Curs 12

January 11, 2022

Timp si spatiu polinomial

$$\begin{aligned}
\mathbf{P} &= \bigcup_{i\geq 1} DTIME(n^i) \\
\mathbf{NP} &= \bigcup_{i\geq 1} NTIME(n^i) \\
\mathbf{PSPACE} &= \bigcup_{i\geq 1} DSPACE(n^i) \\
\mathbf{NSPACE} &= \bigcup_{i\geq 1} NSPACE(n^i) \\
\mathbf{L} &= DSPACE(\log n).
\end{aligned}$$

$$\mathbf{L} \subseteq \mathbf{P} \subseteq \mathbf{NP} \subseteq \mathbf{NSPACE} = \mathbf{PSPACE}.$$

Cel putin o incluziune este stricta: CARE?

Reduceri

Definitie. Masina Turing folosita in reduceri (similar unui translator): o masina determinista M care se opreste pe fiecare intrare a.i. pentru un input w produce un sir $f_M(w)$.

Un limbaj L' este *Turing-reductibil* la L daca exista o mT M a.i. $w \in L'$ ddaca $f_M(w) \in L$.

Definitie. L' este reductibil in timp polinomial la L ($L' \leq_{tp} L$ daca exista o mT M cu timp de lucru polinomial care reduce L' la L.

Translator off-line: o mT M care se opreste pe fiecare intrare, are o banda de intrare care se poate doar citi, o banda auxiliara, si o banda de iesire pe care se poate doar scrie fara a se putea intoarce.

Definitie. L' este reductibil in spatiu logaritmic la L ($L' \leq_{sl} L$ daca exista un translator off-line M cu spatiu de lucru logaritmic care reduce L' la L.

Proprietati ale reducerilor

Teorema. *Fie $L' \leq_{tp} L$. Daca L este in* **(N)P** *atunci L' este in* **(N)P**.
Dem. Demonstram pentru $L \in \mathbf{P}$. Fie $p_1(n)$ (polinom) timpul in care M reduce L' la L. Atunci pentru fiecare $|x| = n$, rezulta $|f_M(x)| \leq p_1(n)$. Pentru ca $L \in \mathbf{P}$, $f_M(x) \in L$ se poate face in timp $p_2(|f_M(x)|) \leq p_2(p_1(n))$. Asadar, $x \in L'$ se poate decide in timp $p_1(n) + p_2(p_1(n))$.

Teorema. *Fie $L' \leq_{sl} L$.*
1. Daca L este in **P** *atunci L' este in* **P**.
2. Daca $L \in (N)(D)SPACE(\log^k n)$ atunci $L' \in (N)(D)SPACE(\log^k n)$.
Dem. 1. Imediat, pentru ca orice reducere in spatiu logaritmic se poate face in timp polinomial.
Fie $L \in DSPACE(\log^k n)$. Fie M_1 translatorul care reduce L' la L is M_2 mT care accepta L in spatiu marginit de $\log^k n$. Lungimea iesirii lui M_1 pe intrarea $|x| = n$ este cel mult $s(n+2)t^{\log^k n}$, unde s este numarul de stari si t numarul de simboluri ale lui M_1. Exista o constanta c a.i. $s(n+2)t^{\log^k n} \leq (2^c)^{\log^k n}$.

Proprietati ale reducerilor

CONTINUARE DEM. Construim M_3 astfel:

- Pe banda de intrare are x, $|x| = n$.
- Are o banda auxiliara pe care simuleaza banda lui M_1.
- Are o banda auxiliara pe care simuleaza banda lui M_2.
- Are o banda auxiliara pe care va pastra pozitia i a capului de citire al lui M_2 pe banda sa de intrare, numar scris in baza 2^c. Deci spatiul folosit pe aceasta banda va fi cel mult $\log^k n$.
- Pentru fiecare astfel de i, M_3 restarteaza simularea lui M_1 pe intrarea x si numara simbolurile scrise la iesire de M_1 pana ajunge la i.
- Al i-lea simbol produs la iesire de M_1 este dat lui M_2 ca simbol curent pe banda sa de intrare.
- Simuleaza miscarea lui M_2 pentru acest simbol si actualizeaza i la $i - 1$ sau $i + 1$, si reia procesul.
- Cazuri particulare: (1) $i = 0$ inseamna ca M_2 citeste marginea stanga a benzii sale de intrare.
  (2) Simularea lui M_1 se opreste fara a produce al i-lea simbol, inseamna ca M_2 citeste marginea dreapta a benzii sale de intrare.

