Complexitatea modelelor de calcul

1. Terminologie
2. Notatia asimptotica
3. Analiza algoritmilor
4. Complexitatea modelelor de calcul

Complexitatea modelelor de calcul

Problema rezolvabila algoritmic în principiu

Problema nerezolvabilă algoritmic în practică

Timpul de calcul

Spaţiul de memorie

Complexitatea modelelor de calcul

Fie limbajul

$$L_k = \{w \in \Sigma^* | \exists\ k \geq 0: w = 0^k 1^k\};$$

$L_k \in$ LIC si

ACC_{LIC} este decidabil

=> L_k = decidabil.

Problema care ne intereseaza acum este insa:

“dupa cat timp o MT raspunde afirmativ la aceasta întrebare?”

Complexitatea modelelor de calcul

(I) Fie $M_1 \in MT$, o MT clasica care calculeaza L_k.

M_1 = "Fie cuvântul de intrare $w \in \Sigma^*$:

1. Se examinează banda de intrare şi se respinge w dacă se descoperă un simbol 0 la dreapta unui simbol 1.
2. Se execută etapa următoare, atât timp cât pe bandă există atât simboluri 0 cât şi simboluri 1:
   3. Se scanează banda şi se barează un singur simbol 0 şi un singur simbol 1.
4. Dacă, după ce toate simbolurile 0 au fost barate pe bandă au mai rămas simboluri 1 nebarate, sau dacă, după ce toate simbolurile 1 au fost barate pe bandă au mai rămas simboluri 0 nebarate, atunci M_1 respinge w.

   Altfel, dacă toate simbolurile 0 şi toate simbolurile 1 de pe bandă au fost barate, atunci M_1 acceptă secvenţa w."

Complexitatea modelelor de calcul

Numărul de paşi executaţi de un algoritm poate depinde de mai mulţi parametri.

Exemplu: data de intrare este un graf →

numărul de paşi: numărul de noduri / muchii, de gradul maxim al grafului, de o combinaţie a acestor factori şi / sau a altora.

Aici, TIMPUL DE EXECUTIE: funcţie de LUNGIMEA SECVENTEI

Se pot aborda:

- <u>cazul cel mai nefavorabil</u>;
- <u>cazul cel mai favorabil</u>;
- <u>cazul mediu</u>.

Complexitatea modelelor de calcul

1. Terminologie
2. Notatia asimptotica
3. Analiza algoritmilor
4. Complexitatea modelelor de calcul

Complexitatea modelelor de calcul

Definitie 1

Fie M o MT determinista cu 1! banda, decidenta,

$w \in \Sigma^*$ un cuvant de intrare,

$C_0 C_1 C_2 \ldots C_d C_f$. cele d+2, $d \geq 0$, configuratii prin care trece M pentru a prelucra cuvantul w;

Complexitatea timp a M pe intrarea w este numarul intreg notat

$TIME_M(w) = d+1$

Definitie 2

Fie M o MT determinista cu 1! banda decidenta.

Complexitatea timp ≡ timpul de executie al M este o functie

$f : N \to N$, definita prin:

f(n) = numărul maxim de paşi executaţi de M pentru o secvenţa de intrare de lungime n, $\forall n \in N$.

Notatia 1

$f(n) = \max \{TIME_M(w) \mid w \in \Sigma^n\} = TIME_M(n)$

Complexitatea modelelor de calcul

Exemplul 1

Fie $f : N \rightarrow N$, $f(n) = 5n^3 + 2n^2 + 22n + 6$;

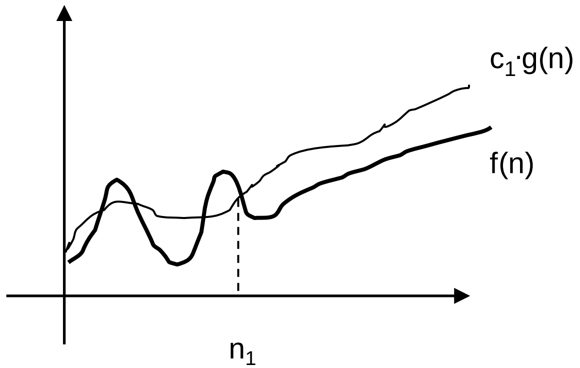
spunem că f tinde asimptotic către n^3 şi notăm acest lucru cu $\boldsymbol{O}(n^3)$

