TEMA LA ALGORITMICA GRAFURILOR

Cehan Petru Serban / Cehan Dan Stefan

Grupa B2

1 Problema1

a)
Inegalitatea $|S| \leq |T|$ este evidenta in felul urmator:

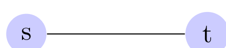
$|S|$=$\sum_{st\in E} \frac{1}{d_G(s)} \leq \sum_{st\in E} \frac{1}{d_G(t)} = |T|$,

pentru ca fiecare $x \in X$ apare de $d_G(x)$ ori in X si $|X|$ va fi format prin insumare de $\frac{d_G(x)}{d_G(x)} = 1$.

Cum $|S| \geq |T|$ (ipoteza), egalitatea rezulta imediat.

Pentru a arata ca fiecare componenta conexa a lui G este graf regulat (evident ca un graf este bipartit $\iff$ fiecare componenta conexa este bipartita), trebuie sa aratam ca pentru orice 2 noduri $s \in S$ si $t \in T$ pentru care $st \in E$, $|S| = |T|$ atunci $d_G(s) = d_G(t)$. Sa demonstram prin inductie dupa lungimea bipartitiilor.

Baza. $|S| = |T| = 1$. Deoarece G este bipartit si nu are varfuri izolate rezulta ca $d_G(s) = d_G(t) = 1$, $s \in S$ si $t \in T$.



Pas inductiv. Presupun ca pentru $|S| = |T| = k$, pentru orice muchie st, $s \in S$ si $t \in T$ are loc $d_G(s) = d_G(t)$.
Sa demonstram pentru $|S| = |T| = k + 1$. Prin R.A presupunem ca exista 2 noduri $s' \in S$ si $t' \in T$, $s't' \in E$ pentru care $d_G(s') \neq d_G(t')$.

$|E| = \sum_{s\in S} d_G(s) = \sum_{s\in S, s\neq s'} d_G(s) + d_G(s') = \sum_{s\in S-s'} d_{G-s'-t'}(s) + d_G(s')$.

Totodata, are loc si :
$|E| = \sum_{t\in T} d_G(t) = \sum_{t\in T, t\neq t'} d_G(t) + d_G(t') = \sum_{t\in T-t'} d_{G-t'-s'}(t) + d_G(t')$.

Deoarece $G - s' - t'$ ramane bipartit (ca subgraf a unui graf bipartit) si bipartitiile $S - s'$ si $T - t'$ sunt de lungime k rezulta ca pentru orice muchie st, $s \in S - s'$ si $t \in T - t'$ are loc $d_G(s) = d_G(t)$ (pasul inductiv). Mai precis, cele doua sume de mai sus vor fi egale, lucru ce ar implica faptul ca $d_G(s') = d_G(t')$. Contradictie.

Asadar, pentru orice doua varfuri adiacente in G avem gradele lor egale. Rezulta imediat ca fiecare componenta conexa este un graf regulat.

b)
Am aratat la punctul a ca fiecare componenta conexa este un graf $k - regulat$.
Teorema.
Fiecare graf $k - regulat$ bipartit,conex are un cuplaj perfect.
Demonstratie. Fie G un graf bipartit conex $k - regulat$ cu bipartitia (S, T). Fie $X \subseteq S$ si e numarul de muchii cu un capat in in X. Deoarece fiecare varf din X are gradul $k \implies k|X| = e$. In mod similar, fiecare varf din $N(X)$ (multimea de vecini ai nodurilor din X) are gradul k, dintr-un motiv similar. Rezulta ca $e \leq k|N(X)| \iff k|X| \leq k|N(X)| \iff |X| \leq |N(X)|$. Din teorema lui Hall, rezulta ca exista un cuplaj ce satureaza toate varfurile lui S. Analog se arata ca exista un cuplaj ce satureaza si varfurile lui T, deci se obtine un cuplaj perfect in G, deoarece $|S| = |T|$.
Egalitatea $|S| = |T|$ se poate arata in mai multe moduri. Deoarece G este $k - regulat$ si tinand cont ca $\sum_{s \in S} d_G(s) = \sum_{t \in T} d_G(t) = |E| \iff k|S| = k|T| \iff |S| = |T|$ pentru ca fiecare nod din S si fiecare nod din T au gradul k.
Desi am folosit si mai sus relatiile $\sum_{s \in S} d_G(s) = \sum_{t \in T} d_G(t) = |E|$, pentru orice G bipartit, sa demonstram ca lucrurile stau chiar in acest mod.
Prin inductie dupa $|E|$.

