Αλγόριθμοι και Πολυπλοκότητα

3η Σειρά Γραπτών Ασκήσεων

Ιωάννης Τσαντήλας, 03120883

Άσκηση 1: Υπολογισιμότητα

Ερώτημα (α)

Προσδιορίστε το στόχο: Για να δείξουμε ότι το πρόβλημα είναι στην τάξη RE, πρέπει να αποδείξουμε ότι υπάρχει μια μηχανή Turing που σταματά με μια απάντηση "ναι" όταν υπάρχει μια ακέραια λύση και μπορεί να τρέχει επ' άπειρον αν δεν υπάρχει τέτοια λύση.

Αρχικά, δημιουργούμε μία μηχανή Turing με την δεδομένη διοφαντική εξίσωση ως είσοδο. Στη συνέχεια, δημιουργούμε όλες τις πιθανές πλειάδες ακεραίων αριθμών που θα μπορούσαν να χρησιμεύσουν ως πιθανές λύσεις της εξίσωσης. Αυτό περιλαμβάνει την απαρίθμηση πλειάδων ακεραίων αριθμών με δομημένο τρόπο (π.χ. ξεκινώντας με όλα τα μηδενικά, στη συνέχεια αυξάνοντας τις τιμές μία προς μία, λαμβάνοντας υπόψη τόσο θετικές όσο και αρνητικές τιμές).

Για κάθε παραγόμενη πλειάδα, αντικαταστούμε τους ακέραιους αριθμούς στην εξίσωση και υπολογίζουμε το αποτέλεσμα της εξίσωσης με την τρέχουσα πλειάδα ακεραίων. Εάν το αποτέλεσμα είναι μηδέν, η εξίσωση ικανοποιείται και η μηχανή απαντά «ναι» και σταματά, δείχνοντας πως υπάρχει ακέραια λύση. Διαφορετικά, συνεχίζουμε με την επόμενη πλειάδα.

Εάν δεν υπάρχει ακέραια λύση, η μηχανή Turing δεν θα βρει ποτέ πλειάδα που να ικανοποιεί την εξίσωση, με αποτέλεσμα να συνεχίζει να παράγει και να ελέγχει νέες πλειάδες επ' άπειρον.

Επομένως, η μηχανή Turing σταματά και δίνει την έξοδο «ναι» όταν βρίσκει μια ακέραια λύση, πληρώντας τα κριτήρια για να ανήκει το πρόβλημα στην κατηγορία RE. Αποδεικνύει την ύπαρξη ενός αλγορίθμου που μπορεί να ημιαποφασίζει το πρόβλημα απαριθμώντας όλες τις περιπτώσεις "ναι". Εάν δεν υπάρχει λύση, η συμπεριφορά της μηχανής που εκτελείται επ' άπειρον χωρίς να σταματάει ευθυγραμμίζεται με τον ορισμό του RE, όπου το πρόβλημα είναι ημιαποφασίσιμο.

Ερώτημα (β)

Το πρόβλημα τερματισμού (HP) ορίζεται ως ο προσδιορισμός του κατά πόσον μια δεδομένη μηχανή Turing M στην είσοδο w τερματίζει ή εκτελείται επ' άπειρον.

Αρχικά, σχεδιάζουμε μια μηχανή Turing H που προσομοιώνει την M με είσοδο w. Εάν η M σταματάει στο w, η H σταματάει επίσης και δίνει την έξοδο «ναι». Διαφορετικά (η M δεν σταματήσει στο w) η H θα συνεχίσει να προσομοιώνει την MM επ' άπειρον. Επομένως, το **HP είναι στο RE**.

Για να είναι RE-hard, κάθε πρόβλημα στο RE πρέπει να μπορεί να αναχθεί στο HP. Θα το δείξουμε αυτό ανάγοντας το πρόβλημα των Διοφαντικών Εξισώσεων (ένα γνωστό πρόβλημα RE) σε HP.