Probleme dificile si complete

Teorema. *Compunerea a doua reduceri de acelasi tip este o reducere de acelasi tip.*

Fie $\mathcal{L}$ o clasa de limbaje.

Def. 1. Un limbaj L este *dificil* pentru $\mathcal{L}$ ($\mathcal{L}$-dificil) in raport cu reducerea $\leq_{tp}$ sau $\leq_{sl}$ daca pentru orice limbaj $L' \in \mathcal{L}$ avem $L' \leq_{tp} L$ sau $L' \leq_{sl} L$.

2. Un limbaj L este *complet* pentru $\mathcal{L}$ ($\mathcal{L}$-complet) in raport cu reducerea $\leq_{tp}$ sau $\leq_{sl}$ daca $L \in \mathcal{L}$ si L este $\mathcal{L}$-dificil in raport cu reducerea $\leq_{tp}$ sau $\leq_{sl}$.

3. Un limbaj este **NP**-complet daca este **NP**-complet in raport cu reducerea $\leq_{tp}$ sau $\leq_{sl}$.

Probleme NP-complete

Teorema. (Cook-Levin) *Problema satisfiabilitatii (SAT) este* **NP**-*completa.*

SAT: Input: o formula in forma normala conjunctiva din calculul propozitional

$$\alpha = C_1 \wedge C_2 \wedge \cdots \wedge C_m, \text{ unde}$$
$$C_i = (x_{i_1}^{e_1} \vee x_{i_2}^{e_2} \vee \ldots x_{i_{k_i}}^{e_{k_i}}),$$
$$x^1 = x \text{ si } x^0 = \bar{x}.$$

Output: YES/NO daca exista/nu exista o asignare a variabilelor care o satisfac.

Limbajul L_{SAT} se defineste astfel: $L_{SAT} = \{< \alpha >| \ \alpha \text{ este satisfiabila } \}$.

Daca $\alpha = C_1 \wedge C_2 \wedge \cdots \wedge C_m$, atunci

$< \alpha >=< C_1 > \wedge < C_2 > \wedge \cdots \wedge < C_m >$.

Daca $C_i = (x_{i_1}^{e_1} \vee x_{i_2}^{e_2} \vee \ldots x_{i_{k_i}}^{e_{k_i}})$, atunci

$< C_i >= (< x_{i_1}^{e_1} > \vee < x_{i_2}^{e_2} > \vee \cdots < x_{i_{k_i}}^{e_{k_i}} >)$.

$< x_i >= x i_{(2)}, < \bar{x}_i >= \bar{x} i_{(2)}$.

Probleme **NP**-complete

Exemplu. $\alpha = (x_1 \vee x_2) \wedge (x_1 \vee \bar{x}_3)$
$< \alpha >= (x1 \vee x10) \wedge (x1 \vee \bar{x}11)$.
Obs. Daca $|\alpha| = n$, atunci $| < \alpha > | \leq n \log n$. Deoarece vom lucra cu reduceri in spatiu logaritmic, vom considera ca $| < \alpha > | = n$, pentru ca $\log(n \log n) \leq 2 \log n$.
$L_{SAT} \in$ **NP**: De ce? Se poate verifica in timp polinomial daca o asignare satisface formula. Cum?

Probleme **NP**-complete

Exemplu. $\alpha = (x_1 \vee x_2) \wedge (x_1 \vee \bar{x}_3)$
$< \alpha >= (x1 \vee x10) \wedge (x1 \vee \bar{x}11)$.
Obs. Daca $|\alpha| = n$, atunci $| < \alpha > | \leq n \log n$. Deoarece vom lucra cu reduceri in spatiu logaritmic, vom considera ca $| < \alpha > | = n$, pentru ca $\log(n \log n) \leq 2 \log n$.
$L_{SAT} \in$ **NP**: De ce? Se poate verifica in timp polinomial daca o asignare satisface formula. Cum?

1. Se determina variabilele care apar in formula. (determinist)
2. Se genereaza nedeterminist o asignare. (Nedeterminist)
3. Se verifica daca asignarea satisface formula. (determinist)
4. Complexitate?

Probleme **NP**-complete

Exemplu. $\alpha = (x_1 \vee x_2) \wedge (x_1 \vee \bar{x}_3)$
$< \alpha >= (x1 \vee x10) \wedge (x1 \vee \bar{x}11)$.
Obs. Daca $|\alpha| = n$, atunci $| < \alpha > | \leq n \log n$. Deoarece vom lucra cu reduceri in spatiu logaritmic, vom considera ca $| < \alpha > | = n$, pentru ca $\log(n \log n) \leq 2 \log n$.
$L_{SAT} \in$ **NP**: De ce? Se poate verifica in timp polinomial daca o asignare satisface formula. Cum?