Definiţie 3

Fie $f, g : N \rightarrow R_+$

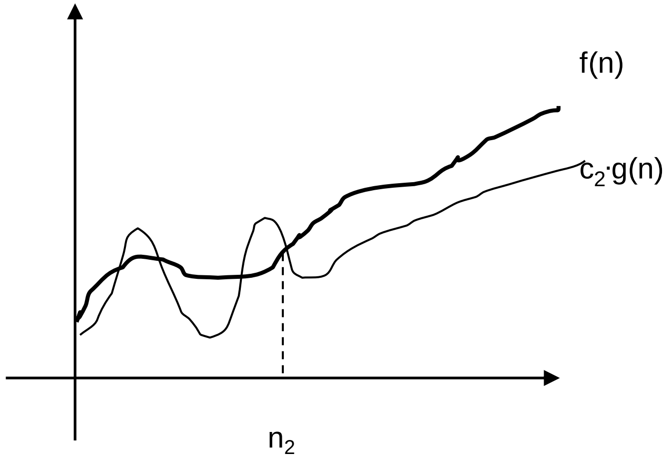
(i) f(n) = O(g(n)) şi citim "f(n) este de ordin cel mult g(n)" sau "f(n) este O mare de g(n)" $\Leftrightarrow$

($\exists$) constantele $c_1 > 0$ şi $n_1 \in N$ astfel încât $f(n) \leq c_1.g(n)$, ($\forall$) $n \geq n_1$.



Complexitatea modelelor de calcul

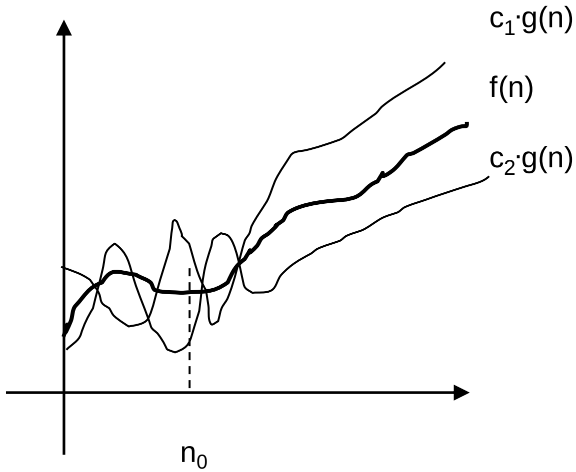
(ii) $f(n) = \Omega\ (g(n))$ şi citim "f(n) este de ordin cel puţin g(n)" sau "f(n) este omega mare de g(n)" $\Leftrightarrow$ $(\exists)$ constantele $c_2 > 0$ şi $n_2 \in N$ astfel încât $f(n) \geq c_2.g(n)$, $(\forall)\ n \geq n_2$.



Complexitatea modelelor de calcul

(iii) $f(n) = \Theta(g(n))$ şi citim "f(n) este de ordin g(n)" sau "f(n) este theta de g(n)" $\Leftrightarrow$

$f(n) = O(g(n))$ şi $f(n) = \Omega(g(n))$.



Spunem că g este o limită asimptotică superioară,

o limită asimptotică inferioară, respectiv

o limită asimptotică pentru f.

Complexitatea modelelor de calcul

<u>Exemplul 2</u>

Revenim la notaţia ***O*** şi la funcţia polinomială

$$f(n) = 5n^3 + 2n^2 + 22n + 6 \Rightarrow$$

$f(n) = \boldsymbol{O}(n^3)$, de exemplu, pentru $c_1 = 6$ şi $n_1 = 10$;

$f(n) = \boldsymbol{O}(n^4)$, de exemplu, pentru $c_1 = 1$ şi $n_1 = 6$ sau

pentru $c_1 = 36$ şi $n_1 = 1$;

$f(n) \neq \boldsymbol{O}(n^2)$,

presupunem prin absurd că există $c_1 > 0$ şi $n_1 \in N$ astfel încât

$5n^3 + 2n^2 + 22n + 6 \leq c_1.n_2, (\forall)\ n \geq n_1 \Leftrightarrow$

$5n^3 + (2- c_1).n^2 + 22n + 6 \leq 0$ etc.