Baza. $|E| = 1 \implies$ trebuie sa existe o muchie intre S si T si clar ca sumele gradelor in ambele bipartitii este 1, deci egale.

Pas inductiv. Presupunem ca pentru $|E| = k$ avem egalitatea celor doua sume, i.e

$\sum_{s \in S} d_G(s) = \sum_{t \in T} d_G(t) = k.$

Consideram un graf bipartit cu $|E| = k + 1$. Stergem o muchie si obtinem un graf bipartit cu $|E|$ = k si din pasul inductiv stim ca $\sum_{s \in S} d_{G'}(s) = \sum_{t \in T} d_{G'}(t) = |E| = k$, unde G' este graful obtinut din G prin stergearea unei muchii. Si, muchia stearsa contribuie la suma in ambele parti cu 1 si prin adaugarea ei se pastreaza egalitatea.
Relatiile privind sumele sunt demonstrate.

Acum, cuplajul perfect se va construi pentru fiecare componenta conexa in parte(graf regulat bipartit) si prin reuniunea lor va rezulta un cuplaj perfect in G.

2 Problema2

a)
Descriem reteaua construita astfel :
$V_R = V \cup \{s, t\}$
$E_R = \{uv, vu | uv \in E\} \cup \{si | i \in V\} \cup \{it | i \in V\}$
$c_{ij} = 1 \; \forall ij \in E$, $c_{si} = m$, $c_{it} = m + 2g - d_G(i) \; \forall i \in V$ si $c_{ij} = 0$ daca mai exista alte arce (i.e care nu apartin lui E_R).

Partitionand V_R in doua multimi, S si T ($s \in S$ si $t \in T$), se observa o taietura $s-t$. Fie $V_1 = S$ - $\{s\}$ si $V_2 = T$ - $\{t\}$.
Daca V_1 este vida, atunci capacitatea sectiunii $c(S,T) = m|V|$.
Altfel, $c(S,T) = \sum_{i\in S, j\in T} c_{ij} = \sum_{j\in V_2} c_{sj} + \sum_{i\in V_1} c_{it} + \sum_{i\in V_1, j\in V_2} c_{ij} = m|V_2|$

$+ (m|V_1|+2g|V_1|-\sum_{i\in V_1} d_G(i)) + \sum_{i\in V_1, j\in V_2} c_{ij} = m|V| + 2|V_1|\ (g- \frac{\sum_{i\in V_1} d_G(i) - \sum_{i\in V_1, j\in V_2} 1}{2|V_1|})$

Deoarece $\frac{\sum_{i\in V_1} d_G(i) - \sum_{i\in V_1, j\in V_2} 1}{2}$ reprezinta numarul de muchii ale subgrafului indus de nodurile din V_1 in G, rezulta ca

$f(V_1) = \frac{\sum_{i\in V_1} d_G(i) - \sum_{i\in V_1, j\in V_2} 1}{2|V_1|})$
Deci $c(S,T) = m|V| + 2|V_1|(g - f(V_1))$.

In continuare, din teorema flux maxim taietura minima, rezulta ca valoarea fluxului maxim este egala cu valoarea unei taieturi minime (de capacitate minima). Fie aceasta C. Din ipoteza stim ca $C < mn = m|V|$. Evident ca daca V_1 este vida, $C = m|V|$ deci nu se intampla. Daca V_1 nu este vida, $m|V| > m|V| + 2|V_1|(g - f(V_1)) \iff 2|V_1|(g - f(V_1)) < 0$. Rezulta imediat ca $f(V_1) > g$ si V_1 subgraf a lui G.

b)
Sa privim mai intai ce valori poate lua f.
Pentru orice graf G , $f(G) \in \{\frac{m'}{n'}, 0 \le m' \le m, 1 \le n' \le n\}$, unde $|V(G)| = n$ si $|E| = m$. Deci $0 \le f(G) \le m$, pentru orice graf G.