1. Se determina variabilele care apar in formula. (determinist)
2. Se genereaza nedeterminist o asignare. (Nedeterminist)
3. Se verifica daca asignarea satisface formula. (determinist)
4. Complexitate?$O(n^2)$.

Probleme **NP**-complete

L_{SAT} este **NP**-dificil (schita).

Fie M o mT nedetrminista oarecare care decide L in timp polinomial $p(n)$.

Construim o formula α a.i. $w \in L(M)$ ddaca α este satisfiabila.

Variabile:

- $T_{k,i,j}$: la pasul k, celula i contine simbolul j; $(p^2(n))$
- $RW_{k,i}$: la pasul k, capul R/W citeste celula i; $(p^2(n))$
- $Q_{k,s}$: la pasul k, starea curenta este s. $(p(n))$

Formula:

$$\alpha = U \wedge I \wedge D \wedge N \wedge E,$$

- U: la orice pas, M se afla intr-o singura stare, capul R/W acceseaza o singura celula, si fiecare celula are o singura litera;
- I: confi

  guratia initiala a masinii, deci pasul $k = 0$;
- D: tranzitiile posibile, conform functiei de tranzitie;
- N: toate celulele neprocesate pastreaza litera de la un pas la altul;
- E: conditia de acceptare.

Probleme **NP**-complete

Teorema. *Problema* $3 - SAT$ *este* **NP**-*completa*.
Dem. Fie o clauza $C = l_1 \vee \cdots \vee l_k$, $k \geq 1$.

- Cazul 1. $k > 3$. Inlocuim C cu
  $C' =$
  $(l_1 \vee l_2 \vee y_1) \wedge (l_3 \vee \bar{y}_1 \vee y_2) \wedge \cdots \wedge (l_{k-2} \vee \bar{y}_{k-4} \vee y_{k-3}) \wedge (l_{k-1} \vee l_k \vee \bar{y}_{k-3})$.
  C este satisfiabila ddaca C' este satisfiabila. De ce?
- Cazul 2. $k = 2$. Inlocuim C cu $C' = (l_1 \vee l_2 \vee y) \wedge (l_1 \vee l_2 \vee \bar{y})$.
- Cazul 3. $k = 1$. Inlocuim C cu
  $C' = (l_1 \vee y_1 \vee y_2) \wedge (l_1 \vee \bar{y}_1 \vee y_2) \wedge (l_1 \vee y_1 \vee \bar{y}_2) \wedge (l_1 \vee \bar{y}_1 \vee \bar{y}_2)$.

Transformarile se pot face in spatiu logaritmic.
Problema. Care este statutul problemei 2-SAT?

Probleme **NP**-complete

Teorema. *Problema VERTEX COVER este* **NP**-*completa.*
VERTEX COVER: Un graf neorientat $G = (V, E)$; $|V| = n$, $|E| = m$, o acoperire a lui G este o submultime $X \in V$ a.i. $\{u, v\} \cap X \neq \emptyset$, pentru orice $\{x, y\} \in E$.
Problema: Pentru un graf G si $k \geq 1$, exista o acoperire a lui G de cardinal maxim k?
Codificarea unei intrari:

Probleme **NP**-complete

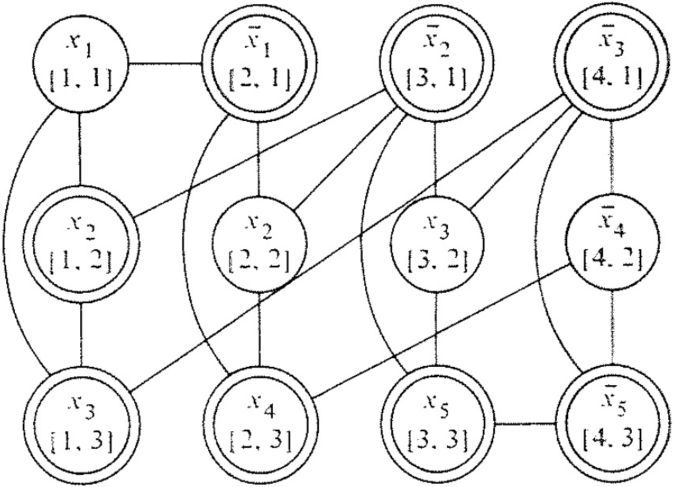
Teorema. *Problema VERTEX COVER este **NP**-completa.*
VERTEX COVER: Un graf neorientat $G = (V, E)$; $|V| = n$, $|E| = m$, o acoperire a lui G este o submultime $X \in V$ a.i. $\{u, v\} \cap X \neq \emptyset$, pentru orice $\{x, y\} \in E$.
Problema: Pentru un graf G si $k \geq 1$, exista o acoperire a lui G de cardinal maxim k?
Codificarea unei intrari:
$k_{(2)}\#v1_{(2)}\#v2_{(2)}\#\ldots\#vn_{(2)}\#(vi1_{(2)}, vj1_{(2)})\#\ldots\#(vim_{(2)}, vjm_{(2)})$.
L_{VC} contine multimea tuturor codificarilor pentru care raspunsul este DA.
L_{VC} este in **NP**? De ce ?

