ΕΘΝΙΚΟ ΜΕΤΣΟΒΙΟ ΠΟΛΥΤΕΧΝΕΙΟ

ΣΧΟΛΗ ΗΛΕΚΤΡΟΛΟΓΩΝ ΜΗΧ/ΚΩΝ & ΜΗΧ/ΚΩΝ ΥΠΟΛΟΓΙΣΤΩΝ
Μάθημα: "Αλγόριθμοι και πολυπλοκότητα"
Λουκάς Άγγελος 03119877

3η Σειρά Ασκήσεων

Άσκηση 1: Υπολογισιμότητα

(α) Για το πρόβλημα των **Διοφαντικών Εξισώσεων**, μπορούμε να καταλάβουμε εάν υπάρχουν ακέραιες λύσεις, όμως δεν υπάρχει γενικός αλγόριθμος που να μπορεί πάντα να βρει τη λύση. Η απόδειξη αυτού του γεγονότος πραγματοποιήθηκε από τον Yuri Matiyasevich και δημοσιεύτηκε στο βιβλίο του [Matiyasevich, Y. V. (1970). "Hilbert's Tenth Problem"].

Ορισμός: Ένα set $S \subseteq N$ ορίζεται ως διοφαντικό εάν υπάρχει μια πολυωνυμική P με βαθμό $k+1$ τέτοια ώστε να ισχύει $P(S)=0$.

Για την αναζήτηση του διοφαντικού set της P θα περάσουμε όλα τα διαφορετικά δυνατά $S \subseteq N$ με έναν ημιαποφασιστικό αλγόριθμο. Ο αλγόριθμος αυτός για κάθε $k \in N$ ξεκινώντας από $k=0$ και με την ιδιότητα πως $\sum_{i=1}^{n} |x_i| = k$ θα τρέξει την πολυωνιμική P με το $S=[x_1, x_2 \dots x_n]$, αυξάνοντας την τιμή του k μετά από κάθε λανθασμένη προσπάθεια. Με τη συστηματική αυτή εξέταση ο αλγόριθμος δίνει το πρώτο set S που επιτυγχάνει. Αυτή η διαδικασία απαριθμεί αποτελεσματικά τα στοιχεία του διοφαντικού συνόλου. Μέσω μιας βολικής επίκλησης στο Church's Thesis, το οποίο υποστηρίζει ότι οτιδήποτε είναι υπολογίσιμο μπορεί να υπολογιστεί από μια μηχανή Turing, αναγνωρίζουμε ότι το διοφαντικό σύνολο είναι ημιαποφασιστικό. Με απλούστερους όρους, μπορούμε να απαριθμήσουμε συστηματικά τα στοιχεία του, αν και χωρίς να εγγυηθούμε μια στάση για τις περιπτώσεις όπου δεν υπάρχει λύση.

(β) Για να δείξουμε πως το **Πρόβλημα Τερματισμού (HP)** είναι RE-πλήρες θα αποδείξουμε αρχικά πως ανήκει στην κλάση RE και ύστερα πως ανήκει στην κλάση RE-πλήρες.

Θα ανάγουμε το *HP* από το Πρόβλημα Διοφαντικών Εξισώσεων (ΠΔΕ).
Έστω:

- Ένα πρόγραμμα που τρέχει την συνάρτηση του *ΠΔΕ* με είσοδο μια οποιασδήποτε πολυωνυμική εξίσωση *P* όπως περιγράφηκε στο προηγούμενο ερώτημα.
- Το πρόγραμμα *HALT* που δέχεται το παραπάνω πρόγραμμά και απαντά *YES* αν το πρόγραμμα αυτό τερματίζει και *NO* αν δεν τερματίζει.

Στην περίπτωση που υπάρχει ακέραια λύση στην οποιαδήποτε πολυωνυμική εξίσωση πρόγραμμα *HALT* θα επιστρέφει *YES*. Αν η πολυωνυμική εξίσωση δεν έχει ακέραια λύση το *HALT* θα επιστρέφει *NO*. Με αυτόν τον τρόπο μπορούμε να λύσουμε το *ΠΔΕ* με χρήση του *HP*.

