3η Σειρά Γραπτών Ασκήσεων

Αλγόριθμοι και Πολυπλοκότητα

Ονοματεπώνυμο: Καμπουγέρης Χαράλαμπος
ΑΜ: 03120098

Άσκηση 1: Υπολογισιμότητα

Η κλάση RE περιέχει όλα τα ημιαποκρίσιμα υπολογιστικά προβλήματα, αυτά για τα οποία υπάρχει αλγόριθμος που ημιαποφασίζει. Με τον όρο ημι-αποφασίζει, εννοούμε πως στην περίπτωση που η απάντηση σε κάποιο ζητούμενο ενός προβλήματος είναι ναι, τότε υπάρχει κάποιος αλγόριθμος που το καθορίζει σε πεπερασμένο χρόνο και χωρίς λανθασμένες επιβεβαιώσεις. Ωστόσο αν η απάντηση είναι όχι, ο αλγόριθμος ~~μπορεί~~ δεν είναι απαραίτητο να τερματίσει.

- Ισοδύναμα, η κλάση RE απαρτίζεται από τα προβλήματα απόφασης για τα οποία μια μηχανή Turing μπορεί να απαριθμήσει όλες τις θετικές απαντήσεις, δηλαδή όλα τα ναι.

a) Για να αποδείξουμε ότι το Πρόβλημα Διοφαντικών Εξισώσεων, με είσοδο μια πολυωνυμική εξίσωση με ακέραιους συντελεστές και πεπερασμένο αριθμό αγνώστων, και ζητούμενο αν έχει ακέραιες λύσεις ή όχι, ανήκει στην κλάση RE θα ~~κάν~~ κάνουμε το εξής:

Αρχικά, υποθέτουμε ότι υπάρχει ένα σύνολο αριθμών K, το οποίο προσδιορίζεται από τη Διοφαντική Εξίσωση $p(x, y_1, y_2, \dots, y_k) = 0$. Εδώ το πρόβλημα απαντάται με "ναι", αν το αποτέλεσμα p ισούται με 0 και με "όχι" ~~όχι~~ σε κάθε άλλη περίπτωση. Έτσι, αν εναλλάξουμε όλους τους πιθανούς μοναδικούς συνδυασμούς αναθέσεων τιμών στις μεταβλητές $x, y_1, y_2, \dots, y_k$ και καταγράφουμε κάθε τιμή του x που αντιστοιχεί σε αποτέλεσμα $p=0$, τελικά θα έχουμε απαριθμήσει όλα τα μέλη του σετ K. Ωστόσο, δεν είναι πάντα εφικτό να βρούμε όλες τις λύσεις μιας

Διοφαντικής Εξίσωσης, καθώς μερικές εξισώσεις μπορεί να έχουν άπειρες λύσεις ή να είναι ανεπίλυτες.

β) Το πρόβλημα τερματισμού HP μπορεί να ορισθεί ως η απόφαση του αν ένα αυθαίρετο υπολογιστικό πρόγραμμα θα σταματήσει να τρέχει ή θα συνεχίσει για πάντα.

Ένα πρόβλημα είναι RE-πλήρες αν:

→ ανήκει στην κλάση RE

→ κάθε πρόβλημα της κλάσης RE μπορεί να αναχθεί στο πρόβλημα αυτό με τρόπο που είναι συνεπές στους περιορισμούς της κλάσης.

Δηλαδή, αν μπορεί να λυθεί το δοθέν πρόβλημα, τότε μπορεί να λυθεί κάθε πρόβλημα της κλάσης RE. Για να δείξουμε πως το Halting Problem (HP) ανήκει στην κλάση RE, θα ανάγουμε το πρόβλημα των Διοφαντικών εξισώσεων σε αυτό ως εξής:

Η αναγωγή του προβλήματος των Διοφαντικών Εξισώσεων στο Halting Problem προϋποθέτει την απόδειξη ότι η επίλυση του ενός μπορεί να οδηγήσει στην επίλυση του άλλου. Πιο συγκεκριμένα, σημαίνει ότι εάν καταφέρουμε να λύσουμε το Halting Problem, τότε θα μπορούμε επίσης να λύσουμε το πρόβλημα της εύρεσης λύσεων σε Διοφαντικές Εξισώσεις.