Probleme **NP**-complete

L_{VC} este **NP**-dificil? Reducem 3-SAT la VERTEX COVER.

Ex. $\alpha = (x_1 \vee x_2 \vee x_3) \wedge (\bar{x}_1 \vee x_2 \vee x_4) \wedge (\bar{x}_2 \vee x_3 \vee x_5) \wedge (\bar{x}_3 \vee \bar{x}_4 \vee \bar{x}_5)$.

$k = 8$



Probleme **NP**-complete

Problema clicii este NP-completa

Problema clicii: Pentru un graf G si $k \geq 1$, exista un subgraf complet al lui G (clica) de cardinal minim k?

Problema VERTEX COVER se poate reduce in spatiu logaritmic/timp polinomial la problema clicii. Cum?

Probleme **NP**-complete

Problema clicii este NP-completa

Problema clicii: Pentru un graf G si $k \geq 1$, exista un subgraf complet al lui G (clica) de cardinal minim k?

Problema VERTEX COVER se poate reduce in spatiu logaritmic/timp polinomial la problema clicii. Cum? Se construieste graful complementar al grafului dat.

Alte probleme **NP**-complete:

1. SUBSET SUM: Se dau numerele naturale $a_1, a_2, \ldots, a_n$ si b. Exista o submultime de suma b?
2. PARTITION: Se dau numerele naturale $a_1, a_2, \ldots, a_n$. Se pot partitiona in doua submultimi avand aceeasi suma?
3. BIN PACKING: Se dau numerele naturale $a_1, a_2, \ldots, a_n$, toate mai mici decat b si $k \geq 2$. Se pot partitiona in cel mult k submultimi fiecare avand suma cel mult b?
4. Problema drumului (circuitului) hamiltonian.
5. Problema 3-colorarii unui graf (planar).
6. SET COVER.

Echivalenta expresiilor regulate

Problema 1. Pentru doua expresii regulate date R_1, R_1, este $L(R_1) = L(R_2)$?

- Este problema in **NP**?

Echivalenta expresiilor regulate

Problema 1. Pentru doua expresii regulate date R_1, R_1, este $L(R_1) = L(R_2)$?

- Este problema in **NP**? Nu se stie.
- Schimbam problema in complementara sa: Sunt doua expresii date non-echivalente? Este in **NP**?

Echivalenta expresiilor regulate

Problema 1. Pentru doua expresii regulate date R_1, R_1, este $L(R_1) = L(R_2)$?

- Este problema in **NP**? Nu se stie.
- Schimbam problema in complementara sa: Sunt doua expresii date non-echivalente? Este in **NP**? Nu se stie. ($w \in L(R_1) \setminus L(R_2)$ sau $w \in L(R_2) \setminus L(R_1)$.)
- Particularizam problema: Expresii regulate *-libere (expresii regulate fara operatia de inchidere Kleene). NEER: Este problema non-echivalentei a doua expresii regulate *-libere in **NP**?

Echivalenta expresiilor regulate

Problema 1. Pentru doua expresii regulate date R_1, R_1, este $L(R_1) = L(R_2)$?

- Este problema in **NP**? Nu se stie.
- Schimbam problema in complementara sa: Sunt doua expresii date non-echivalente? Este in **NP**? Nu se stie. ($w \in L(R_1) \setminus L(R_2)$ sau $w \in L(R_2) \setminus L(R_1)$.)
- Particularizam problema: Expresii regulate *-libere (expresii regulate fara operatia de inchidere Kleene). NEER: Este problema non-echivalentei a doua expresii regulate *-libere in **NP**? Da: $R = ((a+b)aa(a+b) + aba(a+b)b)$.
- $SAT \leq_{tp} NEER$.

