Clasa NP

1. Clasa NP
2. Exemple de probleme cu timp de calcul polinomial nedeterminist

Clasa NP

Dacă pentru unele probleme cu timp de calcul exponenţial s-au găsit – mai uşor sau mai greu – algoritmi care rulează în timp polinomial, pentru altele astfel de algoritmi nu s-au găsit încă.

Totuşi, cu privire la aceste probleme a fost făcută o constatare remarcabilă:

asemenea rezolvabilităţii algoritmice şi complexitatea unor probleme se află în strânsă relaţie cu complexitatea altora

⇒ descoperirea unui algoritm polinomial pentru o astfel de problemă va permite rezolvarea în timp polinomial a unei clase întregi de probleme.

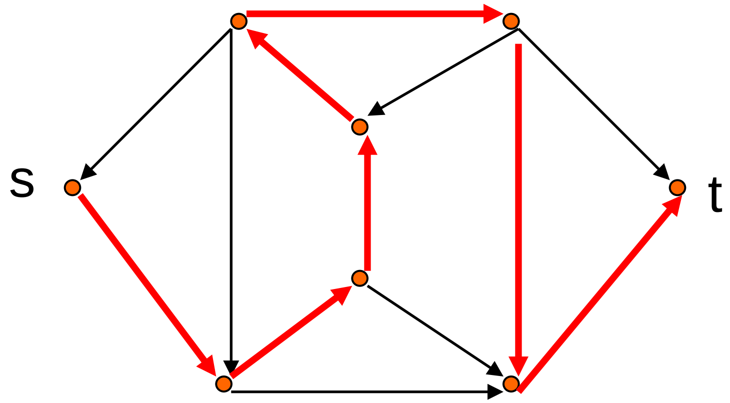
Clasa NP

<u>Exemplul 1:   Problema drumului hamiltonian:</u>

constă în a determina existenţa unui drum hamiltonian (orientat) între două noduri oarecare s şi t ale unui digraf oarecare, G.

Formalizat:

HAMILTPATH = {<G,s,t> | G este un digraf care conţine un drum hamiltonian de la s la t}



Clasa NP

În demonstraţia teoremei "*PATH* $\in$ ***P***" algoritmul de căutare brută (exponenţial) a fost înlocuit cu un alt algoritm, polinomial.

Nu se cunoaşte încă un algoritm polinomial care să rezolve *HAMILTPATH*.

Această problemă are o caracteristică interesantă:

- nu este rezolvabilă în timp polinomial dar
- este verificabilă în timp polinomial.

Dacă determinăm existenţa un drum hamiltonian de la s la t - indiferent cum! –

putem apoi verifica existenţa lui în timp polinomial,

verificând pur şi simplu că fiecare nod apare o dată şi numai o dată:

i.e. un test care se execută în timp $\boldsymbol{O}(m^2)$,

unde m = numărul de noduri din G.

Clasa NP

Exemplul 2: Problema neprimalităţii unui număr:
constă în a determina daca un numar dat este compus sau nu.

Formalizat:

COMPOSITES = { x $\in$ ***N*** | ($\exists$) p, q $\in$ ***N***, p, q > 1 astfel încât x = p**.**q }

nu se cunoaşte un algoritm polinomial care să decidă asupra acestui limbaj

dar verificarea caracterului compus al unui număr natural se poate face în timp polinomial:

e suficient să dispunem de un divizor propriu al acelui număr.

Clasa NP

<u>Observaţia 1</u>

Există şi probleme neverificabile în timp polinomial.

De exemplu, complementul problemei drumului hamiltonian:

$$\overline{HAMILTPATH}$$

Să presupunem că găsim un algoritm care să determine inexistenţa unui drum hamiltonian într-un digraf G;

singura metodă prin care altcineva poate verifica inexistenţa unui astfel de drum constă tot în aplicarea aceluiaşi algoritm exponenţial care a determinat inexistenţa drumului.

Clasa NP

<u>Definiţia 1</u>

Un verificator pentru limbajul L este un algoritm V cu proprietatea:

$$L = \{ w \in \Sigma^* \mid (\exists)\ v \in \Sigma^* \text{ astfel încât } V \text{ acceptă } \langle w,v \rangle \}$$

Măsurăm timpul necesar unui verificator în funcţie de lungimea cuvântului w.

$\Rightarrow$ Un verificator polinomial rulează în timp polinomial in raport cu lungimea cuvântului w.