Κατασκευάζουμε μια μηχανή Turing D που απαριθμεί όλες τις πιθανές πλειάδες ακεραίων αριθμών και ελέγχει αν ικανοποιούν μία δεδομένη διοφαντική εξίσωση. Το πρόβλημα ανάγεται σε HP με το ερώτημα αν η D σταματά σε μια ειδική είσοδο που αντιπροσωπεύει τη διαδικασία απαρίθμησης:

Δημιουργούμε μία άλλη μηχανή Turing, D′. Για μια δεδομένη διοφαντική εξίσωση, η D′ έχει σχεδιαστεί ώστε να σταματά αν βρει μια λύση (μια ακέραια πλειάδα που ικανοποιεί την εξίσωση) και να τρέχει επ' άπειρον αν δεν υπάρχει λύση. Το ερώτημα εάν η D′ σταματάει μετατρέπει ουσιαστικά την απόφαση σχετικά με την ύπαρξη λύσης στη διοφαντική εξίσωση σε πρόβλημα τερματισμού της D′.

Επομένως, εφόσον το HP είναι στο RE και κάθε πρόβλημα στο RE μπορεί να αναχθεί στο HP (όπως φαίνεται από την αναγωγή από το πρόβλημα των Διοφαντικών Εξισώσεων), το HP είναι RE-πλήρες.

Ερώτημα (γ)

Το Καθολικό Πρόβλημα Τερματισμού (ΚΠΤ) αναφέρεται στο αν μια δεδομένη μηχανή Turing M τερματίζει σε κάθε πιθανή είσοδο. Για να είναι ένα πρόβλημα RE-hard, πρέπει να είναι τουλάχιστον τόσο δύσκολο όσο τα δυσκολότερα προβλήματα του RE, δηλαδή οποιοδήποτε πρόβλημα του RE μπορεί να αναχθεί σε αυτό.

Παίρνουμε μια περίπτωση του προβλήματος HP, η οποία είναι μια μηχανή Turing M και μια είσοδο w. Κατασκευάζουμε μια νέα μηχανή Turing M′ η οποία, σε οποιαδήποτε είσοδο x, προσομοιώνει την M στο w, δηλαδή αγνοεί την πραγματική της είσοδο x και εκτελεί την M μόνο στη w. Εάν η M σταματάει στο w, τότε η M′ σταματάει σε κάθε είσοδο x. Διαφορετικά (η M δεν σταματήσει στο w), η M′ δεν σταματά σε καμία είσοδο x.

Η επίλυση του ΚΠΤ για την M′ μας λέει αν το M σταματάει στο w, διότι αν το M′ έχει καθοριστεί να σταματάει σε όλες τις εισόδους, αυτό συνεπάγεται ότι το M σταματάει στο w. Έτσι, αν μπορούσαμε να λύσουμε το ΚΠΤ, θα μπορούσαμε να λύσουμε το HP, αποδεικνύοντας ότι το ΚΠΤ είναι RE-hard.

Άσκηση 2: Πολυπλοκότητα - Αναγωγές

Ερώτημα (α)

Ας ορίσουμε αρχικά τα προβλήματα:

- **UnSat**: δίνεται ένας Boolean τύπος και πρέπει να ελέγξουμε αν δεν υπάρχει τρόπος να αναθέσουμε true/false τιμές για να βγει true.
- **NoLargeClique**: δίνεται ένας γράφος και ένας αριθμός k, και πρέπει να ελέγξουμε αν δεν υπάρχει κλίκα μεγαλύτερη ή ίση από k+1 στο γράφο.

Χρησιμοποιώντας τον τύπο UnSat, κατασκευάζουμε ένα γράφημα όπου κάθε κορυφή αντιπροσωπεύει μια μεταβλητή (ή την άρνησή της). Συνδέουμε δύο κορυφές με μια ακμή αν μπορούν και οι δύο να είναι αληθείς ταυτόχρονα (δεν είναι αναιρέσεις η μία της άλλης και δεν εμφανίζονται στην ίδια πρόταση).

Ορίζουμε k τον αριθμό των ρητρών του τύπου μείον 1. Αυτό συμβαίνει επειδή μια κλίκα μεγαλύτερη από αυτό το μέγεθος θα αντιπροσώπευε ένα σύνολο τιμών που θα μπορούσαν δυνητικά να ικανοποιήσουν τον τύπο, πράγμα που αντιβαίνει στη συνθήκη UnSat.

Αν η UnSat είναι αληθής (ο τύπος είναι μη ικανοποιήσιμος) τότε δεν υπάρχει σύνολο τιμών που να μπορεί να ικανοποιήσει όλες τις προτάσεις. Επομένως, στο γράφημα, δεν μπορεί να υπάρχει κλίκα μεγαλύτερη από k, γιατί μία τέτοια κλίκα θα σήμαινε ικανοποιητική ανάθεση των λογικών.