Θεωρούμε ένα τυχαίο Turing Machine, δηλαδή μια τυχαία περίπτωση του HP. Θεωρούμε επίσης έναν αλγόριθμο που μετατρέπει μια Διοφαντική Εξίσωση σε ένα πρόγραμμα έτσι ώστε το πρόγραμμα να τερματίζει αν και μόνο αν η εξίσωση έχει λύση. Βάζοντας το πρόγραμμα στη μηχανή Turing δηλαδή, αν το μηχάνημα τερματίζει με κάποια είσοδο, τότε η Διοφαντική

Εξίσωση που του ανατέθηκε έχει ακέραιη λύση. Αν δεν τερματίσει, η εξίσωση δεν έχει τέτοια λύση.
Θεωρούμε μια μηχανή H του HP, P ένα πρόγραμμα που βρίσκει αν κάποια Διοφαντική Εξίσωση έχει λύση και I μια οποιαδήποτε Διοφαντική Εξίσωση. Τότε για μηχανή H θα έχω:

$$H(P,I) = \begin{cases} \text{ναι}, & \text{αν το } P \text{ τερματίζει με είσοδο } I \\ \text{όχι}, & \text{αν το } P \text{ δεν τερματίζει με είσοδο } I. \end{cases}$$

Με αυτή την αναγωγή, αποδεικνύεται ότι αν υπήρχε αλγόριθμος ο οποίος μπορεί να αποφασίσει αν μια οποιαδήποτε Διοφαντική Εξίσωση έχει λύση ή όχι, τότε θα μπορούσε να χρησιμοποιηθεί για να λύσει το HP. Αφού έχει αποδειχθεί ότι το HP είναι μη αποφάσιμο, σημαίνει πως δεν μπορεί να υπάρχει τέτοιος αλγόριθμος για Διοφαντικές Εξισώσεις. Οπότε στην περίπτωση που το HP λυνόταν, θα λυνόταν και το Πρόβλημα Διοφαντικών Εξισώσεων.

Υποθέτοντας ότι D είναι το σύνολο των εξισώσεων που έχουν ακέραιες λύσεις ισχύει:

$$I \in D \iff H(P,I) = \text{"ναι"}$$

Αρκεί να δείξουμε ότι κάθε σετ που ανήκει στην RE κλάση ανήκει και στο σετ που ορίζει το Πρόβλημα Διοφαντικών Εξισώσεων.
Έστω K ένα σετ που ορίζεται από μια Διοφαντική Εξίσωση. Υπάρχει δηλαδή κάποιο p τέτοιο ώστε $x \in K \iff \exists y_1, \exists y_2 \ldots \exists y_k \ldots$ τ.ω $p(x, y_1, y_2, \ldots, y_k) = 0$. Έστω ότι υπάρχει

αλγόριθμος που απαντάει στο αν μια Διοφαντική Εξίσωση $p(x, y_1, y_2, \dots, y_k) = 0$ ικανοποιείται. Τότε αποφασίζοντας για την τιμή του x, μπορούμε να εφαρμόσουμε τον αλγόριθμο απόφασης για να απαντήσουμε εάν $x \in K$. Αυτό όμως σημαίνει ότι η υπόθεση ότι υπάρχει κάποιος αλγόριθμος ο οποίος αποφασίζει αν ένα πολυώνυμο ισούται με 0, υπονοεί πως κάθε RE σετ αριθμών K, αν είναι διοφαντικό, μπορεί να έχει απόφαση. Έτσι, αποδεικνύεται ότι κάθε RE πρόβλημα είναι Πρόβλημα Διοφαντικών Εξισώσεων.

Με αυτούς τους 2 ισχυρισμούς, μπορούμε να αποδείξουμε πως κάθε πρόβλημα που ανήκει στην RE μπορεί να αναχθεί στο πρόβλημα τερματισμού HP.

γ) Ένα πρόβλημα θεωρείται RE-δύσκολο (RE-hard) αν κάθε πρόβλημα της κλάσης RE μπορεί να αναχθεί σε αυτό.