Γνωρίζουμε όμως από το προηγούμενο ερώτημα πως το πρόβλημα *ΠΔΕ* ανήκει στην κλάση *RE*, δηλαδή το *HP* επιστρέφει *YES* αν βρει ακέραια λύση αλλά δεν μπορούμε να ξέρουμε αν θα τερματίσει η *ΠΔΕ* ώστε να μπορέσει να επιστρέψει τιμή *NO*. Αφού δεν θα επιστρέψει *NO* και μπορεί να επιστρέψει μόνο τιμή *YES* όταν βρεθεί λύση, είναι προφανές πως το Πρόβλημα Τερματισμού ανήκει στην κλάση *RE*.

Αν υποθέσουμε ένα οποιοδήποτε πρόβλημα *X* της κλάσης *RE* και ανάγουμε με τον ίδιο τρόπο το *HP* από την κλάση *X* μπορούμε πολύ εύκολα να δείξουμε πως εφόσον το *X* είναι της κλάσης *RE*,το *HP* θα επιστρέφει τιμή *YES* για επιτυχή επίλυση του *X* αλλά δεν είναι σίγουρο αν τελειώσει ποτέ ο αλγόριθμος.

Αν το *HP* θα μπορούσε να μας επιστρέφει *YES* ή *NO* για κάθε πρόβλημα *X* της κλάσης του *RE*, θα αποδεικνύαμε πως το *X* δεν είναι πλέον μέλος του *RE*. Αυτό θα οδηγούσε σε αφαίρεση όλων των προβλημάτων *X* από την κλάση *RE* στην κλάση **Recursive (R).**

Συνεπώς το *HP* ανήκει στην RE-πλήρες.

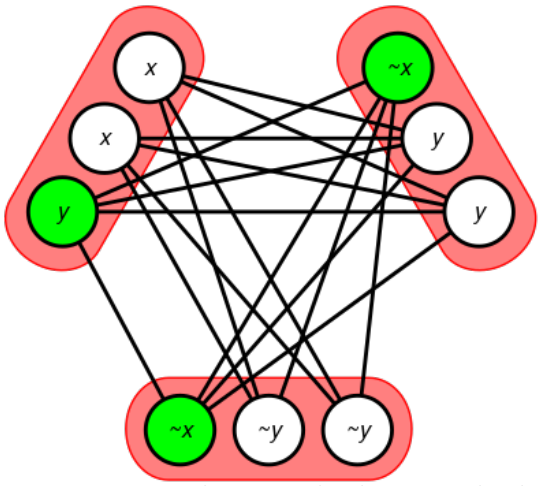
(γ) Για να δείξουμε πως το **Πρόβλημα Καθολικού Τερματισμού (UHP)** είναι *RE-δύσκολο* θα θεωρήσουμε το πρόβλημα *X* της κλάσης *RE*. Το μηχάνημα *M* θα παίρνει σαν είσοδο το *w* και για κάθε διαφορετικό πρόβλημα *X* μας θα τρέχει ένα συγκεκριμένο αλγόριθμο για την επίλυσή του. Αν το *M* σταματήσει τότε έχουμε έξοδο του προβλήματος μας *YES* αλλιώς έχουμε *NO* στην περίπτωση που τρέχει επ' άπειρον. Αυτή η αναγωγή αποδεικνύει ότι η επίλυση του UHP συνεπάγεται την ικανότητα επίλυσης οποιουδήποτε προβλήματος στην *RE*. Ενώ το *HP* είναι αναποφάσιστο, το *UHP* είναι ακόμη πιο πολύπλοκο επειδή απαιτεί τον προσδιορισμό της συμπεριφοράς διακοπής ενός προγράμματος σε όλες τις πιθανές εισόδους, όχι μόνο σε μία. Έτσι συμπεραίνουμε πως το *UHP* είναι *RE-δύσκολο*.