Complexitatea modelelor de calcul

Exemplul 3

Fie $f_1 : \mathbf{N} \to \mathbf{N}$, $f_1(n) = 3n.\log_2 n + 5n.\log_2(\log_2 n) + 2 \Rightarrow$

$f_1(n) = \boldsymbol{O}(n.\log n)$ pentru că log n domină log(log n).

Analog, $f_2(n) = \boldsymbol{O}(n^2) + \boldsymbol{O}(n) \rightarrow f_2(n) = \boldsymbol{O}(n^2)$

pentru că $\boldsymbol{O}(n^2)$ domină $\boldsymbol{O}(n)$.

Observaţia 1

A) Specificarea bazei logaritmilor nu este necesară ea intervenind cu cel mult un coeficient constant, conform formulei: $\log_a x = \frac{\log_b x}{\log_b a}$

B) Analog, nici specificarea bazei exponenţialei nu este necesară pentru că:

$$\forall x > 0:\ x = 2^{\log_2 x} \rightarrow n^c = 2^{c \cdot \log_2 n} \Rightarrow$$

$2^{\boldsymbol{O}(\log n)}$ este o limită superioară pentru n^c, unde c este o constantă oarecare. Evident, şi $2^{\boldsymbol{O}(n)}$ este o limită superioară pentru n^c.

Complexitatea modelelor de calcul

Observaţia 2

Limitele asimptotice de tipul n^c se numesc limite polinomiale.

Limitele asimptotice de tipul 2^{n^δ} se numesc limite exponenţiale.

Limitele asimptotice de tipul k.n se numesc limite lineare.

Limitele asimptotice de tipul $\sqrt{n}$ se numesc limite sublineare.

Observaţia 3

Pe lângă notaţiile **O** şi **Ω** mai există şi notaţiile **o** şi **ω**, obţinute din Definiţia 2 prin înlocuirea inegalităţii ≤ cu inegalitatea strictă <, sau

$$f(n) = o(g(n)) \Leftrightarrow \lim_{n\to\infty} \frac{f(n)}{g(n)} = 0$$

Complexitatea modelelor de calcul

Exemplul 4

$\sqrt{n} = o(n)$

n =***o***(n**.**log log n)

n**.**log log n = ***o***(n**.**log n)

n**.**log n = ***o*** (n^2)

n^2 = ***o*** (n^3)

Complexitatea modelelor de calcul

Propozitia 1

(i) Notaţiile O, Ω, Θ, o, ω sunt tranzitive:

$f(n) = O(g(n))$ & $g(n) = O(h(n))$ $\rightarrow$ $f(n) = O(h(n))$ etc.;

(ii) Notaţiile O, Ω, Θ, sunt reflexive, dar nu şi o, ω

$f(n) = O(f(n))$ dar $f(n) \neq o(f(n))$ etc.;

(iii) Notaţia Θ este simetrică dar nu şi celelalte notaţii:

$f(n) = \Theta(g(n)) \Leftrightarrow g(n) = \Theta(f(n))$.

Complexitatea modelelor de calcul

Propozitia 2

Notaţiile O, Ω etc. pot fi manipulate algebric, dar cu precautie (de ex. egalitatea din formula (5) are loc intr-un singur sens: de la stanga la dreapta):

1. $c.O(f(n)) = O(f(n))$;
2. $O(f(n)) + O(f(m)) = O(f(n))$;
3. $O(O(f(n))) = O(f(n))$;
4. $O(f(n)) \cdot O(g(n)) = O(f(n) \cdot g(n))$;
5. $O(f(n) \cdot g(n)) = f(n) \cdot O(g(n))$.

Complexitatea modelelor de calcul

1. Terminologie
2. Notatia asimptotica
3. Analiza algoritmilor
4. Complexitatea modelelor de calcul

Complexitatea modelelor de calcul

$L_k = \{w \in \Sigma^* \mid \exists\ k \geq 0: w = 0^k 1^k\}$.

Fie $|0^k 1^k| = n$.

Pentru a efectua prima etapă a algoritmului trebuie ***n*** paşi

$\Rightarrow$ ***O***(*n*) paşi.

