turing–acceptat dar nu este decidabil.

<u>Cazul unar:</u>

Fie D_1={p_1|p_1 este un polinom intr-o singura variabila x, care admite cel putin o radacina intreaga}.

Construim o MT, M_1, care accepta D_1:

M_1 = “Fie p_1 un polinom oarecare in variabila x:

1. Se evalueaza p_1 succesiv pentru x=0, x=1, x=-1, x=2, x=-2, x=3, …,
2. Daca la un moment oarecare se obtine 0, atunci M_1 accepta p_1.”

Variante de masini Turing

<u>Cazul general:</u>

putem construi o MT, M, similara care va testa fiecare polinom de intrare p pentru diferite combinatii de valori, in functie de numarul de variabile.

n=2: $(x_1,x_2) \in \{(0,0), (0,1), (1,0), (1,1), (0,2), (2,0), (1,2), (2,1), (2,2), (0,3), (3,0),\ldots\}$;

n=3: $(x_1,x_2,x_3) \in \{(0,0,0), (0,0,1), (0,1,0), (0,1,1), (1,0,0), (1,0,1),\ldots$

Observam ca: M_1 si M recunosc limbajele D_1, respectiv D, dar nu decid asupra lor.

Variante de masini Turing

Teorema 4

Daca un polinom p de o singura variabila admite radacini intregi,

acestea se afla in intervalul $\left[-k\cdot\frac{|c_{\max}|}{|c_1|}, k\cdot\frac{|c_{\max}|}{|c_1|}\right]$

unde: k=nr de termeni (monoame) din p;

c_{max} = coeficientul cel mai mare in valoare absoluta

c_1= coeficientul termenului de grad maxim.

Variante de masini Turing

=> M_{11} = “Fie p_1 un polinom oarecare in variabila x:

1. Se calculeaza $t_2= k*|c_{max}/c_1|$, $t_1=-t_2$ si $z=[t_2]$.
2. Se evalueaza p_1 pentru z.
3. Daca $p_1(z)=0$, atunci M_{11} accepta; altfel, z:=z-1.
4. Daca $z<t_1$ atunci M_{11} respinge; altfel, reia de la P2.”

Teorema lui MATIJASEVIČ

Variante de masini Turing

1. Terminologia de descriere a MT
2. MT cu 3 “deplasari”
3. MT cu mai multe benzi
4. MT nedeterminsite
5. Enumeratoare
6. Echivalenta modelelor de calculabilitate
7. Problema a 10-a a lui Hilbert