Άσκηση 2: Πολυπλοκότητα - Αναγωγές

(α) Αναγωγή από το πρόβλημα **UnSat** στο πρόβλημα **NoLargeClique**: Γνωρίζουμε πως κάθε πρόβλημα UnSat μπορεί να γραφεί σε **συζευκτική κανονική μορφή (CNF)** της μορφής $A_1 \wedge A_2 \wedge \ldots A_n$ όπου για τον κάθε όρο A_i να ισχύει $A_i = (\ldots \vee \ldots \vee \ldots)$, δηλαδή μια διάζευξη γραμματικών με στοιχεί που μπορούν να πάρουν τιμές x_m ή $\overline{x_m}$. Η αναγωγή του προβλήματος UnSat στο πρόβλημα NoLargeClique θα γίνει με βάση το CNF μας θεωρώντας το κάθε στοιχείο των ορών μας ώς κόμβους. Κάθε στοιχείο του A_i θα το συνδέσουμε με κάθε άλλο στοιχείο του A_j με $i \neq j$ εφόσον ισχύει το παρακάτω:

- αν $x_m \in A_i$ και $\overline{x_n} \in A_j$ τότε θα πρέπει $m \neq n$

Έτσι δημιουργήσαμε έναν γράφο όπου κάθε στοιχείο x_m συνδέεται με μία ακμή δύο κατευθύνσεων με κάθε άλλο στοιχείο που δεν ανήκει στον ίδιο όρο A_i ή είναι η άρνηση του x_m. (Ο σχεδιασμός του γράφου γίνεται σε πολυωνυμικό χρόνο και ο γράφος είναι πολυωνυμικού μεγέθους)



Παράδειγμα για $CNF = (y \vee x \vee x) \wedge (\overline{x} \vee y \vee y) \wedge (\overline{x} \vee \overline{y} \vee \overline{y})$

Αν το πρόβλημα NoLargeClique το τρέξουμε στον γράφο που δημιουργήσαμε, θα βρούμε το πολύ ένα στοιχείο (κόμβο) του κάθε όρου A_i σε οποιοδήποτε Clique. Το πρόβλημά μας θα βρεί για αυτόν τον γράφο ένα Clique μεγέθους το πολύ k που σημαίνει πως μπορούμε να πάρουμε τιμή *True* σε k το πολύ όρους του CNF μας.

Αν από την αναγωγή μας στο NoLargeClique επιστρέψει ο αλγόριθμός μας *True* για μέγεθος το πολύ k=d-1 όπου d ο αριθμός των όρων A_i, τότε το UnSat επιστρέφει *True* καθώς δεν υπάρχουν συνδυασμοί στοιχείων x_m και $\overline{x_m}$ ώστε να ικανοποιείται το CNF. Ενώ αν επιστραφεί *False* από το NoLargeClique τότε το UnSat επιστρέφει *False* και αφού το UnSat ανήκει στην κλάση coNP-complete τότε θα ανήκει το NoLargeClique στο coNP-complete.

(β) Η αναγωγή του προβλήματος Sat στο πρόβλημα LargeClique λειτουργεί με παρόμοιο τρόπο με την αναγωγή του UnSat στο NoLargeClique μόνο που ψάχουμε αν υπάρχει ακέραιος k (ίσο με τον αριθμό των όρων A_i) που να βρίσκει ένα clique μεγέθους k. Δεδομένου πως το Sat πρόβλημα ανήκει στην κλάση NP-complete αποδεικνύεται πως και το LargeClique ανήκει στην ίδια κλάση.

<u>Θεώρημα:</u>

Έστω μια κλάση πολυπλοκότητας C. Αν ένα πρόβλημα Π είναι C-πλήρες ως προς μια αναγωγή $\leq_R$ τότε το συμπλήρωματικό πρόβλημα Π′ είναι co C-πλήρες ως προς την $\leq_R$.

Με το παραπάνω θεώρημα μπορούμε να δείξουμε μέσα από την γνωστή αναγωγή Sat σε LargeClique, πως το πρόβλημα NoLargeClique ανήκει στην κλάση coNP-Complete.

