Εθνικό Μετσόβιο Πολυτεχνείο
Σχολή Ηλεκτρολόγων Μηχανικών & Μηχανικών Υπολογιστών
Τομέας Τεχνολογίας Πληροφορικής και Υπολογιστών
Εργαστήριο Λογικής και Επιστήμης Υπολογισμών

Δομική Πολυπλοκότητα

Εαρινό Εξάμηνο 2011-2012

Σειρά Ασκήσεων

Η παράδοση των ασκήσεων:
-είναι απαραίτητη προϋπόθεση για προβιβάσιμο βαθμό στο μάθημα.
-γίνεται σε εκτυπωμένη μορφή, ή ηλεκτρονικά στο antony.ant1985@gmail.com.
-πρέπει να γίνει μέχρι το τέλος του εξαμήνου (19/7/2012).

1. (α′) Ένα μη-ντετερμινιστικό κύκλωμα C έχει δύο εισόδους $x = x_1x_2\cdots x_n$ και $y = y_1y_2\cdots y_m$. Το κύκλωμα C αποδέχεται το x αν και μόνο αν $\exists y\ C(x, y) = 1$. Δείξτε ότι κάθε γλώσσα στο **MA** έχει μη-ντετερμινιστικά κυκλώματα πολυωνυμικού μεγέθους. (*6 μον.*)

   (β′) Δείξτε ότι $\mathcal{BP} \cdot co\mathbf{NP} = co\mathbf{AM}$. (*4 μον.*)

2. Δείξτε ότι δοθέντος ενός verifier V και ενός input x, μπορούμε να υπολογίσουμε την μέγιστη πιθανότητα (πάνω σε όλες τις δυνατές στρατηγικές του prover P^*) του ενδεχομένου $V(x) = 1$ (δηλ. ο verifier V να αποδεχθεί την είσοδο x μετά από αλληλεπίδραση με τον P^*). (*9 μον.*)
   Επίσης, μπορείτε να αποδείξετε ότι ο βέλτιστος αυτός prover είναι ντετερμινιστικός. Ποιο σημαντικό αποτέλεσμα προκύπτει από αυτή την διαπίστωση;(*3 μον.*)

(*Interactive Proof Systems*)

3. (α′) Ορίστε τις κλάσεις $\#\mathbf{P}$, $\oplus\mathbf{P}$, $\mathbf{C}_=\mathbf{P}$ και **SPP**. (*2 μον.*)

   (β′) Η κλάση $Gap\mathbf{P}$ ορίζεται ως το σύνολο των συναρτήσεων f για τις οποίες υπάρχει μία μη-ντετερμινιστική TM M τέτοια ώστε για κάθε $x \in \Sigma^*$:

$$f(x) = \Delta M(x) = \#M(x) - \#\overline{M}(x) = \#acc(x) - \#rej(x)$$

   όπου $\#acc(x)$ και $\#rej(x)$ το πλήθος των μονοπατιών που αποδέχονται και απορρίπτουν, αντίστοιχα.

   i. Δείξτε ότι αν $f \in Gap\mathbf{P}$, τότε και $-f \in Gap\mathbf{P}$.

   ii. Χρησιμοποιήστε μια συνάρτηση της κλάσης $Gap\mathbf{P}$ για να ορίσετε εναλλακτικά τις κλάσεις **PP**, **SPP** και $\mathbf{C}_=\mathbf{P}$.

   (*4 μον.*)

   (γ′) Το βασικό μειονέκτημα της $\#\mathbf{P}$ είναι ότι περιλαμβάνει μόνο μη-αρνητικές συναρτήσεις, και έτσι δεν είναι κλειστή ως προς την αφαίρεση. Η $Gap\mathbf{P}$ περιλαμβάνει και συναρτήσεις που παίρνουν αρνητικές τιμές.

   i. Δείξτε ότι κάθε συνάρτηση $f \in \mathbf{FP}$ ανήκει στην $Gap\mathbf{P}$.

   ii. Δείξτε ότι αν $f \in Gap\mathbf{P}$ τότε μπορεί να γραφτεί ως διαφορά δύο συναρτήσεων που ανήκουν στην $\#\mathbf{P}$.