Aducerea cursorului în extr. stângă pentru a începe bararea: ***n*** paşi

$\Rightarrow$ ***O***(*n*) paşi.

În etapele 2, 3 MT scanează în mod repetat banda pentru a bara simbolurile 0 şi 1.

Fiecare scanare necesită ***n*** paşi iar la fiecare scanare se barează 2 simboluri

$\Rightarrow$ se fac ***n/2*** scanări de câte ***n*** paşi fiecare $\Rightarrow$ ***O***(n^2) paşi.

În etapa 4, MT face o singură scanare pentru a hotărî dacă acceptă / respinge secvenţa de intrare

$\Rightarrow$ se execută maximum ***n*** paşi $\Rightarrow$ ***O***(*n*) paşi.

$\Rightarrow f(n) = \boldsymbol{O}(n) + \boldsymbol{O}(n) + \boldsymbol{O}(n^2) + \boldsymbol{O}(n) = \boldsymbol{O}(n^2)$

Complexitatea modelelor de calcul

Definiţia 4

Fie o functie $t : N \rightarrow R^+$.

Se defineşte clasa de complexitate *timp polinomial* prin:

$TIME(t(n))=\{L\subseteq\Sigma^*|(\exists)$ o MT care decide asupra L în timp $O(t(n))\}$

Exemplul 5

$L_k = \{w\in\Sigma^*|\exists\ k\geq 0: w=0^k1^k\} \in TIME(n^2)$ pentru că:

- M_1 decide asupra L_k în timp $O(n^2)$,
- $TIME(n^2)$ conţine toate limbajele asupra cărora se poate decide în timp $O(n^2)$.

Putem defini o MT care să decidă asupra limbajului L_k mai repede?

Complexitatea modelelor de calcul

(II) Modificăm MT M_1 astfel încât la fiecare etapă 3 barăm câte două simboluri 0 şi două simboluri 1 $\Rightarrow$

reducem numărul de scanări din etapa 3 la jumătate $\Rightarrow$

avem n/4 scanări de câte maximum ***n*** paşi fiecare $\Rightarrow$

$O(n^2) \Rightarrow$

această reducere inseamnă un factor constant egal cu 2

=> nu afectează timpul asimptotic de execuţie al algoritmului;

Complexitatea modelelor de calcul

M_2 = “Fie cuvântul de intrare $w \in \Sigma^*$:

1. Se scaneaza banda de intrare; daca w nu este de forma 0^+1^+ atunci M_2 respinge.
2. Se execută următoarele două etape, atât timp cât pe bandă există atât simboluri 0 cât şi simboluri 1 nebarate:

   3. Se scanează banda pentru a determina dacă numărul total de simboluri 0 şi 1 aflat pe bandă este impar.

   Dacă da, atunci M_2 respinge.

   4. Se scanează banda din nou si se bareaza fiecare al doilea simbol 0 incepand de la primul 0 si apoi fiecarea al doilea simbol 1 incepand de la primul 1.

5. Dacă, după ce toate simbolurile 0 au fost barate pe bandă au mai rămas simboluri 1 nebarate, sau dacă, după ce toate simbolurile 1 au fost barate pe bandă au mai rămas simboluri 0 nebarate, atunci secvenţa w se respinge.

   Altfel, dacă toate simbolurile 0 şi toate simbolurile 1 de pe bandă au fost barate, atunci secvenţa w se acceptă.”

Complexitatea modelelor de calcul

Este evidentă similitudinea dintre masinile M_1 şi M_2;
=> M_2 decide asupra limbajului L_k.
Calculăm acum timpul său de execuţie:

- observăm că fiecare etapă se execută în $O(n)$ paşi;
- numărăm de câte ori se execută fiecare etapă:

  etapa 1: 1! dată,

  etapa 5: 1! dată,

  etapa 4: la fiecare scanare se barează cel puţin 1/2 din numarul de simboluri 0 şi 1/2 din numarul de simboluri 1 $\Rightarrow$

  e nevoie de cel mult $(1+\log_2 n)$ scanări pentru a bara toate simbolurile $\Rightarrow$

  etapele 2,3,4 se execută de cel mult $(1+\log_2 n)$ ori.