Για να αποδείξουμε ότι το Πρόβλημα Καθολικού Τερματισμού είναι RE-δύσκολο, πρέπει να δείξουμε ότι κάθε πρόβλημα στην κλάση RE μπορεί να αναχθεί σε αυτό. Αυτό μπορεί να γίνει με την επίδειξη ότι η επίλυση του θα επέτρεπε την επίλυση οποιουδήποτε άλλου προβλήματος στην κλάση RE.

Στο πρόβλημα καθολικού τερματισμού, δίνεται μηχανή Turing M και ζητείται αν η M τερματίζει για όλες τις εισόδους ή όχι. Εαν καταφέρουμε να αποφασίσουμε αυτό το πρόβλημα για κάθε μηχανή Turing, τότε μπορούμε να αποφασίσουμε για κάθε πρόβλημα στην κλάση RE, καθώς κάθε τέτοιο μπορεί να αναπαρασταθεί από μια μηχανή Turing της μορφής:

$$M(P, I) = \begin{cases} \text{"ναι"}, & \text{αν το πρόγραμμα } P \text{ τερματίζει με την είσοδο } I \\ \text{"όχι"}, & \text{αν το πρόγραμμα } P \text{ δεν τερματίζει με την είσοδο } I \end{cases}$$

Όμως το πρόβλημα Καθολικού Τερματισμού είναι αποδεδειγμένα μη επιλύσιμο. Αυτό σημαίνει ότι δεν υπάρχει γενικός αλγόριθμος που μπορεί να αποφασίσει για κάθε δυνατή μηχανή Turing M αν θα τερματίσει ή όχι για όλες τις εισόδους.

Για την απόδειξη αυτού θεωρούμε ένα πρόγραμμα C:

αλγόριθμος C (X) :

```
if M(X, X) == "ναι" then
     loop forever
else
     halt
end if
```

Στο πρόγραμμα αυτό αν θέσουμε όπου X τον εαυτό του τότε ουσιαστικά δηλώνεται πως αν η μηχανή M που έχουμε θεωρήσει ως καθολική Μηχανή Turing βρει πως το πρόγραμμα C με είσοδο C τερματίζει τότε το C(C) θα τρέχει για πάντα. Αν από την άλλη τρέχει για πάντα, τότε αυτό θα τερματίζει. Αυτό αποτελεί ένα παράδοξο που αποδεικνύει την μη αποφασιμότητα του προβλήματος αυτού.

Επομένως το πρόβλημα καθολικού τερματισμού είναι RE-δύσκολο, ~~καθώς~~ και μη επιλύσιμο καθώς η επίλυση του θα επέτρεπε την επίλυση οποιουδήποτε προβλήματος στην κλάση RE, αλλά δεν υπάρχει γενικός αλγ. που

Άσκηση 2: Πολυπλοκότητα - Αναγωγές

α) Για να περιγράψουμε αναγωγή πολυωνυμικού χρόνου από το πρόβλημα UnSat, δηλαδή το συμπλήρωμα του προβλήματος Satisfiability στο πρόβλημα NoLargeClique ορίζουμε αρχικά:

- Αναγωγή Πολυωνυμικού Χρόνου
- UnSat: Το συμπλήρωμα του προβλήματος SAT, δηλαδή η απόφαση του αν μια πρόταση δεν έχει ερμηνεία.
- No Large Clique: Σε ένα γράφο G, ζητείται αν κάθε κλίκα του έχει μέγεθος το πολύ k. Πρόκειται για το συμπλήρωμα του Clique Problem.

Τώρα μπορούμε να κάνουμε αναγωγή:

Έστω πως το πρόβλημα της NoLargeClique μπορεί να διαγνωστεί σε πολυωνυμικό χρόνο.
Έστω επίσης SAT τύπος: $(a_1 \vee \ldots \vee f_1) \wedge \ldots \wedge (o_k \vee \ldots \vee d_k)$
Δημιουργείται γράφος του οποίου οι κόμβοι είναι κάθε μια μεταβλητή του τύπου και οι αρνήσεις τους, και ακμή ανάμεσα σε 2 κόμβους υπάρχει αν οι μεταβλητές που τους αντιστοιχούν είναι είτε αντίθετες μεταξύ τους, είτε ανήκουν σε διαφορετικές φράσεις και δεν είναι αντίθετες μεταξύ τους.
Ορίζουμε τον ακέραιο k ίσο με τον αριθμό των όρων και πλέον μπορούμε να τρέξουμε τον αλγόριθμο ύπαρξης κλίκας στον γράφο που παράγεται:

- Αν ο αλγόριθμος απαντήσει θετικά, δηλαδή πως υπάρχει δεν υπάρχει κλίκα μεγαλύτερη από K, τότε ο τύπος είναι ανικανοποίητος.

