# Reduceri polinomiale

4 ianuarie 2018

**Reducerea polinomială**

A = algoritm pentru problema PA (pe care o reduc)

B = algoritm pentru problema PB (la care reduc)

F = algoritm **determinist polinomial** de transformare a intrărilor pentru A în intrări pentru B

Dacă există F algoritm determinist cu complexitate polinomială care transformă **orice** instanță i a problemei PA într-**o** instanță F(i) a problemei PB a.î. **A(i)=1 <=>B(F(i))=1**, atunci spunem că PA se reduce polinomial la PB: **PA** $\leq_p$ **PB**.

**Obs0**: diferența dintre reducerea polinomială și reducerea Turing este că aici F trebuie să fie determinist polinomial.

**Obs1**: faptul ca reducerea nu merge (neapărat) și de la PB la PA se vede în sublinierea cuvintelor "orice" și "o". Când reducem PA la PB luăm o intrare oarecare pentru PA și construim în mod convenabil o intrare pentru PB.

**Obs2**: demonstrația faptului ca o problemă se reduce polinomial la alta nu e completă dacă nu demonstrați **implicația în ambele sensuri**! Întâi trebuie demonstrat că A(i)=1 =>B(F(i))=1, apoi ca B(F(i))=1 =>A(i)=1.

Consecințe ale stabilirii unei relații de reducere polinomială:

$A \leq_p B \land B \in P \Rightarrow A \in P$

$A \leq_p B \land A \notin P \Rightarrow B \notin P$

Aceste consecințe se folosesc pentru a demonstra apartenența unor probleme la o anumită clasă de probleme, folosind reduceri (de) la probleme a căror (in)tractabilitate este cunoscută.

**Clase de probleme**

**P** = { A | ∃ B un algoritm determinist polinomial care rezolvă A }

**NP** = { A | ∃ B un algoritm nedeterminist polinomial care rezolvă A }

**NP-hard** = { A | ∀ Q ∈ NP • Q $\leq_p$ A }

**NP-complete** = { A | A ∈ NP ∧ A∈ NP-hard }

Strategii de a demonstra apartenența la o clasă de probleme

**P**: se construiește un algoritm determinist polinomial

**NP**: se construiește un algoritm nedeterminist polinomial SAU se arată ca soluția poate fi verificată în timp polinomial

**NP-hard**: se găsește o problema cunoscută ca NP-hard și se reduce această problemă la problema curentă (nu invers!)

**NP-complete**: se arată că e NP si NP-hard

**Alte clase de probleme**

**PSPACE** = clasa problemelor care sunt rezolvate de algoritmi deterministi in timp nelimitat si utilizând spațiu polinomial

**NPSPACE** = clasa problemelor care sunt rezolvate de algoritmi NEdeterministi în timp nelimitat șî utilizând spațiu polinomial

**PSPACE** = NPSPACE (Savitch, 1970)

*Teorema 1.* Relația de reducere polinomială $\leq_p$ este tranzitivă.

*Teorema 2.* O problemă Q2 este NP-hard dacă și numai dacă există o problema NP-completă Q1 astfel încat Q1 $\leq_p$ Q2.

*Teorema 3.* Problemele NP-complete formează o clasă de echivalență în raport cu relația $\leq_p$ (altfel spus, dacă Q1, Q2 sunt NP-complete atunci Q1 $\leq_p$ Q2 și Q2 $\leq_p$ Q1).

*Teorema 4.* Dacă există o problemă Q $\in$ NP-completă și Q $\in$ P, atunci P = NP.

*Teorema 5.* Dacă există o problemă Q$\in$ NP-hard și Q $\in$ P, atunci P = NP.

Demonstrați reducerile polinomiale următoare:

1. 3-Colorare $\leq_p$ 4-Colorare

   Problema **k-Colorare** (k$\geq$2): Dându-se un graf neorientat G = (V, E) și k culori, se pot colora nodurile grafului folosind cele k culori astfel încât niciun nod sa nu aibă un vecin de aceeași culoare?

2. Partitie $\leq_p$ q-Sume

   **Problema Partiție**: Dându-se un număr n și o mulțime de n întregi A = $\{a_1, ..., a_n\}$, există S $\subset$ {1, ..., n} astfel încât $\sum_{i \in S} a_i = \sum_{i \notin S} a_i$?

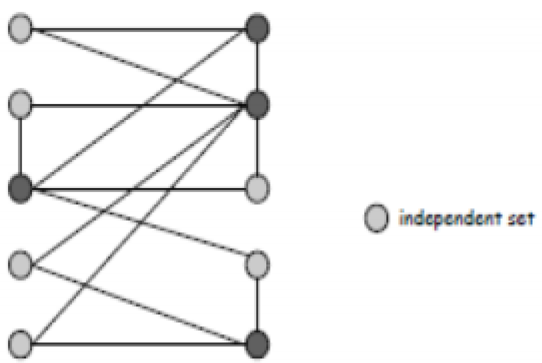
   **Problema q-Sume(Subset Sum)**: Dându-se numerele t, q și o mulțime de t întregi B = $\{b_1, ..., b_t\}$, există $S^{'} \subseteq$ {1, ..., t} astfel încât $\sum_{i \in S^{'}} b_i$ = q?

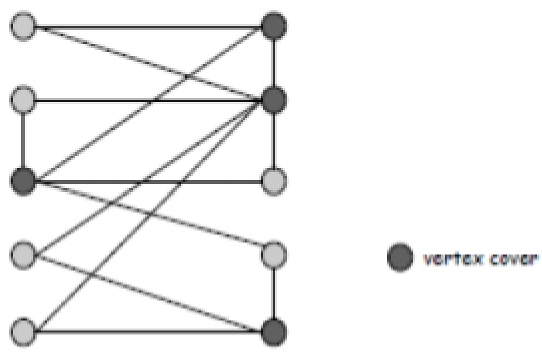
3. Independent Set $\leq_p$ Vertex Cover

   **Problema Vertex Cover**: Dându-se un graf neorientat G = (V, E) și un număr k, există o submulțime S de noduri, |S| = k, astfel încât fiecare muchie are <u>cel puțin</u> un capăt în S?

   **Problema Independent Set**: Dându-se un graf neorientat $G^{'}$ = ( $V^{'}$, $E^{'}$) și un număr $k^{'}$, există o submulțime $S^{'}$ de noduri, $|S^{'}| = k^{'}$, astfel încât fiecare muchie are <u>cel mult</u> un capăt în $S^{'}$?

   Exemplu:



4. Vertex Cover $\leq_p$ Set Cover

   **Problema Set Cover**: Dându-se o mulțime U, o colecție $S_1$, $S_2$, ..., $S_m$ de submulțimi ale lui U și un număr $k^{'}$, există o colecție de $k^{'}$ astfel de submulțimi care reunite să dea U?