$\Rightarrow$ timpul de execuţie pentru M_2 este de:

$$1.O(n) + (1+\log_2 n).O(n) + 1.O(n) = O(n.\log n) + O(n) = O(n.\log n)$$

$\Rightarrow L_k \in TIME\ (n.\log n)$

Complexitatea modelelor de calcul

Putem micsora in continuare timpul de executie?

$o(n.\log n)$=?

<u>Teorema 1</u>

Fie $f : N \rightarrow R^+$ cu proprietatea: $f(n) = o(n.\log n)$

$\Rightarrow TIME(f(n)) = \{L \subseteq \Sigma^* \mid L \in LR\}$

Complexitatea modelelor de calcul

(III) Limbajul L_k poate fi decis chiar în timp linear dacă MT respectivă are două benzi. Fie masina Turing M_3 cu doua benzi definită astfel:

M_3 = “Fie cuvântul de intrare $w \in \Sigma^*$:

1. Se scaneaza banda de intrare; daca exista cel putin un simbol 0 la dreapta unui simbol 1, atunci M_3 respinge.
2. Se scanează subşirul de simboluri 0 de pe banda 1 a MT până la întâlnirea primului simbol 1 şi se copiază pe banda 2 a MT.
3. Se scanează subşirul de simboluri 1 rămas pe banda 1 a MT până la sfârşit, barându-se – pentru fiecare simbol 1 scanat – câte un simbol 0 de pe Banda 2.
4. Dacă toate simbolurile 0 de pe banda 2 au fost barate dar pe banda 1 au mai rămas simboluri 1 de citit sau dacă pe banda 2 au mai rămas simboluri 0 nebarate dar pe banda 1 nu mai sunt simboluri 1 de citit, atunci M_3 respinge.

   Altfel, dacă toate simbolurile 0 de pe Banda 2 au fost barate iar pe banda 1 nu au mai rămas simboluri 1 de citit, atunci M_3 acceptă.”

Complexitatea modelelor de calcul

Evident, M_3 decide asupra limbajului L_k;

Calculam timpul său de execuţie:

fiecare etapă necesita cel mult O(n) pasi si se execută o singură dată

=> timpul de executie este linear si

el nu mai poate fi imbunatatit pt ca citirea secventei de intrare necesita n pasi.

Complexitatea modelelor de calcul

Observaţie importantă

- cu o MT cu o singură bandă, L_k este decidabil în timp *O*(n^2) sau *O*(n.log n);
- nici o MT determinista cu o singură bandă nu poate ameliora performanţa de mai sus;
- cu o MT cu două benzi, L_k este decidabil în timp *O*(n).

  $\Rightarrow$ complexitatea timp a problemei L_k poate fi ameliorata dacă găsim un algoritm mai eficient (o MT mai eficientă)

  $\Rightarrow$ complexitatea problemei L_k depinde de modelul de calculabilitate ales.

Complexitatea modelelor de calcul

- în teoria calculabilităţii:

Teza Church-Turing arată că toate modelele de calculabilitate “valabile” sunt echivalente între ele;

- în teoria complexităţii:

alegerea unui model de calculabilitate afectează complexitatea-timp a limbajului (problemei):

un limbaj care – intr-un model de calculabilitate – este decidabil în timp linear poate să fie – in alt model de calculabilitate – decidabil doar în timp polinomial.

Faptul că acelaşi limbaj poate necesita timpi de calcul diferiţi în diferite modele de calculabilitate pare să torpileze orice incercare de clasificare a problemelor de calculabilitate în funcţie de complexitatea timp a acestora.

Necesarul de timp de calcul nu diferă în mod esenţial la nivelul modelelor de calcul deterministe.

Complexitatea modelelor de calcul

1. Terminologie
2. Notatia asimptotica
3. Analiza algoritmilor
4. Complexitatea modelelor de calcul

Complexitatea modelelor de calcul

Vom analiza trei modele de calculabilitate:

- **MT** cu o singură bandă (clasică);
- **MT** cu mai multe benzi;
- **MT** nedeterministă.

Complexitatea modelelor de calcul

Teorema 2

Fie $t : N \to N$ o functie cu proprietatea că $\forall\, n \in \mathit{N} : t(n) \geq n$.