- Αν ο αλγόριθμος απαντήσει αρνητικά τότε ο τύπος μπορεί να ικανοποιηθεί.

Η αναγωγή αυτή είναι πολυωνυμικού χρόνου, καθώς ο γράφος G και ο ακέραιος K μπορούν να δημιουργηθούν από την αρχική λογική έκφραση σε χρόνο που είναι πολυωνυμικός στο μέγεθος της έκφρασης.

β) Έστω C μια κλάση πολυπλοκότητας και Π ένα πρόβλημα που είναι C-πλήρες. Αυτό σημαίνει ότι:

- Το Π ανήκει στην κλάση C
- κάθε άλλο πρόβλημα στην κλάση C μπορεί να αναχθεί στο Π μέσω μιας αναγωγής $\leq R$.

Το συμπλήρωμα του Π, το ονομάζουμε Π'. Θέλουμε να δείξουμε ότι το Π' είναι co-C-πλήρες.

Ξεκινάμε αναφέροντας ότι το πρόβλημα SAT είναι NP-πλήρες. Το πρόβλημα UnSat είναι το συμπλήρωμα του, και είναι coNP-πλήρες.

Από την προηγούμενη αναγωγή πολυωνυμικού χρόνου από το UnSat στο NoLargeClique, συμπεραίνουμε πως το δεύτερο είναι coNP-πλήρες και άρα το LargeClique είναι NP-hard.

Για να είναι το NoLargeClique coNP-πλήρες, πρέπει να αποδείξουμε ότι ανήκει στη κλάση coNP και ότι κάθε

άλλο πρόβλημα στην κλάση NP-hard, ανάγεται σε αυτό.
Η αναγωγή από το UnSat στο NoLargeClique, μαζί με την απόδειξη ότι το NoLargeClique ανήκει στην κλάση coNP, αρκεί για να αποδείξει ότι το NoLargeClique είναι coNP-πλήρες.

γ) Για να δείξουμε ότι αν ένα NP-πλήρες πρόβλημα ανήκει στην κλάση $NP \cap coNP$ τότε $NP = coNP$ έχουμε:

Αν υποθέσουμε ότι υπάρχει NP-πλήρες πρόβλημα Pr, το οποίο ανήκει επίσης στην κλάση $NP \cap coNP$, αυτό σημαίνει ότι για αυτό το πρόβλημα μπορεί να επαληθευτεί τόσο ένα πιθανό "ναι", όσο και ένα πιθανό "όχι", σε πολυωνυμικό χρόνο.
Δεδομένου ότι κάθε πρόβλημα NP μπορεί να αναχθεί σε ένα NP-πλήρες σε πολυωνυμικό χρόνο και αφού το δεδομένο πρόβλημα ανήκει στην $NP \cap coNP$ κλάση αυτό οδηγεί στο συμπέρασμα ότι $NP = coNP$.

δ) Για να δείξουμε πως το NAE3SAT είναι NP-πλήρες, θα πρέπει να ανάγουμε πολυωνυμικά κάποιο άλλο NP-πλήρες πρόβλημα σε αυτό. Ένα τέτοιο πρόβλημα είναι το 3SAT. Για την αναγωγή:

Έστω τύπος 3-SAT:

$$(a_1 \vee b_1 \vee c_1) \wedge (a_2 \vee b_2 \vee c_2) \wedge \dots \wedge (a_k \vee b_k \vee c_k)$$

Για να μετασχηματίσουμε τον παραπάνω τύπο σε τύπο NAE3SAT, για κάθε ένα από τα k clauses σχηματίζουμε νέα clauses και δημιουργούμε επιπλέον μεταβλητές, κανόνες για αυτά, προκειμένου να εξασφαλίσουμε ότι σε κάποια ανάθεση που ικανοποιεί το 3-SAT, δεν θα μπορούν να ικανοποιούνται λιγότερα από ένα και παραπάνω από δύο literals ανά clause. Το clause 1 για παράδειγμα μετασχηματίζεται:

$(a_1 \vee b_1 \vee e_1) \wedge (a_1 \vee c_1 \vee \bar{e}_1)$.