Presupunem prin reducere la absurd ca G^0 nu este solutie a problemei P. Atunci fie G^1 diferit de G^0 solutie a problemei P (problema sigur are solutie deoarece sigur exista m si n astfel incat raportul sa fie maxim, de fapt se incearca maximizarea unei fractii). Scriem $f(G^0) := \frac{m^0}{n^0}$ si $f(G^1) := \frac{m^1}{n^1}$. Prin scadere se obtine:

$D = \frac{m^0}{n^0}$ - $\frac{m^1}{n^1} = \frac{m^0n^1 - m^1n^0}{n^0n^1}$
Daca $n^0 = n^1 \implies |D| = |\frac{m^0-m^1}{n^0}| \ge |\frac{m^0-m^1}{n}| \ge \frac{1}{n}$ pentru ca m^0 difera de m^1 prin macar o unitate, altfel ar fi acelasi subgraf.
Altfel daca $n^0 \ne n^1 \implies n^0n^1 \le n(n-1) \implies |D| \ge \frac{1}{n(n-1)}$.
Asadar, in ambele cazuri gasim o contradictie la relatia data in ipoteza de la punctul *b)*.
Deci G^0 este solutia problemei P.

$c)$
Algoritmul este prezentat in continuare :
$SolveP()$
{
  $st = 0$; $dr = m$; $V_1 = \emptyset$;
  $while(dr - st \geq \frac{1}{n(n-1)})$
  {
    $g = \frac{(st+dr)}{2}$;
    Construieste $R = (V_R, E_R)$;
    Gaseste taietura-minima (S,T); // echivalent cu a gasi fluxul maxim

    $if\ S = \{s\}\ then$
      $dr = g$;
    $else$
    {
      $st = g$;
      $V_1 = S$ - $\{s\}$;
    }
  }
return subgraful indus de V_1;
}

V_1 va avea $f(V_1) \geq st$ pe tot parcursul algoritmului. Cand algoritmul se termina, datorita punctului $b)$ stim sigur ca nu exista un subgraf H cu valoarea $f(H) \geq st + \frac{1}{n(n-1)}$, deci subgraful returnat va avea f maxim (i.e V_1).

Algoritmul are complexitatea timp $O(A(n, n+m)logm)$ deoarece bucla while se executa in timp logaritmic. Pentru fiecare iteratie a buclei, se construieste reteaua cu $n+2$ noduri si $2m+2n$ muchii, deci $O(n)$ varfuri si $O(n+m)$ muchii. Iar gasirea taieturii minime echivaleaza cu a gasi fluxul maxim deci complexitatea $A(n, n+m)$.

3 Problema3

$a)$
In primul rand sa observam ca orice digraf admite o multime 2hops stabila. Acest lucru il demonstram chiar aratand cum se construieste o astfel de multime, bineinteles in timp liniar.
Fie $D_1 = (V, E_1)$ si $D_2 = (V, E_2)$ unde $E_1 = \{xy \in E | x < y\}$ si $E_2 = \{xy \in E | y < x\}$. Deoarece D_1 si D_2 sunt aciclice (prin modul de constructie) atunci in fiecare digraf D_i exista o multime stabila (i.e o multime de noduri intre care nu exista arce si din care exista un arc direct in celelalte noduri, mai exact 1 hop stabila), chiar unica (este usor de demonstrat acest lucru, fie X multimea care se construieste, deoarece digraful este aciclic exista cel putin un nod cu gradul interior 0, acestea se pun prima oara in X, apoi se repeta aceasta operatie in

mod recursiv, adaugand nodurile cu gradul interior 0 din D - $N^+(X)$, la X, si asa mai departe, in final X fiind o multime 1 hop stabila. Este evident ca o astfel de multime X se construieste in timp liniar. Nu ne intereseaza sa demonstram unicitatea, dar se poate usor procedand prin reducere la absurd). Atunci fie H_1 aceasta multimea 1 hop stabila pentru D_1 si H_2 multimea 1 hop stabila pentru sub(di)graful lui D_2 indus de nodurile din H_1(se iau in consideratie doar arcele ce pleaca din nodurile din H_1 nu si cele care intra, daca exista). Deoarece $d(H_1, v) \leq 1$ $\forall v \in V$ si $d(H_2, x) \leq 1$ $\forall x \in H_1$ rezulta ca $d(H_2, v) \leq 2$ $\forall v \in V$. Urmeaza imediat ca H_2 este 2 hops stabila in D.

In continuare, descriem algoritmul liniar care construieste o multime 2 hops stabila intr-un digraf D.