Το ερώτημα (α) απέδειξε την κλάση του NoLargeClique μέσα από αναγωγή του UnSat (συμπληρωματικό του Sat). Αυτό δεν μας βόλευε στο ερώτημα αυτό καθώς για την χρήση του θεωρήματος χρειάστηκε να βρούμε την κλάση που ανήκει το συμπληρωματικό του NoLargeClique.

(γ) Τα NP-complete προβλήματα θεωρούνται δύσκολα, δηλαδή δεν μπορούν να επιλυθούν σε πολυωνυμικό χρόνο. Εάν ένα NP-complete πρόβλημα βρεθεί αν ανήκει στο $\boldsymbol{NP \cap coNP}$, σημαίνει ότι τόσο το πρόβλημα όσο και το συμπλήρωμα του μπορούν να επαληθευτούν αποτελεσματικά. Και αφού τα NP προβλήματα μπορούνε αν αναχθούνε στο NP-complete πρόβλημά μας, άρα όλα τα NP προβλήματα μετατρέπονται σε coNP δηλαδή $\boldsymbol{NP = coNP}$.

(ε) Σύμφωνα με το πρόβλημα που παρουσιάστηκε, εξετάζουμε την εφικτότητα της επιλογής αντιπροσώπων από διάφορες κοινωνικές ομάδες για να συμμετάσχουν σε ένα τοπικό κοινοβούλιο, δεδομένου ενός περιορισμένου αριθμού θέσεων. Αναζητούμε λύση που να εξασφαλίζει ότι κάθε κοινωνική ομάδα έχει τουλάχιστον έναν εκπρόσωπο στο κοινοβούλιο.
Η λύση μας χρησιμοποιεί γράφο, όπου οι κορυφές αντιστοιχούν σε κοινωνικές ομάδες, και οι ακμές δείχνουν κοινά μέλη μεταξύ τους ομάδων. Ένας χρωματισμός του γράφου αντιστοιχεί σε επιλογή αντιπροσώπων. Οι κορυφές με το ίδιο χρώμα υποδηλώνουν ότι οι αντίστοιχες ομάδες έχουν έναν κοινό εκπρόσωπο, ενώ οι κορυφές με διαφορετικά χρώματα αντιπροσωπεύουν διαφορετικές ομάδες.

Χρησιμοποιώντας αυτήν την προσέγγιση, αποδείξαμε ότι το πρόβλημα είναι NP-πλήρες, καθώς είναι τουλάχιστον τόσο δύσκολο όσο το γνωστό πρόβλημα του 3-Coloring.

Άσκηση 3: Προσεγγιστικοί Αλγόριθμοι

(α)

WeightedVertexCover$(G(V,E),w)$

1 $C \leftarrow \emptyset$;
2 $\forall\ v \in V, t(v) \leftarrow w(v);$
3 $\forall\ e \in E, c(e) \leftarrow 0;$
4 $k \leftarrow 1$
5 **while** C δεν είναι vertex cover **do**
6 $\quad e=\{u,v\}$ μια ακάλυπτη ακμή;
7 $\quad \delta \leftarrow min\{t(u), t(v)\};$
8 $\quad t(u) \leftarrow t(u)-\delta;$
9 $\quad t(v) \leftarrow t(v)-\delta;$
10 $\quad c(e) \leftarrow \delta;$
11 $\quad C \leftarrow C \cup \{z \in \{u,v\} \vee t(z)=0\};$
12 **return**(C)**;**

1.Το καλύπτει όλες τις ακμές:
Το while loop τελειώνει όταν το είναι vertex cover, δηλαδή όταν το C είναι ένα σύνολο κορυφών που καλύπτουν όλες τις ακμές. Κατά τη διάρκεια κάθε επανάληψης, επιλέγουμε μια ακμή που δεν έχει καλυφθεί (Γραμμή 6) και προσθέτουμε τουλάχιστον μία από τις δύο κορυφές που συνδέουν την ακμή αυτή στο (Γραμμή 11). Στη χειρότερη περίπτωση, όσον αφορά τον αριθμό των επαναλήψεων, θα επιλέγεται κάθε φορά μια κορυφή βαθμού 1. Συνεπώς, θα χρειαστούν $|E|$ επαναλήψεις για να καλυφθούν όλες οι κορυφές. Έτσι, το while loop τερματίζει όταν επιτυγχάνεται το επιθυμητό αποτέλεσμα (το C καλύπτει όλες τις ακμές).
Σημειώνεται ότι σε κάθε επανάληψη ισχύει $\boldsymbol{t(u)=0 \vee t(v)=0}$. Στην περίπτωση όπου $\boldsymbol{t(u)=0 \wedge t(v)=0}$, τότε και οι δύο κορυφές προστίθενται στο C.