=> $\forall$ M$\in$MT cu mai multe benzi şi cu timp de lucru t(n)

→ $\exists$ S$\in$MT, cu 1! bandă de intrare şi cu timp de lucru $O(t^2(n))$: L(S)=L(M).

Complexitatea modelelor de calcul

demonstratie

Fie $M \in MT$ cu k benzi de intrare care rulează în timp $t(n)$, $t : \mathcal{N} \to \mathcal{N}$, $t(n) \geq n$, $\forall n \in \mathcal{N}$.

Construim (după metoda utilizată anterior) o MT S cu o singură bandă de intrare care să simuleze M şi analizăm timpul de lucru care necesar.

<u>Iniţial</u>, S depune pe banda sa de intrare conţinuturile celor k benzi al M, separându-le printr-un caracter special, $\#$, $\# \notin \Sigma_S = \Sigma_M$;

în plus, S simulează poziţiile cursoarelor lui M prin bararea simbolurilor corespunzătoare de pe banda sa (înlocuieşte un simbol s cu s', $\forall s \in \Gamma_S = \Gamma_M$).

<u>Apoi</u>, S simulează paşii efectuaţi de M pentru un cuvânt de intrare oarecare, $w \in \Sigma^*$. Pentru a simula un pas efectuat de M, S:

- trebuie să scaneze toată informaţia depusă pe banda sa de intrare ca să poată astfel determina simbolurile aflate "sub" cursoarele lui M;
- apoi, S îşi parcurge din nou banda de intrare pentru a actualiza conţinutul acesteia şi a repoziţiona cursoarele virtuale, conform definiţiei funcţiei de tranziţie a lui M, δ_M.
- dacă vreunul dintre cursoarele reale ale M se deplasează la dreapta pe propria sa bandă de lucru peste o celulă vidă, atunci S trebuie să-şi mărească spaţiul alocat pe unica sa bandă de intrare.

  S face acest lucru deplasând toată informaţia (aflată pe banda sa la dreapta celulei respective) cu o locaţie la dreapta.

Complexitatea modelelor de calcul

Analizăm acum această simulare din punctul de vedere al timpului de calcul necesar.

Pentru fiecare pas executat de M, S parcurge de 2 ori porţiunea activă (cu informaţie) a benzii sale de intrare:

- prima parcurgere serveşte la obţinerea informaţiei necesare pentru a determina mişcarea următoare;
- a doua parcurgere serveşte la efectuarea acestei mişcări.

Timpul necesar efectuării acestor parcurgeri este evident determinat de numărul celulelor cu informaţie de pe banda lui S.

=> căutăm deci o limită superioară pentru această lungime.

Complexitatea modelelor de calcul

Conform constructiei lui S:

marchează extremitateastângă a benzii

numărul de celulecu informaţie de pe prima bandă

delimitatorul celor două benzi alăturate

$$\lambda(S) = 1 + \lambda(M_1) + 1 + \lambda(M_2) + 1 + \ldots + \lambda(M_k) + 1 \leq$$
$$\leq 1 + t(n) + 1 + t(n) + 1 + \ldots + t(n) + 1 \leq 1 + k \cdot t(n) + k$$

Conform ipotezei, M rulează în timp t(n), deci efectuează t(n) paşi utilizând:

exact t(n) celule de pe banda sa, dacă cursorul se deplasează mereu la dreapta;

mai puţin de t(n) celule, dacă există şi deplasări la stânga.

Complexitatea modelelor de calcul

Deci, S scanează porţiunile cu informaţie de pe banda sa de intrare în *O(t(n))* paşi (modulo constantele k şi k + 1, banda lui S este “la fel de lungă” ca oricare dintre benzile lui M). **(*)**

b) Evaluăm acum fiecare pas al lui S:

Pentru a simula oricare dintre paşii lui M , S efectuează 2 scanări ale benzii sale şi maximum k deplasări ale informaţiei sale la dreapta.

Conform **(*)**, fiecare dintre aceşti paşi consumă:

- un timp de calcul de ordinul *O(t(n))* pentru fiecare dintre cele două scanări şi
- un timp de lucru constant pentru cele maximum k deplasări la dreapta (când cursoarele virtuale ajung peste delimitatori).