Με τη νέα μεταβλητή e_j εξασφαλίζεται:

→ Οποιαδήποτε ανάθεση ικανοποιούσε τα αρχικά clauses και άρα την αρχική πρόταση, θα ικανοποιεί και τον μετασχηματισμό τους.

→ Οποιαδήποτε ανάθεση που ικανοποιούσε τον μετασχηματισμό της πρότασης ικανοποιεί και την αρχική πρόταση

Ο παραπάνω μετασχηματισμός μπορεί να γίνει σε πολυωνυμικό χρόνο. Οπότε ισχύει:

$$\left\{\begin{array}{l} 3\text{-SAT} \le_p NAE3SAT \\ 3\text{-SAT} \in NP\text{-complete} \end{array}\right\} \Rightarrow NAE3SAT \in NP$$

ε) Προκειμένου να αποδείξουμε πως το πρόβλημα Επιλογής Αντιπροσώπου είναι NP-πλήρες, α.ν.δ.ο:

- Το πρόβλημα ανήκει στην κλάση NP. Αυτό αποδεικνύεται καθώς με δεδομένη κάποια λύση ένα σύνολο R, τ.ω. $|R| \le k$, μπορούμε σε πολυων. χρόνο να επαληθεύσουμε αν πράγματι κάθε ομάδα U_i έχει κάποιο αντιπρόσωπο, δηλαδή $U_i \cap R \neq \emptyset$ $i \le r$.

- Κάποιο άλλο NP-πλήρες πρόβλημα μπορεί να αναχθεί σε αυτό σε πολυων. χρόνο. Ένα κατάλληλο NP-πλήρες πρόβλημα είναι το Set Cover.

Για την αναγωγή από το πρόβλημα αυτό σε πρόβλημα Επιλογής Αντιπροσώπου θεωρούμε ότι το U στο Set Cover αντιστοιχεί στο σύνολο ατόμων U στο πρόβλημα Επιλογής Αντιπροσώπων. Θεωρούμε επίσης κάθε υποσύνολο του S στο πρόβλημα Set Cover ως μια ομάδα U_i στο πρόβλημα Επιλογής αντιπροσώπων. Αν υπάρχει λύση C για το Set Cover η οποία θα καλύπτει όλο το U με $|C| \leq k$, τότε αντίστοιχα υπάρχει λύση σύνολο R στο Επιλογής Αντιπροσώπου, το οποίο αντιπροσωπεύει κάθε ομάδα U_i

Τελικά αφού ισχύουν:

$$\left.\begin{array}{l} \text{Set Cover} \leq_P \text{Επιλογή-Αντιπροσώπων,} \\ \text{Set Cover} \in \text{NP-complete} \end{array}\right\} \Rightarrow \text{Επιλογή Αντιπροσώπων} \in \text{NP}$$

Άσκηση 3: Προσεγγιστικοί Αλγόριθμοι

α) Δίνεται αλγόριθμος προσέγγισης για το Weighted Vertex Cover και πρέπει να αποδείξουμε ότι επιτυγχάνει λόγο προσέγγισης 2, και ότι ικανοποιούνται 3 παρατηρήσεις που δίνονται:

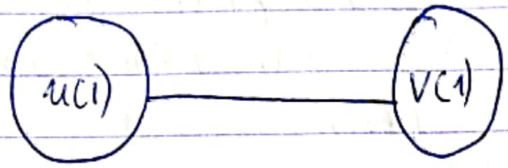
- Το C καλύπτει όλες τις ακμές: Η παρατήρηση ισχύει καθώς μέσα στο while loop εξετάζεται κάθε φορά κάποια από τις ακμές που δεν έχουν καλυφθεί, ενώ ο αλγόριθμος για κάθε ακμή που ελέγχει αποθηκεύει τουλάχιστον τη μία από τις κορυφές της στο C.