Ο κατασκευασμένος γράφος και το σύνολο k αποτελούν μια περίπτωση του προβλήματος NoLargeClique. Εάν το πρόβλημα UnSat είναι αληθές (ο τύπος είναι μη ικανοποιήσιμος), τότε το πρόβλημα NoLargeClique θα είναι επίσης αληθές (ο γράφος δεν περιέχει μεγάλη κλίκα), ολοκληρώνοντας την αναγωγή.

Ερώτημα (β)

Το πρόβλημα UnSat είναι το συμπλήρωμα του προβλήματος SAT, το οποίο είναι γνωστό ότι είναι NP-πλήρες. Εφόσον το SAT είναι NP-πλήρες, το **UnSat είναι στο coNP-πλήρες** επειδή το συμπλήρωμα ενός NP-πλήρους προβλήματος είναι coNP-πλήρες.

Από το μέρος (α), γνωρίζουμε ότι υπάρχει αναγωγή σε πολυωνυμικό χρόνο από το UnSat στο NoLargeClique. Έτσι, αφού το UnSat είναι coNP-πλήρες, τότε το NoLargeClique είναι τουλάχιστον coNP.

Ωστόσο, για να ολοκληρωθεί το επιχείρημα ότι η NoLargeClique είναι coNP-πλήρης, είναι κρίσιμο να δείξουμε ότι το NoLargeClique είναι coNP. Μπορούμε να το αποδείξουμε εάν το συμπλήρωμα του (ο προσδιορισμός του κατά πόσον ένας γράφος περιέχει μια κλίκα μεγαλύτερη από k) είναι NP. Πράγματι, αφού μπορεί κανείς να μαντέψει και να επαληθεύσει μη-ντετερμινιστικά μια τέτοια κλίκα σε πολυωνυμικό χρόνο.

Ερώτημα (γ)

Η ύπαρξη ενός NP-πλήρους προβλήματος στο NP ∩ coNP σημαίνει ότι υπάρχει ένα πρόβλημα που είναι εξίσου δύσκολο με τα δυσκολότερα προβλήματα στο NP (αφού είναι NP-πλήρες), το οποίο έχει επίσης το συμπλήρωμά του στο NP (αφού είναι στο coNP).

Αυτό θα σήμαινε ότι για αυτό το πρόβλημα, και επομένως για όλα τα προβλήματα που μπορούν να αναχθούν σε αυτό, οι περιπτώσεις «όχι» μπορούν να επαληθευτούν όπως και οι περιπτώσεις "ναι", σε πολυωνυμικό χρόνο.

Εάν μπορούμε να επαληθεύσουμε τόσο τις περιπτώσεις "ναι" όσο και τις περιπτώσεις "όχι" σε πολυωνυμικό χρόνο για ένα πρόβλημα NP-πλήρες (και επομένως για όλα τα προβλήματα στο NP, αφού τα NP-πλήρη προβλήματα είναι τόσο δύσκολα όσο και κάθε πρόβλημα στο NP), αυτό σημαίνει ότι η διάκριση μεταξύ NP και coNP καταρρέει.

Επομένως, η ύπαρξη ενός τέτοιου προβλήματος συνεπάγεται ότι οι κλάσεις NP και coNP είναι ισοδύναμες, οδηγώντας στο συμπέρασμα NP = coNP.

Ερώτημα (δ)

Για να αποδείξουμε το ζητούμενο, πρέπει να δείξουμε ότι ένα ήδη γνωστό NP-πλήρες πρόβλημα μπορεί να αναχθεί σε NAE3SAT σε πολυωνυμικό χρόνο. Επιλέγουμε το 3-SAT για το σκοπό αυτό:

- **Πρόβλημα 3-SAT**: Δεδομένου ενός Boolean τύπου σε CNF (Conjunctive Normal Form) όπου κάθε όρος έχει ακριβώς τρία literals, προσδιορίστε αν υπάρχει μια ανάθεση αλήθειας στις μεταβλητές που κάνει τον τύπο αληθή.