$2HOPSTABLESET(D, V, E)$
{
    Construieste D_1 si D_2 ca mai sus;
    $H_1 = 1HOPSTABLESET(D_1, V, E_1)$;
    Construieste SD_2-subdigraf a lui D_2 indus de nodurile din H_1;
    $H_2 = 1HOPSTABLESET(SD_2, V_{SD_2}, E_{SD_2})$;
    return H_2;
}

Functia $1HOPSTABLESET$ returneaza o multime 1hop stabila si o prezentam in continuare :

$1HOPSTABLESET(D, V, E)$
{
    $S = \emptyset$;
    $while(D$ is not empty)
    {
        $S' = \{x|\ x \in V$ si $d_D^-(x) = 0\ \}$;
        $S = S \cup S'$;
        $D = D - S' - N_D^+(S)$;
    }
    return S;
}

Corectitudine :
*Lema*1. Functia $1HOPSTABLESET$ returneaza pentru digraful aciclic $D = (V, E)$ o multime 1hop stabila.
Demonstratie. La fiecare pas S' este o multime stabila(i.e nu exista arce intre nodurile din ea deoarece gradul interior al fiecarui nod este 0) si S este stabila pentru ca nu exista vreun arc intre 2 noduri din S' la iteratii diferite (deoarece toti vecinii exteriori unui nod din S' sunt indepartati din D la fiecare iteratie si D este aciclic). Asadar este si 1 hop stabila pentru ca, fiind stabila, in plus

pentru orice nod $x \in V - S$ exista un nod y in S astfel inca $yx \in E$.
*Lema*2. Functia $2HOPSTABLESET$ returneaza pentru un digraf $D = (V, E)$ o multime 2 hops stabila.
Demonstratie. Datorita observatiilor din constructia de la inceput.

In plus, precizam ca in cazul in care nodurile digrafului nu sunt etichetate cu numere (i.e 1,...,n) trebuie stabilita o relatie de ordine totala pe nodurile digrafului, inclusiv in digrafurile D_1 si D_2 construite.
De asemenea, un rezultat bine cunoscut este ca orice digraf aciclic finit are macar un nod cu gradul interior 0 si macar un nod cu gradul exterior 0.
Demonstratie. Fie $P = (v_1, v_2, ..., v_k)$ un drum de lungime maxima in D. Atunci v_1 are gradul interior 0 si v_k are gradul exterior 0. Presupunem ca de exemplu exista arc xv_1. Atunci fie $x \notin v_2, v_3, ..., v_k$, dar atunci $(x, v_1, ..., v_k)$ este un drum de lungime mai mare decat P (fals); fie $x = v_i$ dar se formeaza un ciclu (fals iar).

Complexitate timp :
Functia $1HOPSTABLESET$ s-ar putea implementa in mod echivalent in complexitatea $O(|V|+|E|)$ daca s-ar proceda in felul urmator : deoarece digraful este aciclic se poate face o sortare topologica ; primul nod din sortarea topologica are gradul interior 0 intotdeauna ; asadar se pune nodul in S, se coloreaza cu negru si toti vecinii exteriori acestui nod se coloreaza cu negru ; se continua cu urmatorul nod in sortarea topologica inca necolorat, se pune in S se coloreaza cu negru si iar vecinii exteriori se coloreaza cu negru. Se continua in acest fel pana se coloreaza toate nodurile. $1HOPSTABLESET$ implementata in acest fel are complexitatea $O(|V|+|E|)$ deoarece preprocesarea (sortarea topologica se face in aceasta complexitate) si apoi pentru fiecare nod necolorat inca il colorez impreuna cu toti vecinii sai, deci tot $O(|V| + |E|)$. Algoritmul echivaleaza cu cel precedent deoarece toate nodurile de grad interior 0 vor face oricum parte din S (deoarece n-are cum sa se coloreze cu negru pentru ca nu intra niciun arc in el) si in plus "scapam" de pasul de stergere a nodurilor si muchiilor din digraful D (i.e urmatorul nod necolorat ar fi avut gradul interior 0 daca s-ar fi sters muchiile care intra in el, deci e corect) si de verificarea gradelor interioare daca sunt egale cu 0.
Functia $2HOPSTABLESET$ are complexitatea $O(|V| + |E|)$ pentru ca digrafurile D_1 si D_2 se construiesc in timp liniar pe baza listelor de adiacenta existente pentru D (i.e pentru D_1 se scoate din fiecare lista a nodului i nodurile j cu $j < i$ si pentru D_2 se procedeaza similar). Cele 2 apeluri ale functiei $1HOPSTABLESET$ se executa in timp liniar si constructia subdigrafului SD_2 tot in timp liniar.