- Το συνολικό βάρος του C είναι μικρότερο ή ίσο με $2\sum_{e\in E} c(e)$: Αυτή η παρατήρηση ισχύει, καθώς κάθε ακμή e συμβάλλει στο βάρος C με το ποσό $c(e)$ και κάθε κορυφή στο C μπορεί να συμμετέχει σε το πολύ 2 ακμές. Έτσι, το συνολικό βάρος του C δεν μπορεί να ξεπερνάει το διπλάσιο του συνολικού βάρους που ανατίθεται στις ακμές.

- Το συνολικό βάρος της βέλτιστης λύσης είναι μεγαλύτερο ή ίσο με $\sum_{e\in E} c(e)$: Αυτό συμβαίνει καθώς ο αλγόριθμος θα εξετάσει μέσα στο while loop όλες τις ακμές, και για κάθε ακμή e το βάρος $c(e)$ αντιπροσωπεύει το ελάχιστο απαραίτητο που μπορεί να ανατεθεί σε μια κορυφή ώστε να καλυφθεί η ακμή. Έτσι, κάθε βέλτιστη λύση θα πρέπει τουλάχιστον να αντιστοιχίζει αυτό το βάρος σε κάθε ακμή για να καλύψει όλες τις ακμές.

Παράδειγμα Γραφήματος:

Ένα απλό γράφημα, που δείχνει ότι ο αλγόριθμος μπορεί να δώσει λύση με διπλάσιο βάρος από τη βέλτιστη λύση είναι:



Ο αλγόριθμος θα επιλέξει να βάλει και τις 2 κορυφές στο Vertex Cover C, κάνοντας το τελικό βάρος ίσο με 2. Στη βέλτιστη λύση, όμως, συμπεριλαμβάνεται μόνο ο ένας από τους 2 κόμβους, και το τελικό βάρος είναι ίσο με 1.

β) Για να δείξουμε ότι δεν υπάρχει πολυωνυμικός αλγόριθμος που να επιτυγχάνει λόγο προσέγγισης K για το TSP, εκτός αν ισχύει P=NP, θα χρησιμοποιήσουμε μια παραλλαγή της αναγωγής από το πρόβλημα Hamilton Cycle στο TSP.

→ Hamilton Cycle Problem: Δίνεται γράφος G(V,E) και ζητείται αν υπάρχει ένας κύκλος που περνά ακριβώς μια φορά από κάθε κορυφή του γράφου.

→ Δημιουργούμε μια είσοδο για το TSP από ένα γράφο του Hamilton Cycle Problem ως εξής:

- Αν 2 κορυφές είναι συνδεδεμένες στο γράφο του Hamilton Cycle, τότε η απόσταση τους στο TSP είναι 1

- Αν οι 2 κορυφές δεν είναι συνδεδεμένες, η απόσταση τους στο TSP είναι $k+1$, όπου k είναι η σταθερά προσέγγισ

Κάθε ακμή στον κύκλο Hamilton έχει μήκος 1 στο TSP. Έτσι, αν ο αρχικός γράφος έχει κάποιον Hamilton κύκλο, τότε θα υπάρχει κάποιο μονοπάτι στο γράφο TSP με συνολικό μήκος ίσο με τον αριθμό των κορυφών του γράφου.
Αν δεν υπάρχει κύκλος Hamilton, τότε οποιαδήποτε διαδρομή στο TSP θα πρέπει να περιλαμβάνει τουλάχιστον μια ακμή με μήκος $k+1$, και επομένως το συνολικό μήκος θα είναι μεγαλύτερο από τον αριθμό των κορυφών συν k.

→ Αν υπήρχε κάποιος πολυωνυμικός προσεγγιστικός αλγόριθμος με λόγο προσέγγισης k για το TSP, τότε θα μπορούσε να λυθεί και το πρόβλημα Hamilton Cycle. Δηλαδή, αν βρισκόταν κάποια διαδρομή με συνολικό μήκος μικρότερο ή ίσο με τον αριθμό των κορυφών, τότε θα υπήρχε κάποιος κύκλος Hamilton.
Έτσι, θα οδηγούμασταν στην πολυωνυμική λύση ενός NP-πλήρους προβλήματος (Hamilton Cycle), δηλαδή ενός προβλήματος στο οποίο μπορούν να αναχθούν όλα τα υπόλοιπα NP προβλήματα. Οπότε θα ίσχυε $P = NP$.