$SAT \leq_{tp} NEER$

Fie $\alpha = C_1 \wedge C_2 \wedge \cdots \wedge C_p$ cu variabilele $x_1, x_2, \ldots, x_n$; construim doua expresii regulate *-libere peste $\{0, 1\}$:

$$R_1 = (0+1)^n$$

$$R_2 = Z_1 + Z_2 + \cdots + Z_p,$$

$$Z_i = Z_i^1 \cdot Z_i^2 \cdot \cdots \cdot Z_i^n \text{ si } Z_i^j = \begin{cases} 0, & \text{daca } x_j \text{ apare in } C_i \\ 1, & \text{daca } \bar{x}_j \text{ apare in } C_i \\ (0+1), & \text{altfel} \end{cases}$$

Z_i descrie asignarile care invalideaza C_i.

Probleme complete pentru alte clase

- Problema satisfiabilitatii unei formule din calculul cu predicate (logica de ordinul intai) fara variabile libere este **PSPACE**-completa.
- Problema apartenentei pentru o gramatica dependenta de context (monotona) este **PSPACE**-completa. (Surprinzator pentru ca este in $NSPACE(n)$).
- Problema trivialitatii limbajului generat de o gramatica independenta de context este **P**-completa.
- Problema reachabilitatii este **NL**-completa.

$$\mathbf{EXP} = \bigcup_{i \geq 1} DTIME(2^{n^i})$$

$$\mathbf{NEXP} = \bigcup_{i \geq 1} NTIME(2^{n^i}).$$

$$\mathbf{L} \subseteq \mathbf{P} \subseteq \mathbf{NP} \subseteq \mathbf{NSPACE} = \mathbf{PSPACE} \subseteq \mathbf{EXP} \subseteq \mathbf{NEXP}.$$

Teorema Daca $\mathbf{P} = \mathbf{NP}$ atunci $\mathbf{EXP} = \mathbf{NEXP}$.

P, NP, co-NP, PSPACE

O problema este in **co-NP** daca complementara sa este in **NP**. Definitie cu masini Turing?

P, NP, co-NP, PSPACE

O problema este in **co-NP** daca complementara sa este in **NP**. Definitie cu masini Turing?

Exemplu de problema: TAUT=este tautologie o formula data?

Teorema. TAUT este **co-NP**-completa.

Dem. 1. TAUT este in **co-NP**. De ce?

P, NP, co-NP, PSPACE

O problema este in **co-NP** daca complementara sa este in **NP**. Definitie cu masini Turing?

Exemplu de problema: TAUT=este tautologie o formula data?

Teorema. TAUT este **co-NP**-completa.

Dem. 1. TAUT este in **co-NP**. De ce?

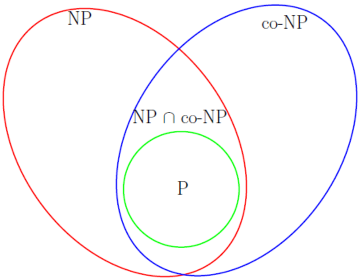
Pentru ca $\overline{\mathrm{TAUT}}$ este in **NP**.

2. TAUT este **co-NP** dificila. De ce?

Orice problema **NP**-completa are complementara **co-NP**-completa. Fie L **NP**-complet, atunci $\bar{L} \in \mathbf{co} - \mathbf{NP}$. Fie $L' \in \mathbf{co} - \mathbf{NP}$, deci $\bar{L}' \in \mathbf{NP}$; prin urmare $\bar{L}' \leq_{tp} L$. Rezulta ca $L' \leq_{tp} \bar{L}$.

Dar $SAT \leq_{tp} \overline{TAUT}$.

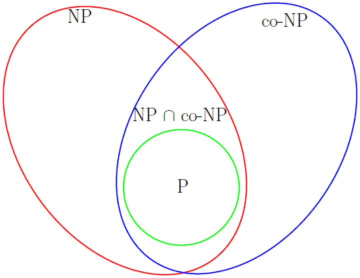
P, NP, co-NP, PSPACE



Relatii.

1. **P**=**co-P**.
2. **P** $\subseteq$ **NP** $\cap$ **co** $-$ **NP**. Sunt egale? (Conjectura NU). Extrem de important in criptografie.
3. **NP**=**co-NP**? (Conjectura: NU) Ar fi egale ddaca complementul unei probleme **NP**-complete are fi in **NP**. De ce?

P, NP, co-NP, PSPACE



Relatii.

1. **P**=**co-P**.
2. **P** $\subseteq$ **NP** $\cap$ **co** $-$ **NP**. Sunt egale? (Conjectura NU). Extrem de important in criptografie.
3. **NP**=**co-NP**? (Conjectura: NU) Ar fi egale ddaca complementul unei probleme **NP**-complete are fi in **NP**. De ce? Daca $L' \leq_{sl} L$ atunci $\overline{L}' \leq_{sl} \overline{L}$.