2.Το συνολικό βάρος του C είναι μικρότερο ή ίσο του $2\sum_{e \in E} c(e)$**:**
Θεωρούμε τα παρακάτω

- e_i η i-οστή ακμή που επιλέγεται από τον αλγόριθμο στην Γραμμή 6.
- w_i το βάρος του συνόλου C στην i-οστή επανάληψη
- z_i η i-οστή κορυφή που επιλέγεται να μπει στο C
- $E_z(i)$ οι ακμές που προσπίπτουν στην κορυφή z_i

Με το βάρος να ορίζεται ως

$$w_i = w_{i-1} + c(e_i) + \sum_{e \in E_z(i) \setminus e_i} c(e)$$

με $w_0 = 0$

Αυτό, γιατί το βάρος που καταλήγει στο $w(z_i) = c(e_i) + \sum_{e \in E_z(i) \setminus e_i} c(e)$,καθώς σε κάθε επανάληψη μέχρι την i το βάρος που βρίσκεται στην $t(z_i)$ μειώνεται κάθε φορά που ο αλγόριθμος επισκέπτεται μία ακμή που προσπίπτει στην z_i . Στον τερματισμό, κάθε $c(e)$ θα έχει προστεθεί στο συνολικό βάρος κάθε φορά που μία από τις δυο κορυφές της ακμής e μπαίνουν στο C. Επειδή κατασκευάζουμε vertex cover αυτό θα γίνει τουλάχιστον 1, αλλά το πολύ 2 φορές. Συνεπώς:

$$W \leq 2 \cdot \sum_{e \in E} c(e)$$

Σημειώνουμε ότι, χωρίς βλάβη του συλλογισμού, μπορούμε να θεωρήσουμε ότι στις περιπτώσεις όπου $t(u) = t(v) = 0$, τα βάρη των δυο κορυφών που προστίθενται στο C στην ίδια επανάληψη του αλγορίθμου, μοντελοποιούνται σε δυο συνεχόμενες επαναλήψεις της αναδρομικής σχέσης.

3. Το συνολικό βάρος της βέλτιστης λύσης είναι μεγαλύτερο ή ίσο του $\sum_{e \in E} c(e)$:

Θεωρούμε τα παρακάτω

- W_{opt} το βάρος της βέλτιστης λύσης
- C^* το βέλτιστο vertex cover
- nακμή που βρίσκεται στο βέλτιστο vertex cover
- E_n το σύνολο των ακμών που προσπίπτουν στην ακμή n

Για κάθε ακμή n που βρίσκεται στο βέλτιστο vertex cover έχουμε δύο περιπτώσεις. Η πρώτη είναι η ακμή αυτή έχει το ελάχιστο βάρος μεταξύ όλων των γειτονικών ακμών της. Έτσι, όταν ο προσεγγιστικός αλγόριθμος επιλέξει για πρώτη φορά μια από τις ακμές που συνδέονται με τον κόμβο n, τότε αυτή η ακμή θα έχει βάρος $c(e) = \delta = w(n)$. Οι υπόλοιπες ακμές καλύπτονται από το C και, συνεπώς, το κόστος $c(e)$ για αυτές είναι 0, διότι δεν θα επιλεγούν από τον αλγόριθμο. Η δεύτερη είναι να υπάρχουν ακμές γειτονικές στον κόμβο n με μικρότερο βάρος, ή υπάρχουν ακμές που συνδέονται με άλλους κόμβους και έχουν μικρότερο $t(v)$ από το $t(n)$. Αυτό συμβαίνει επειδή η ακμή που επιλέγεται από τον αλγόριθμο σε προηγούμενη επανάληψη έχει μικρότερο βάρος. Σε κάθε επανάληψη που επιλέγεται μια τέτοια ακμή, το $t(n)$ μειώνεται. Εάν αυτή η ακμή επιλεχθεί να προστεθεί στο σύνολο C, τότε η πρώτη περίπτωση ισχύει ξανά.