Για κάθε πρόταση στον 3-SAT τύπο, προσθέτουμε μια επιπλέον μεταβλητή **x** στην πρόταση και προσθέτουμε επίσης δύο επιπλέον προτάσεις που περιέχουν την **x** και την άρνησή της **¬x**. Αυτό διασφαλίζει ότι το **x** δεν μπορεί να είναι το ίδιο (είτε όλα αληθή είτε όλα ψευδή) σε όλες τις προτάσεις, επιβάλλοντας τη συνθήκη NAE χωρίς να αλλάζει η ικανοποιησιμότητα του αρχικού τύπου.

Ο τροποποιημένος τύπος είναι τώρα μια έγκυρη περίπτωση το NAE3SAT. Εάν ο αρχικός τύπος 3-SAT είναι ικανοποιήσιμος, τότε υπάρχει μια ανάθεση αλήθειας που ικανοποιεί τη συνθήκη NAE για τον μετασχηματισμένο τύπο. Αντιστρόφως, αν ο μετασχηματισμένος τύπος είναι ικανοποιήσιμος σύμφωνα με τη συνθήκη NAE, ο αρχικός τύπος 3-SAT είναι επίσης ικανοποιήσιμος.

Έχουμε δείξει λοιπόν ότι το NAE3SAT είναι NP επειδή μπορούμε να επαληθεύσουμε μια λύση σε πολυωνυμικό χρόνο. Η αναγωγή σε πολυωνυμικό χρόνο από το 3-SAT (ένα NP-πλήρες πρόβλημα) στο NAE3SAT αποδεικνύει ότι η επίλυση του NAE3SAT είναι τουλάχιστον εξίσου δύσκολη με την επίλυση του 3-SAT, άρα το NAE3SAT είναι NP-hard. Συνδυάζοντας τα παραπάνω σημεία, το NAE3SAT είναι NP-πλήρες επειδή είναι τόσο NP όσο και NP-hard.

Ερώτημα (ε)

Για να αποδείξουμε ότι ένα πρόβλημα ανήκει στο NP, πρέπει να αποδείξουμε ότι αν μας δοθεί μια προτεινόμενη λύση, μπορεί να επαληθευτεί αν είναι σωστή σε πολυωνυμικό χρόνο. Δεδομένου ενός υποσυνόλου R ως προτεινόμενη λύση, μπορούμε να ελέγξουμε σε πολυωνυμικό χρόνο αν το R περιέχει τουλάχιστον ένα μέλος από κάθε ομάδα U_i και αν $|R| \leq k$, επαληθεύοντας τη λύση.

Για να διαπιστώσουμε την NP-πληρότητα, πρέπει να αναγάγουμε ένα γνωστό NP-πλήρες πρόβλημα στο πρόβλημα Agent Selection σε πολυωνυμικό χρόνο. Ένας κατάλληλος υποψήφιος για το σκοπό αυτό είναι το πρόβλημα Set Cover, το οποίο είναι γνωστό ότι είναι NP-πλήρες:

- **Set Cover**: Δεδομένου ενός σύμπαντος U, μιας συλλογής υποσυνόλων S του U και ενός ακέραιου k, ελέγχουμε αν υπάρχει μια συλλογή από k ή λιγότερα υποσύνολα στο S που καλύπτουν μαζί κάθε στοιχείο του U.

Αρχικά, αντιστοιχίζουμε το σύμπαν U και τη συλλογή υποσυνόλων S από το Set Cover απευθείας στο σύνολο των ατόμων και των κοινωνικών ομάδων στο Agent Selection, αντίστοιχα. Έπειτα, χρησιμοποιούμε το ίδιο k για το όριο μεγέθους του αντιπροσωπευτικού συνόλου στο Agent Selection με το όριο του αριθμού των υποσυνόλων για την κάλυψη του U στο Set Cover.

