Examen CC

Problema 3

SAT face parte din clasa de probleme care pot fi reduse la un număr polinomial de apeluri ale unui oracol de fezabilitate (sau satisfiabilitate). Cel mai clasic exemplu ~~este~~ al unei astfel de probleme este găsirea unui ciclu hamiltonian:

- scoatem câte o muchie, și întrebăm oracolul dacă mai există un ciclu hamiltonian. Dacă NU, punem muchia înapoi.
- După ce aplicăm pașii de mai sus, care necesită $O(M)$ apeluri ale oracolului, rămânem cu un ciclu hamiltonian.

Vom aplica un algoritm similar.

Fie o MT ORACOL, care primește ca

input o instanță SAT, și întoarce DA / NU.

Presupunem că inputul este codificat în felul următor:

$$(X_{a_1} \vee \overline{X}_{a_2} \vee \dots \vee X_{a_n}) \wedge (\overline{X}_{b_1} \dots \vee X_{b_m}) \wedge \dots$$

$$\Downarrow$$

$$\#1X\,bin(a_1)\,\$0X\,bin(a_2) \dots \#0X\,bin(b_1) \dots$$

altfel spus, avem:

- "#" între conjuncții
- "$" între disjuncții
- Fiecare variabilă este codificată
  - 0/1 dacă este negată sau nu
  - X
  - numărul variabilei în binar.

Obs putem în timp liniar și spațiu logaritmic să "traducem" formatul prezentat în curs la acesta.

Pași folosiți de mașina noastră T:

Pas 1 - Își scrie pe a doua bandă Id-ul variabilei cele mai mari care să existe. Fie acesta N.

Pas 2 - Pentru fiecare ID de la 1 la N (putem să ne incrementăm în O(4) ID-ul pe a 3-a bandă, încearcă:

- să adauge "#0×ID" la sfârșitul benzii.
- simulează mașina ORACOL pe ~~toată banda~~ noul input
- Dacă ORACOL răspunde DA, înseamnă că există o asignare validă cu $X_{ID} = 0$, deci adăugăm un 0 pe a 4-a bandă care fixe mintea răspunsul.
- Dacă ORACOL răspunde NU, înseamnă că nu pot exista soluții decât cu $X_{ID} = 1$, deci punem 1 pe a 4-a bandă, și înlocuim "#0×ID" cu "#1×ID".

Obs prin ID se înțelege scrierea acestuia în binar.

Pas 3: Verificăm dacă avem o soluție validă prin apelarea oracolului pe inputul modificat.

- Dacă DA
  - ↳ soluția este pe bandă 4
- Dacă NU
  - ↳ Nu avem soluție, și refuzăm inputul

Complexitate:

- Pasul 1 costă $O(|input|)$, pentru a găsi ID-ul maxim. Presupunem că inputul este corect, și apar toate variabilele de la 1 la N (ID-ul maxim). $N \leq |input|$
- Pasul 2 face $O(\log(N))$ operații pe bandă și simulează o dată ORACOL pentru fiecare ID de la 1 la N, deci face în total $O(TIME_{ORACOL}(2 \cdot input) \cdot N + N \log N)$

Cum $O(|input|) \geq O(N \log N)$, pas 2 face $O(TIME_{ORACOL}(2 \cdot input))$

Am spus 2· input pentru că adăugăm elemente pe bandă

dar nu mai mult de dublul acestora.

- Pasul 3 simulează un apel de ORACOL.

Cum simularea ORACOL pe un input de dimensiune M are cel puțin O(M), avem complexitatea totală de

$O(ORACOL \cdot N)$, unde:

- ORACOL este timpul dacă simulă oracol pe o instanță de cel mult 2. input
- N este ID-ul maxim ($N \leq |input|$).

Așadar, avem o complexitate polinomială, presupunând că ORACOL este polinomial.

Corectitudine:

- Dacă nu există soluții cu $X_K = 0$, orice soluție validă va avea $X_K = 1$, deci putem să presupunem acest lucru, și să adaugăm (X_K) în clauzele SAT.