Τελικά $$\sum_{e \in E_n} c(e) \leq w(n)$$

Προσθέτοντας όλα τα βάρη των κορυφών του βέλτιστου vertex cover έχουμε:

$$\sum_{n \in C^*} \sum_{e \in E_n} c(e) \leq W_{opt}$$

Δεδομένου ότι το $C*$ είναι ένα σύνολο κόμβων που αποτελεί κάλυψη κορυφών (vertex cover), για κάθε ακμή e, το βάρος c(e) της θα προστεθεί τουλάχιστον μία φορά στο αριστερό μέλος της παραπάνω ανίσωσης.

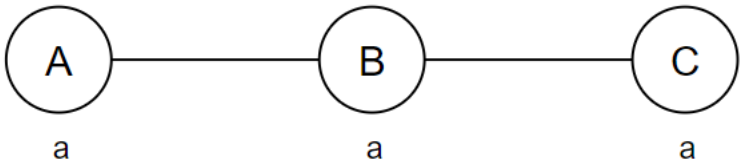
$$\sum_{e \in E} c(e) \leq W_{opt}$$

Τελικά θα έχουμε:

$$W_{opt} \geq \sum_{e \in E} c(e) \geq \frac{W}{2} \Rightarrow \frac{W}{W_{opt}} \leq 2$$

Άρα ο αλγόριθμος μας επιτυγχάνει λόγο προσέγγισης 2.

Ένα γράφημα όπου ο αλγόριθμος υπολογίζει ένα Vertex Cover με συνολικό βάρος διπλάσιο από αυτό της βέλτιστης λύσης:



Στο παράδειγμα αυτό έχουμε 3 κόμβους με ίδιο βάρος a. Η λύση του αλγορίθμου μας θα δώσει W=2a αφού θα διαλέξει δυο κόμβους μιας και έχουν τα ίδια βάρη. Η βέλτιστη λύση είναι W'= W/2 = a διαλέγοντας τον μόνο τον κόμβο B.

(β)
θα κάνουμε αρχικά αναγωγή του TSP στο πρόβλημα Hamilton Cycle.
Για το σύνολο των κόμβων V που ανήκουν στο TSP θα έχουμε τον γράφο $G=(V,E)$ με $E=(u,v,w(u,v))$ για κάθε $u,v\in V$ όπου $w(u,v)$ το βάρος της κάθε ακμής. Έχουμε έναν πλήρη γράφο τώρα και βρίσκοντας λύση για το Hamilton Cycle βρίσκουμε και μια λύση για το TSP (Ο Hamilton Cycle με το μικρότερο ολικό βάρος θα είναι και η βέλτιστη λύση του TSP). Αν δεν υπάρχει κύκλος Hamilton στον γράφο μας τότε δεν μπορεί να υπάρχει λύση στο TSP.

Μπορούμε να δούμε ότι μια k-προσεγγιστική λύση για το πρόβλημα TSP θα μπορούσε να αποκαλύψει την ύπαρξη ή μη κύκλου Hamilton στον γράφο G του προβλήματος Hamilton Cycle. Κατά συνέπεια, αν ένας πολυωνυμικός αλγόριθμος μπορεί να επιλύσει το πρόβλημα TSP με προσέγγιση με λόγο k, τότε θα μπορούσε επίσης να επιλύσει και το πρόβλημα Hamilton Cycle με πολυωνυμικό χρόνο. Αυτό θα σήμαινε ότι P = NP, καθώς το πρόβλημα Hamilton Cycle είναι γνωστό ότι είναι NP-πλήρες. Επομένως, κανένας πολυωνυμικός αλγόριθμος δεν μπορεί να επιτύχει προσέγγιση με λόγο k για το πρόβλημα TSP, εκτός εάν P = NP.