Un limbaj L este polinomial verificabil dacă admite un verificator polinomial.

<u>Observaţia 2</u>

Cuvântul $v \in \Sigma^*$ din definitie reprezintă informaţia auxiliară utilizată de verificator pentru a verifica apartenenţa cuvântului $w \in \Sigma^*$ la limbajul L.

Acest cuvânt se numeşte certificat sau demonstratie a apartenenţei la L.

Clasa NP

<u>Exemplul 3</u>

HAMILTPATH:

certificatul corespunzător secvenţei de intrare <G,s,t>∈*HAMILTPATH* este însuşi drumul hamiltonian de la s la t.

COMPOSITES : certificatul corespunzător numărului natural x∈*COMPOSITES* este unul dintre divizorii acestuia.

În ambele cazuri, dându-se certificatele corespunzătoare, verificatoarele pot verifica în timp polinomial apartenenţa la limbajul respectiv a secvenţelor de intrare.

Clasa NP

<u>Teorema 1</u>

$\forall$ V$\in$P $\Leftrightarrow$ $\exists$ N$\in$MT nedeterminista cu timp de lucru polinomial a.i. L(V)=L(N).

"$\Rightarrow$"

Fie L un limbaj care admite un verificator V polinomial.

Presupunem că V este o MT care rulează în timp n^k şi construim MT N astfel:

N = "Fie cuvântul de intrare w $\in$ Σ*, |w| = n:

1. Selectăm în mod nedeterminist un cuvânt v de lungime cel mult n^k.
2. Se rulează V pe intrarea <w,v>.
3. Dacă V acceptă <w,v> atunci şi N acceptă w; altfel respinge."

Din construcţia de mai sus rezultă imediat că N decide asupra limbajului L în timp polinomial şi este echivalentă cu V.

Clasa NP

“⇐“

Fie L un limbaj şi N o **MT** nedeterministă care rulează în timp polinomial şi decide asupra limbajului L .

Construim un verificator V polinomial pentru L astfel:

V = “Fie secvenţa de intrare <w,v>, unde w, v $\in \Sigma^*$, oarecari:

1. Se simulează N pe intrarea w; fiecare simbol din v este tratat ca o descriere a alegerii următorului nod din arborele de derivare (alegere care trebuie făcută la fiecare pas; a se vedea demonstraţia teoremei de echivalenta dintre MT deterministe si MT nedeterministe).
2. Dacă această ramură de calcul a lui w din N acceptă atunci şi V acceptă intrarea <w,v>, altfel respinge.”

Din construcţia de mai sus rezultă imediat că verificatorul V este polinomial şi echivalent cu N.

Clasa NP

<u>Definiţia 2</u>

Fie $t : N \rightarrow R^{+}$.

Se defineşte clasa de complexitate *timp polinomial nedeterministic* prin:

NTIME(t(n)) = { $L \subseteq \Sigma^*$ | (∃) o MT nedeterministă cu 1! banda care decide asupra limbajului L în timp *O*(t(n)) }

<u>Definiţia 3</u>

NP este clasa limbajelor care admit verificatoare polinomiale.

NP este clasa limbajelor decidabile în timp polinomial de către MT nedeterministe cu o singură bandă de intrare

$$NP = \bigcup_{k \in N} NTIME\,(n^k)$$

Clasa NP

<u>Observaţia 3</u>

- arătăm că fiecare etapă a algoritmului are o implementare în timp polinomial nedeterminist într-un model de calculabilitate nedeterminist “rezonabil”;
- arătăm că implementarea fiecărei ramuri de calcul necesită un număr de etape de ordin polinomial.

Clasa NP

1. Clasa NP
2. Exemple de probleme cu timp de calcul exponential

Clasa NP

<u>Teorema 2</u>

HAMILTPATH = { <G,s,t> | G este un digraf care conţine un drum hamiltonian de la s la t } $\in$ *NP*.

demonstraţie

Construim o **MT** nedeterminsită NH care să decidă asupra limbajului HAMILTPATH în timp polinomial.