   (*4 μον.*)

(*Counting Complexity*)

4. (α′) Δείξτε ότι $\mathbf{PCP}(\log n, 1) \subseteq \mathbf{NP}$. (*2 μον.*)

   (β′) Δείξτε ότι $\mathbf{PCP}(\log n, 1) = \mathbf{PCP}(\log n, poly(n))$ (*2 μον.*)

   (γ′) Δείξτε ότι αν $\mathtt{SAT} \in \mathbf{PCP}(r(n), 1)$, για $r(n) = o(\log n)$, τότε $\mathbf{P} = \mathbf{NP}$. (*8 μον.*)
   Ποιά σημαντική ένδειξη μας δίνει αυτό το αποτέλεσμα; (*2 μον.*)

(*Probabilistically Checkable Proofs*)

5. Χρησιμοποιείστε το αποτέλεσμα του Feige:

> ''Δεν υπάρχει $(1-\delta)\ln n$-προσεγγιστικός αλγόριθμος για το `UNWEIGHTED SET COVER` (όπου $\delta > 0$ δοθείσα σταθερά), εκτός αν $\mathbf{NP} \subseteq \mathbf{DTIME}(n^{\mathcal{O}(\log\log n)})$.''

για να αποδείξετε ότι δεν υπάρχει $(1 - \frac{1}{e} + \varepsilon)$-προσεγγιστικός αλγόριθμος για το `UNWEIGHTED MAXIMUM COVERAGE`, εκτός αν $\mathbf{NP} \subseteq \mathbf{DTIME}(n^{\mathcal{O}(\log\log n)})$ (όπου $\varepsilon > 0$ δοθείσα σταθερά). (*8 μον.*)
Υπενθυμίζουμε ότι στο πρόβλημα `UNWEIGHTED MAXIMUM COVERAGE` δίνεται μία οικογένεια συνόλων, που αποτελούν υποσύνολα ενός ''σύμπαντος '' $\mathcal{U}$, και ένα budget $k > 0$, και ζητείται να καλύψουμε όσα περισσότερα στοιχεία του $\mathcal{U}$ γίνεται, χρησιμοποιώντας το πολύ k σύνολα.

(*Inapproximability*)

6. (*Η επόμενη παρατήρηση οφείλεται στον Avi Wigderson:*) Δείξτε ότι *δεν* υπάρχει natural proof για το ότι το πρόβλημα του Διακριτού Λογαρίθμου χρειάζεται κυκλώματα μεγέθους 2^{n^ε}, για κάποια σταθερά $\varepsilon > 0$. (*8 μον.*)
Υπενθυμίζουμε ότι στο πρόβλημα του Διακριτού Λογαρίθμου μας δίνεται ένας πρώτος p, και $g, y \in \mathbb{Z}_p^*$, όπου $g \neq 1$, και ζητείται $x \in \mathbb{Z}_p^*$ τέτοιο ώστε $y = g^x \mod p$.

(*Natural Proofs*)

7. Έστω $G : \{0,1\}^* \to \{0,1\}^*$ συνάρτηση υπολογίσιμη σε πολυωνυμικό χρόνο, τέτοια ώστε $|G(x)| = S(|x|)$ για κάθε $x \in \{0,1\}^*$, όπου $S : \mathbb{N} \to \mathbb{N}$ (θυμηθείτε ότι η S ονομάζεται stretch function). Η G ονομάζεται ''μη-προβλέψιμη'' αν για κάθε $B \in \mathbf{BPP}$ υπάρχει μία αμελητέα[1] συνάρτηση $\varepsilon : \mathbb{N} \to [0,1]$ τέτοια ώστε

$$\mathbf{Pr}[B(1^n, y_1, y_2, \ldots, y_{i-1}) = y_i] \leq \frac{1}{2} + \varepsilon(n)$$

δηλαδή, το να προβλέψουμε το i-οστό bit δεδομένων των $i-1$ προηγούμενων, είναι υπολογιστικά δύσκολο για κάθε πιθανοτικό αλγόριθμο πολυωνυμικού χρόνου.
Δείξτε ότι μία συνάρτηση G που είναι pseudorandom generator (με stretch function S, και ε-ψευδοτυχαία έναντι σε κάθε $\mathbf{BPP}$ adversary), είναι και μη-προβλέψιμη. (*10 μον.*)