=> timpul total în care S simulează un pas oarecare l lui M este de ordin *O(t(n)).*

Complexitatea modelelor de calcul

c) Insumăm acum şi găsim o limită superioară a timpului total:

În prima etapă, în care S copiază pe banda sa informaţiile de pe cele k benzi ale M şi le separă prin delimitatorii #, lui S îi sunt necesari $O(n)$ paşi deoarece, conform ipotezei, M rulează – pe o intrare de lungime n – în timp t(n).

Ulterior, S simulează fiecare dintre cei (conform ipotezei) t(n) paşi ai lui M în O(t(n)) paşi

=> această parte a simulării lui M de către S consumă.

$$t(n) \times O(t(n)) = O(t^2(n))$$

$$\Downarrow$$

În total, întreaga simulare consumă

$O(n) + O(t^2(n))$ paşi

Dar, cf. ip.: $t(n) \geq n, (\forall) n \in N$.

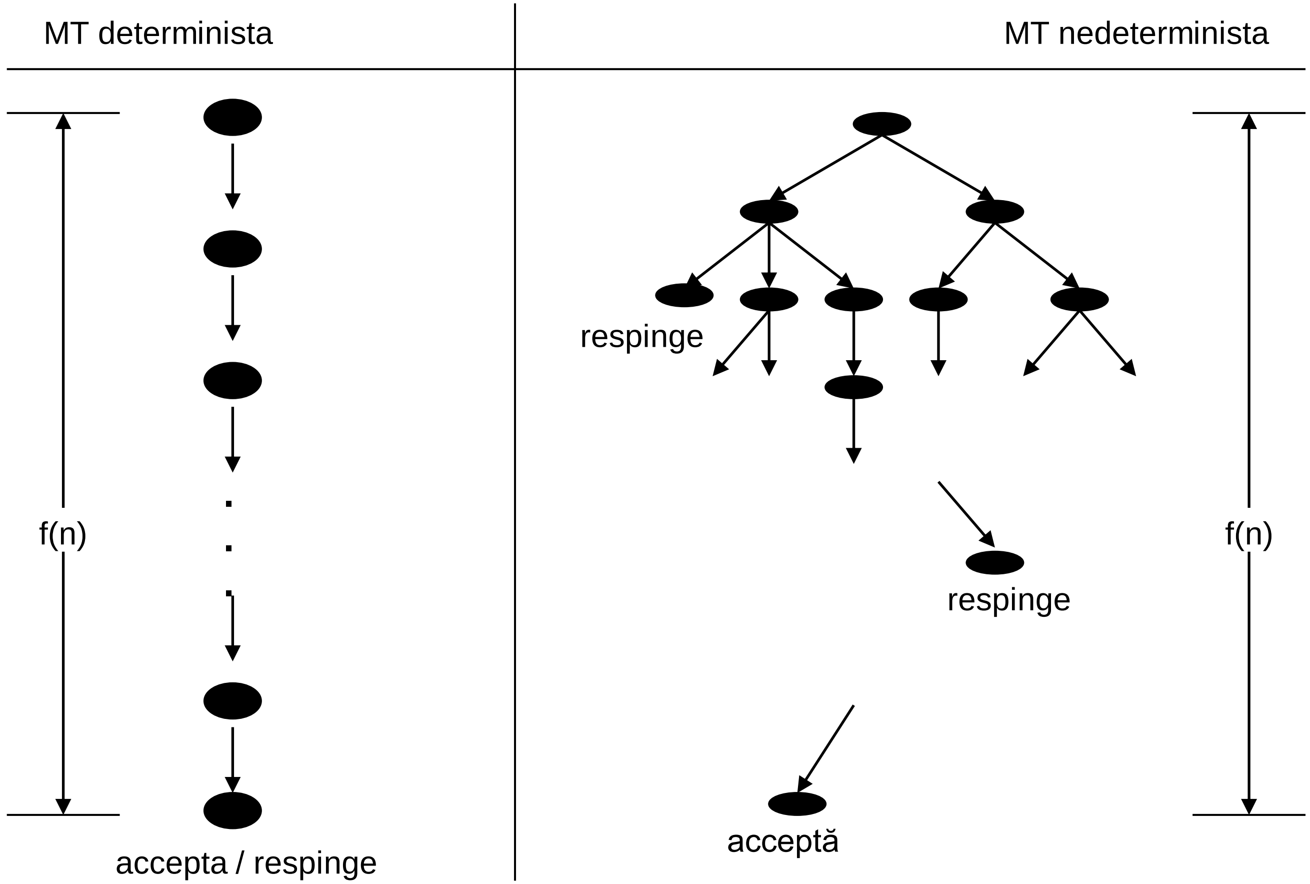
=> putem aprecia timpul total de execuţie al lui S la $O(t^2(n))$.