NH = "Fie secvenţa de intrare <G,s,t>, unde G este un digraf oarecare iar s, t sunt oricare două dintre nodurile sale:

1. Se compune o listă de **m** numere naturale $p_1, p_2, \ldots, p_m$, unde m = numărul de noduri ale G. Fiecare număr din listă este ales nedeterminist din mulţimea {1, 2, …, m}.
2. Dacă lista conţine repetiţii, atunci NH respinge.
3. Se verifică dacă $s = p_1$ şi $t = p_m$; dacă oricare dintre condiţii nu are loc, atunci NH respinge.
4. Pentru fiecare $i : 1 \leq i \leq m-1$, se verifică dacă (p_i, p_{i+1}) este un arc din G; dacă cel puţin una dintre condiţii nu are loc atunci NH respinge intrarea <G,s,t>;

   dacă toate condiţiile sunt îndeplinite atunci NH acceptă."

Clasa NP

Analizăm acest algoritm din punct de vedere al complexităţii timp:

etapa 1: generarea nedeterministă a numerelor $p_1, p_2, \ldots, p_m$ dintre numerele 1, 2, … , m

=> timp de lucru polinomial nedeterminist;

etapa 2: găsirea unei repetiţii, cel puţin

=> se fac teste de tipul $p_i=p_j$, unde $i=1,2,\ldots,m-1$ şi $j=i+1,\ldots,m$
=> se execută cel mult m^2 teste;

etapa 3: efectuarea a 2 verificări

=> un factor constant, egal cu 2;

etapa 4: verificarea fiecăreia dintre cele $m.(m-1)$ perechi de numere generate nedeterminist (p_i, p_{i+1}) cu cele maximum $m.(m-1)$ arce din G

=> se execută cel mult m^4 teste.

⇒ algoritmul rulează în timp polinomial nedeterminist.

Clasa NP

<u>Teorema 3</u>

CLIQUE = { <G,n> | G este un graf care conţine o n-clică } $\in$ ***NP.***

demonstratie (1): cu verificator

Luand clica insasi ca certificat, contruim urmatorul verificator K pt CLIQUE:

K = “Fie secventa de intrare <<G,n>,C>:

1. Se testeaza daca C este un set de n noduri din G.
2. Se testeaza daca G contine toate muchiile care leaga nodurile din C.
3. Daca ambele teste sunt trecute, atunci K accepta; altfel, respinge.”

demonstratie (2): cu MT nedeterminista

N = “Fie secventa de intrare <G,n>, unde G=(V,X) este un graf iar $n \in \{3,\ldots,card(V)\}$:

1. Se alege in mod nedeterminist o submultime C de n noduri din G.
2. Se testeaza daca G contine toate muchiile care unesc intre ele cele n noduri din submultimea $C \subseteq V$.
3. Daca da, atunci N accepta; altfel, respinge.”

Clasa NP

SUBSET-SUM = { <S,t> | S = $\{x_1, x_2, \ldots, x_n\}$ şi $(\exists)$ $\{y_1, y_2, \ldots, y_k\} \subseteq \{x_1, x_2, \ldots, x_n\}$ astfel încât $\sum_{i=1}^{k} y_i = t$ }

Exemplu: < {4,11,16,21,27} , 25 > $\in$ SUBSET-SUM: 4 + 21 = 25

<u>Teorema 4</u>

SUBSET-SUM = { <S,t> | S = $\{x_1, x_2, \ldots, x_n\}$ şi $(\exists)$ $\{y_1, y_2, \ldots, y_k\} \subseteq \{x_1, x_2, \ldots, x_n\}$ a.i. $\sum_{i=1}^{k} y_i = t$ } $\in$ ***NP.***

Clasa NP

demonstratie (1): cu verificator

Luand subsetul insasi ca certificat, contruim urmatorul verificator S pt SUBSET-SUM:

S = “Fie secventa de intrare <<S,t>,C>:

1. Se testeaza daca C este o colectie de numere a caror suma este egala cu t.
2. Se testeaza daca S contine toate numerele care fac parte din C.
3. Daca ambele teste sunt trecute, atunci S accepta; altfel, respinge.”

demonstratie (2): cu MT nedeterminista

N = “Fie secventa de intrare <S,t>:

1. Se alege in mod nedeterminist un subset C de numere din S.
2. Se testeaza daca C este un multiset ale carui elemente insumate sunt egale cu t.
3. Daca da, atunci N accepta; altfel, respinge.”

Clasa NP

$\overline{CLIQUE},\ \overline{HAMILTPATH}\ \in\ NP\,?$

Definiţia 4

coNP = $\{L \subseteq \Sigma^* \mid \Sigma^* \setminus L \in NP\}$

Observatia 4

? coNP $\neq$ NP ?

Clasa NP

1. Clasa NP
2. Exemple de probleme cu timp de calcul exponential