

Analiza algoritmilor

Tema 4: Polynomial time reduction from kVertexCover to SAT

Student: Gheorghe Adrian – Valentin

Grupa: 321CB

Username HR: adrian_gheorghe

Deducerea formulei de transformare a KvertexCover în SAT

Pentru formula am analizat inputul și outputul dat ca exemplu în enuntul temei. Am descoperit ca fiecare nod are K codificari, corespunzatoare celor K alegeri de noduri care se fac, conform enuntului problemei KvertexCover, conform formulei:

$$x = n * K - i$$

cu n fiind nodul curent, K fiind primul numar dat ca input și i un indice care ia valori de la 0 la K – 1, semnificand cele K alegeri. Astfel, am ales sa împart output ul în 3 părți:

Partea 1:

$$(7V3V8V4) \wedge (7V1V8V2) \wedge (3V5V4V6) \wedge (7V9V8V10) \wedge (3V1V4V2) \wedge (5V7V6V8) \wedge (5V9V6V10)$$

Aici, fiecare clauza corespunde unei muchii date ca input. Avem, în total, $n \times k$ variabile, cu n fiind numărul de noduri dat ca input și k este primul numar din fișierul de intrare. Astfel, pentru prima muchie din input, (4,2), notam :

p = nodul de „plecare” al muchiei (în cazul de fata, 4)

i = nodul de „intrare” al muchiei (în cazul de fata, 2)

Putem deduce formula $((k * p - 1) \vee (k * p) \vee (k * i - 1) \vee (k * i))$. Pentru cazul general, clauza ar arata așa: $(k * p - j) \vee (k * i - j)$, cu j de la 0 la k – 1. Ca și insemnatate, analizăm dacă fiecare nod care face parte dintr-o muchie este ales la pasul i.

Partea 2:

$$(1V3V5V7V9) \wedge (\sim 1V \sim 3) \wedge (\sim 1V \sim 5) \wedge (\sim 1V \sim 7) \wedge (\sim 1V \sim 9) \wedge (\sim 3V \sim 5) \wedge (\sim 3V \sim 7) \wedge (\sim 3V \sim 9) \wedge (\sim 5V \sim 7) \wedge (\sim 5V \sim 9) \wedge (\sim 7V \sim 9) \wedge (2V4V6V8V10) \wedge (\sim 2V \sim 4) \wedge (\sim 2V \sim 6) \wedge (\sim 2V \sim 8) \wedge (\sim 2V \sim 10) \wedge (\sim 4V \sim 6) \wedge (\sim 4V \sim 8) \wedge (\sim 4V \sim 10) \wedge (\sim 6V \sim 8) \wedge (\sim 6V \sim 10) \wedge (\sim 8V \sim 10)$$

Pentru partea a doua m-am folosit de o formula pe care am învățat-o la laboratorul 11 de AA: pentru 3 variabile codate 1, 2, 3, pentru a testa ca doar una din ele ia valoarea true, putem folosi secvența : $(1V2V3) \wedge (\sim 1V \sim 2) \wedge (\sim 1V \sim 3) \wedge (\sim 2V \sim 3)$. Am recunoscut aceasta relație și mai sus, însă de ordin 5. Astfel, pe cazul de fata putem sparge outputul în 2 părți:

$$(1V3V5V7V9) \wedge (\sim 1V \sim 3) \wedge (\sim 1V \sim 5) \wedge (\sim 1V \sim 7) \wedge (\sim 1V \sim 9) \wedge (\sim 3V \sim 5) \wedge (\sim 3V \sim 7) \wedge (\sim 3V \sim 9) \wedge (\sim 5V \sim 7) \wedge (\sim 5V \sim 9) \wedge (\sim 7V \sim 9)$$

Aceasta se refera la primul din cele $K = 2$ noduri pe care le alegem și înseamnă ca orice nod poate fi ales primul, însă doar unul poate fi ales. Astfel, dacă sunt alese cel puțin 2 noduri, aceasta secvența va returna 0(false).

$$(2V4V6V8V10) \wedge (\sim 2V \sim 4) \wedge (\sim 2V \sim 6) \wedge (\sim 2V \sim 8) \wedge (\sim 2V \sim 10) \wedge (\sim 4V \sim 6) \wedge (\sim 4V \sim 8) \wedge (\sim 4V \sim 10) \wedge (\sim 6V \sim 8) \wedge (\sim 6V \sim 10) \wedge (\sim 8V \sim 10)$$

Asemenea celei de mai sus, secvența aceasta se refera la al doilea nod ales din cele K pe care trebuie sa le alegem. Explicatia este aceeași ca mai sus.

$$(\sim 1V \sim 2) \wedge (\sim 3V \sim 4) \wedge (\sim 5V \sim 6) \wedge (\sim 7V \sim 8) \wedge (\sim 9V \sim 10)$$

Ultima parte a formulei din output se refera la faptul ca un nod nu poate fi ales de mai multe ori. Astfel, o formula generala ar fi :

$$\wedge_{i \leq n} (V_{j < k} \sim (i * K - j))$$

În final, conform codului, care se poate găsi pe HackerRank, algoritmul făcut are o complexitate de $O(N^2)$.