Μια λύση στο πρόβλημα Set Cover (επιλογή k ή λιγότερων υποσυνόλων που καλύπτουν το UU) αντιστοιχεί άμεσα σε μια λύση στο πρόβλημα Agent Selection (ένα σύνολο k ή λιγότερων ατόμων που αντιπροσωπεύουν όλες τις ομάδες). Επομένως το Agent Selection είναι:

- **NP**: Αποδεικνύεται δείχνοντας ότι δεδομένου ενός προτεινόμενου αντιπροσωπευτικού συνόλου R, μπορούμε να ελέγξουμε σε πολυωνυμικό χρόνο αν καλύπτει όλες τις ομάδες και αν σέβεται το όριο μεγέθους k.
- **NP-hard**: Αποδεικνύεται μέσω της αναγωγής σε πολυωνυμικό χρόνο από το Set Cover, υποδεικνύοντας ότι η επίλυση του προβλήματος επιλογής πρακτόρων είναι τουλάχιστον τόσο δύσκολη όσο η επίλυση οποιουδήποτε προβλήματος στο NP.
- **NP-complete**: Συνδυάζοντας τα παραπάνω, το πρόβλημα Επιλογής Πράκτορα είναι NP-πλήρες επειδή βρίσκεται στο NP και είναι NP-hard.

Άσκηση 3: Προσεγγιστικοί Αλγόριθμοι - Vertex Cover -TSP

<u>Ερώτημα (α)</u>

- *Μέρος 1: Απόδειξη Αλγορίθμου*

Ο αλγόριθμος βρίσκει ένα Vertex Cover ενός σταθμισμένου γραφήματος G(V,E), όπου κάθε κορυφή v ϵ V έχει θετικό βάρος.

Ο στόχος του αλγορίθμου είναι να βρει ένα υποσύνολο κορυφών C ⊆ V τέτοιο ώστε κάθε ακμή e ϵ E να καλύπτεται από τουλάχιστον μία κορυφή στο C και το συνολικό βάρος του C να ελαχιστοποιείται. Αρχικά, το C είναι κενό, κάθε κορυφή v έχει ένα δοκιμαστικό βάρος t(v)=w(v), και κάθε ακμή e έχει μηδενικό κόστος, c(e)=0.

Στην συνέχεια, επαναλαμβάνει τις ακμές, επιλέγοντας μια ακμή e={u, v} χωρίς κάλυψη σε κάθε βήμα. Μειώνει τα δοκιμαστικά βάρη t(u) και t(v) κατά $\min\{t(u), t(v)\}$, επιλέγοντας ουσιαστικά την κορυφή (ή τις κορυφές) με το μικρότερο εναπομείναν βάρος για την κάλυψη της ακμής. Αυτό το ποσό μείωσης προστίθεται στο c(e), και αν κάποιο από τα t(u), t(v) μηδενιστεί, η αντίστοιχη κορυφή προστίθεται στο C.

Συμπερασματικά:

1. **Το C καλύπτει όλες τις ακμές**: ο αλγόριθμος εξασφαλίζει ότι κάθε ακμή καλύπτεται πριν τερματίσει. Για κάθε ακμή e={u, v} που δεν καλύπτεται, τουλάχιστον μία από τις u, v θα μηδενίσει το δοκιμαστικό της βάρος και θα προστεθεί στο C, εξασφαλίζοντας την κάλυψη.
2. **Συνολικό βάρος C $\leq 2 \cdot \sum_{e\, \epsilon\, E} c(e)$**: για κάθε ακμή e={u, v}, ο αλγόριθμος αναθέτει c(e) ως το $\min\{t(u), t(v)\}$ τη στιγμή της κάλυψής της. Δεδομένου ότι το c(e) υπολογίζεται δύο φορές στο συνολικό βάρος του C (μία φορά για κάθε κορυφή), το συνολικό βάρος του C μπορεί να θεωρηθεί ως το άθροισμα των βαρών των κορυφών τη στιγμή που καλύπτουν τις ακμές τους. Ωστόσο, καμία κορυφή δεν πληρώνει περισσότερο από το αρχικό της βάρος, και η συνεισφορά κάθε ακμής e στην κάλυψη των κορυφών είναι το πολύ δύο φορές c(e), ικανοποιώντας την προσέγγιση 2.
3. **Συνολικό βάρος βέλτιστης λύσης $\geq \sum_{e\, \epsilon\, E} c(e)$:** η βέλτιστη λύση πρέπει να επιλέξει τουλάχιστον μία κορυφή από κάθε ακμή e και το κόστος που ανατίθεται στην e, c(e), είναι το ελάχιστο πρόσθετο βάρος που απαιτείται για την κάλυψη της e στο βήμα που εξετάζεται. Έτσι, το άθροισμα των τιμών c(e) είναι ένα κατώτερο όριο για το συνολικό βάρος οποιασδήποτε κάλυψης κορυφής, συμπεριλαμβανομένης της βέλτιστης.