Asadar algoritmul prezentat este corect si se executa in timpul $O(|V| + |E|)$.

b)
Fie definita urmatoarea problema de decizie :

$2HOPSTABLE$
Instanta: $D = (V, E)$ digraf, $x \in V$.
Intrebare: Exista in D o multime 2hops stabila care sa contina pe x ?

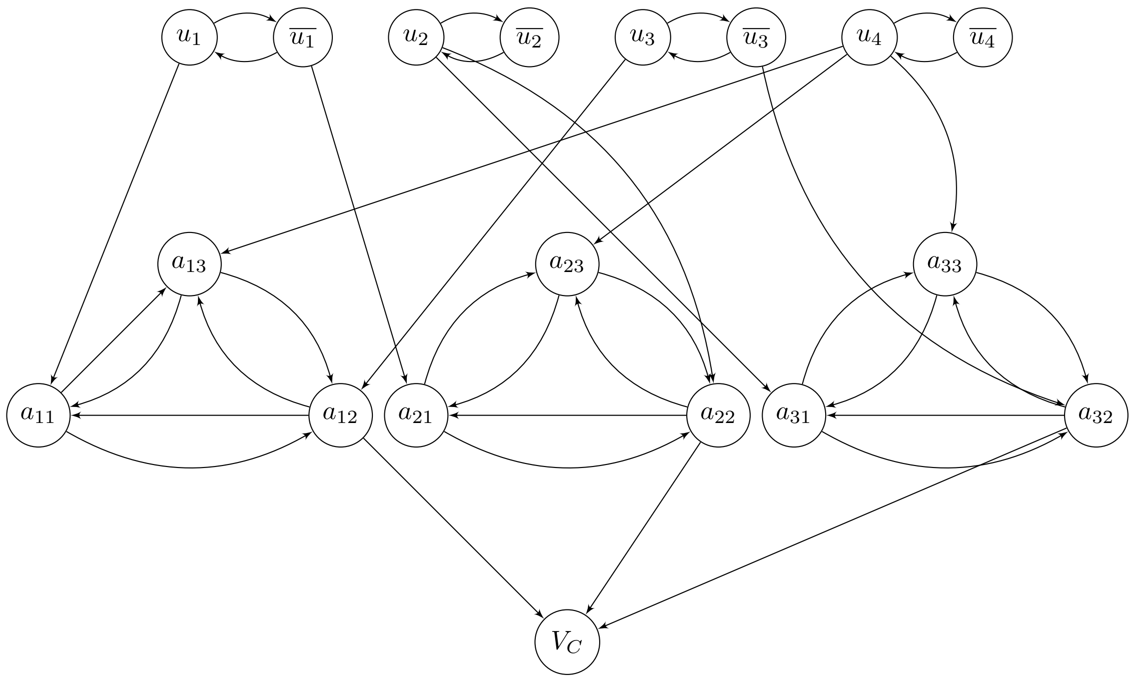
Demonstram ca $3SAT \ \alpha \ 2HOPSTABLE$.
Fie $U = \{u_1, u_2, ..., u_n\}$ $(n \in \mathbb{N}^*)$, $C = C_1 \wedge C_2 \wedge ... \wedge C_m$ $(m \in \mathbb{N}^*)$ si $C_i = v_{i_1} \vee v_{i_2} \vee v_{i_3}$, $\forall i = 1...m$, (unde $\forall v_{i_j}$ $\exists \alpha \in \{1, .., n\}$ astfel incat $v_{i_j} = u_\alpha$ sau $v_{i_j} = \overline{u_\alpha}$), reprezentand datele unei instante oarecare a problemei $3SAT$.

Vom construi in timp polinomial in raport cu $n + m$ un digraf D astfel incat exista o atribuire t a valorilor de adevar/fals pentru variabilele booleene din U ca sa faca adevarata formula $C \iff$ exista in digraful construit o multime 2hops stabila care sa contina un varf al digrafului dat. Constructia este asemanatoare celei din curs referitoare la reducerea $3SAT \ \alpha \ SM$.