Complexitatea modelelor de calcul

<u>Observaţia 4</u>

Ipoteza $t(n) \geq n$, $(\forall)\ n \in \mathbf{N}$ nu este restrictivă, ci dimpotrivă: în caz contrar, M "nu ar avea timp" nici măcar să <u>citească</u> toată informaţia, darămite să o prelucreze!

Complexitatea modelelor de calcul



Complexitatea modelelor de calcul

Definitie 5

Fie N o MT nedeterminista cu 1! banda, decidenta,

$w \in L(M) \subseteq \Sigma^*$;

Complexitatea timp a N pe intrarea w este numarul intreg notat $TIME_N(w)$ care reprezinta lungimea celui mai scurt calcul care accepta w, efectuat de N.

Definitie 6

Fie N o MT nedeterminista cu 1! banda, decidenta,

Complexitatea timp $\equiv$ timpul de executie al N este o functie

$f : N \rightarrow N$, definita prin:

f(n) = numărul maxim de paşi executaţi de N pentru un cuvant $w \in L(N) \cap \Sigma^n$, $\forall n \in N$.

Notatia 2

$f(n) = \max \{TIME_N(w) \mid w \in L(N) \cap \Sigma^{n}\} = TIME_N(n)$

Complexitatea modelelor de calcul

<u>Teorema 3</u>

Fie $t : N \rightarrow N$ cu proprietatea că $\forall n \in N$: $t(n) \geq n$.

=> $\forall$ $N \in MT$ nedeterministă cu o singură bandă de intrare şi cu timp de lucru $t(n)$

$\rightarrow$ $\exists$ $D \in MT$ deterministă cu o singură bandă de intrare şi cu timp de lucru $2^{O(t(n))}$: $L(D)=L(N)$.

Complexitatea modelelor de calcul

demonstraţie

Reluăm procedura prin care – în teorema anterioară – am construit MT cu 3 benzi, M_3, pornind de la MT nedeterministă cu 1! bandă, N. Incepem prin a examina arborele de derivare al lui N.

Fie o secvenţă de intrare oarecare $w \in \Sigma^*$, $|w|=n$, $n \in \boldsymbol{N}$. => Îi putem asocia un arbore de derivare în care să figureze TOATE posibilităţile prin care N poate calcula w. Acest arbore are următoarele caracteristici:

(i) lungimea maximă a ramurilor de calcul este, cf.ip., t(n);

(ii) numărul maxim de descendenţi ai unui nod oarecare este b, unde b=nr.max. de alegeri corecte din definiţia funcţiei de tranziţie a lui N, δ_N.

=> (iii) numărul maxim de frunze ale acestui arbore este $b^{t(n)}$ şi

(iv) numărul total de noduri din acest arbore este de ordin $O(b^{t(n)})$.

Complexitatea modelelor de calcul

Examinăm acum modul în care M_3 simulează N.

Simularea presupune explorarea arborelui de derivare;

cf.ip.: N este decidentă

=> N se opreşte pe orice intrare $w \in \Sigma^*$

=> putem folosi oricare dintre metodele de parcurgere a arborilor (inclusv DEPTH-FIRST!)

=> putem aprecia că timpul necesar atingerii unui nod plecând din rădăcină este O(t(n)) (cf. obs. (i));

=> timpul de lucru pentru M_3 este – cf.obs. (iv): $O(t(n).b^{t(n)}) = 2^{O(t(n))}$

=> timpul de lucru necesar convertirii M_3 la D (MT deterministă cu 1! bandă) este – cf. teoreme ant. : $(2^{O(t(n))})^2=2^{O(2.t(n))}= 2^{O(t(n))}$.

Complexitatea modelelor de calcul

1. Terminologie
2. Notatia asimptotica
3. Analiza algoritmilor
4. Complexitatea modelelor de calcul