- *Μέρος 2: Ένα απλό παράδειγμα*

Θεωρούμε ένα απλό γράφημα αποτελούμενο από μόνο δύο κορυφές που συνδέονται με μία μόνο ακμή. Αναθέτουμε βάρη στις u και v έτσι ώστε $w(u) = w(v) = 1$.

Δεδομένου ότι υπάρχει μόνο μία ακμή, ο αλγόριθμος θα καλύψει αυτή την ακμή μειώνοντας τα δοκιμαστικά βάρη τόσο του u όσο και του v μέχρι ένα από αυτά να μηδενιστεί. Θα επιλέξει και τις δύο, επειδή μειώνει τα βάρη και των δύο ταυτόχρονα (εάν έχουν το ίδιο βάρος).

Αντίθετα, η βέλτιστη λύση είναι είτε η u είτε η v, όχι και οι δύο. Αυτό οδηγεί σε ένα συνολικό βέλτιστο βάρος 1. Επομένως, ο αλγόριθμος υπολογίζει ένα Vertex Cover με συνολικό βάρος διπλάσιο από αυτό της βέλτιστης λύσης.

<u>Ερώτημα (β)</u>

Πρόβλημα HC: Δεδομένου ενός γραφήματος G(V,E), προσδιορίστε αν υπάρχει ένας κύκλος που επισκέπτεται κάθε κορυφή ακριβώς μία φορά και επιστρέφει στην αρχική κορυφή.

Πρόβλημα TSP: Δεδομένου ενός πλήρους γράφου $G'(V',E')$ όπου κάθε ακμή έχει ένα βάρος, βρείτε τη συντομότερη δυνατή διαδρομή που επισκέπτεται κάθε κορυφή ακριβώς μία φορά και επιστρέφει στην αρχική κορυφή.

Η κλασική αναγωγή από το πρόβλημα HC σε TSP περιλαμβάνει τη δημιουργία μιας περίπτωσης TSP από ένα δεδομένο γράφημα G για το πρόβλημα HC. Σε κάθε ακμή του G που υπάρχει στην E ανατίθεται βάρος 1, και σε κάθε ακμή που δεν υπάρχει στην E (για να γίνει το G' πλήρες) ανατίθεται απαγορευτικά μεγάλο βάρος.

Για να αποδείξουμε ότι δεν υπάρχει αλγόριθμος προσέγγισης k-πολυωνυμικού χρόνου για το TSP εκτός αν P=NP, προσαρμόζουμε τα βάρη των ακμών στην αναγωγή ως εξής:

- Αν υπάρχει ακμή στο G, θέτουμε το βάρος της ίσο με 1.
- Διαφορετικά, αντί να την θέσουμε απαγορευτικά μεγάλη, της θέτουμε βάρος λίγο μεγαλύτερο από k φορές τον αριθμό των κορυφών στο G, δηλαδή $k \cdot |V| + 1$.

Εάν το G περιέχει έναν HC, το βάρος της βέλτιστης περιήγησης TSP στην G′ είναι ακριβώς |V|, αφού ο κύκλος θα χρησιμοποιούσε μόνο ακμές βάρους 1. Διαφορετικά, οποιαδήποτε περιήγηση στο G′ πρέπει να χρησιμοποιεί τουλάχιστον μία ακμή που δεν ανήκει στο G, καθιστώντας το βάρος της τουλάχιστον $k \cdot |V| + 1$.

Ωστόσο, μια k-προσεγγιστική λύση θα έχει συνολικό βάρος όχι μεγαλύτερο από $k \cdot |V|$ αν και μόνο αν υπάρχει ένας HC στο G. Αν δεν υπάρχει HC, το βάρος οποιασδήποτε k-προσεγγιστικής περιήγησης πρέπει να είναι σημαντικά υψηλότερο.

Αν υπάρχει ένας αλγόριθμος προσέγγισης k-πολυωνυμικού χρόνου για το TSP, θα μπορούσαμε να τον χρησιμοποιήσουμε για να αποφασίσουμε αν οποιοσδήποτε δεδομένος γράφος G έχει έναν HC εξετάζοντας το βάρος της προσεγγιστικής λύσης της TSP. Αυτό όμως συνεπάγεται ότι P=NP, άτοπο.