D se construieste astfel :
1. Pentru orice $i \in \{1, ..., n\}$ consideram digrafurile disjuncte formate din perechi de arce simetrice $T_i = (\{u_i, \overline{u_i}\}, \{u_i\overline{u_i}, \overline{u_i}u_i\})$.
2. Pentru orice $j \in \{1, ..., m\}$ consideram digrafurile disjuncte formate din arce simetrice $Z_j = (\{a_{j1}, a_{j2}, a_{j3}\}, a_{j1}a_{j2}, a_{j2}a_{j1}, a_{j1}a_{j3}, a_{j3}a_{j1}, a_{j2}a_{j3}, a_{j3}a_{j2})$.
3. Pentru orice $j \in \{1, ..., m\}$ consideram multimea de arce $E_j = \{v_{j1}a_{j1}, v_{j2}a_{j2}, v_{j3}a_{j3}\}$, unde $v_{j1} \vee v_{j2} \vee v_{j3}$ este factorul C_j.
4. Daca V_C este varful dat din digraf, atunci se introduc si arcele dinspre un nod din fiecare triunghi de arce creat spre V_C.

Evident, constructia este polinomiala in raport cu $m + n$, i.e D are $2n + 3m + 1$ varfuri si $2n + 10m$ arce.

Exemplu: $U = \{u_1, u_2, u_3, u_4\}$.
$C = (u_1 \vee u_3 \vee u_4) \wedge (\overline{u_1} \vee u_2 \vee u_4) \wedge (u_2 \vee \overline{u_3} \vee u_4)$.
$V_C =$ nodul dat din digraf.



Folosind acest "gadget" vom demonstra ca $3SAT \ \alpha \ 2HOPSTABLE$.

Demonstratie implicatie directa : $3SAT \implies 2HOPSTABLE$.
Sa presupunem ca raspunsul la problema $3SAT$ este da, deci exista o atribuire $t : U \to \{A, F\}$ astfel incat $t(C_j) = A$, $\forall j = \overline{1, m}$.
In continuare demonstram ca pentru digraful D construit mai sus si un varf V_C fixat, raspunsul la problema $2HOPSTABLE$ este da, (i.e exista o multime 2hops stabila in D care sa-l contina pe V_C).
Fie $S = \{u_i | u_i = true\} \cup \{\overline{u_i} | u_i = false\}$ (i.e multimea literalilor care sunt true, prin literal ma refer inclusiv la negatia sa). Este evident ca nu poate fi si u_i si $\overline{u_i}$ in S, pentru i fixat. Asadar, deoarece formula C este satisfiabila, i.e $t(C) = A$, dupa cum am zis mai sus, rezulta ca in fiecare clauza exista macar un literal $true$ care isi va "domina" triunghiul Z_j in care se duce, prin drumuri de lungime cel mult 2. De asemenea, fiecare nod din digrafurile T_i vor fi ori in S ori vor fi atinse prin arc direct de un nod din S, deci prin drum de lungime 1. Sa observam ca V_C nu este adiacent cu niciun nod din digrafurile T_i si este posibil ca acesta sa nu poata fi atins prin drum de lungime cel mult 2, ci prin drum de lungime 3. Prin urmare, il adaugam la multimea S, ea ramanand in continuare stabila.
In final, multime S va fi multimea 2hops stabila ce contina pe V_C si deci raspunsul la problema $2HOPSTABLE$ este da.

Demonstratie implicatie inversa : $2HOPSTABLE \implies 3SAT$.

Presupunem ca raspunsul la problema $2HOPSTABLE$ pentru instanta D si V_C fixat este *da*. Fie S o multime 2 hops stabila ce contine pe V_C. Din definitia unei astfel de multimi, rezulta ca S poate avea cel mult un varf din orice $V(T_i)$ si cel mult un varf din orice $V(Z_j)$ $(i = \overline{1,n},\ j = \overline{1,m})$.

Definim $t : U \rightarrow \{A, F\}$ astfel : $t(u_i) = \begin{cases} F \text{ , daca } S \cap V(T_i) = \{\overline{u_i}\} \\ A \text{ , daca } S \cap V(T_i) = \{u_i\} \end{cases}$

Sa observam ca V_C face parte din multimea noastra 2hops stabila S. Dar V_C nu domina nici un nod din digraful nostru construit, deci raman nodurile din T_i si Z_j inca neatinse (prin "domina" ma refer ca nodul atinge alte noduri prin drumuri de lungime cel mult 2 in contextul nostru). Deoarece in orice u_i nu pot ajunge decat prin negatia sa, inseamna u_i ori este in S, ori este dominat de un nod din S (i.e negatia sa). Prin modul de definire a lui t rezulta ca orice literal *true* din S isi va domina clauza din care face parte, mai exact un anume triunghi Z_j. Deci pentru orice j, $t(C_j) = A$, deci $t(C) = A$.
Asadar, avand t definita in acest fel, se observa ca formula C este satisfiabila, deci raspunsul la problema $3SAT$ este *da*.