(*Pseudorandom Generators*)

8. Έστω $\mathbf{prRP}$ η κλάση των promise problems που αντιστοιχεί στην κλάση $\mathbf{RP}$, δηλαδή είναι η κλάση των γλωσσών L που αποφασίζονται από μία Πιθανοτική T.M. $M(x,r)$ τέτοια ώστε αν $x \in \Pi_{YES}$ τότε $\mathbf{Pr}[M(x,r) \text{ accepts}] \geq 2/3$, και αν $x \in \Pi_{NO}$ τότε $\mathbf{Pr}[M(x,r) \text{ accepts}] = 0$.
Δείξτε ότι αν $\mathbf{P} = \mathbf{prRP}$, τότε $\mathbf{P} = \mathbf{BPP}$. (*8 μον.*)

(*Promise Problems*)

9. Έχετε τρία νομίσματα από τα οποία τα δύο είναι κίβδηλα. Ένα κίβδηλο επιστρέφει γράμματα με πιθανότητα $0 < \alpha < 1/2$, ενώ το άλλο επιστρέφει γράμματα με πιθανότητα $1 - \alpha$. Το τρίτο νόμισμα είναι γνήσιο και επιστρέφει γράμματα με πιθανότητα $1/2$. Τα νομίσματα είναι αδιαχώριστα ''με το μάτι''.

   (α′) Δώστε τον τύπο της στατιστικής απόστασης μεταξύ δύο πιθανοτικών κατανομών. (*4 μον.*)

   (β′) Περιγράψτε έναν αλγόριθμο που τερματίζει πάντοτε και που χρησιμοποιεί τα τρία νομίσματα με τις ελάχιστες δυνατές ρίψεις ώστε να επιστρέφει έξοδο που να είναι *ομοιόμορφα κατανεμημένη* στο $\{0,1\}$. Αν δεν μπορείτε να βρείτε τέτοιο αλγόριθμο δώστε έναν αλγόριθμο που τερματίζει πάντοτε και πλησιάζει την ομοιόμορφη κατανομή όσο καλύτερα γίνεται (με την έννοια της στατιστικής απόστασης). Σε κάθε περίπτωση δώστε πλήρη αιτιολόγηση της απάντησής σας. (*6 μον.*)
   (*Σημείωση: ο αλγόριθμός σας δεν πρέπει να χρησιμοποιεί άλλη πηγή τυχαιότητας πέρα από τα τρία νομίσματα.*)

(*Randomness Extractors*)

[1] Μία συνάρτηση $\varepsilon : \mathbb{N} \to [0,1]$ λέγεται αμελητέα αν $\varepsilon(n) = n^{-\omega(1)}$.

10. (*Μείωση σφάλματος* **RP** *αλγορίθμων με χρήση expanders*:) Έστω $A \in \mathbf{RP}$. Ως γνωστόν, μπορούμε να μειώσουμε την πιθανότητα σφάλματος του A επαναλαμβάνοντας τον αλγόριθμο λ φορές, το οποίο μειώνει εκθετικά την πιθανότητα λάθους, αλλά αυξάνει τα random bits που χρησιμοποιούμε κατά ένα παράγοντα λ. Θα χρησιμοποιήσουμε μια εναλλακτική μέθοδο για την μείωση του σφάλματος, που χρησιμοποιεί έναν (n, d, h)-expander που δίνεται σε explicit μορφή:

Έστω ότι ο αλγόριθμος A χρησιμοποιεί k random bits κατά την εκτέλεσή του, και έστω $G = (V, E)$ το (n, d, h)-expander γράφημα, με $V = \{0,1\}^k$, και $h << \frac{1}{3}$ (όπου $\frac{1}{3}$ το άνω φράγμα στο σφάλμα του **RP** αλγορίθμου, όπως αναφέρεται στον κλασσικό ορισμό). Έστω ο αλγόριθμος A':

```
(1) Pick a vertex v_0 ∈ V uniformly at random.
(2) Start from it a random walk (u_0, u_1, ..., u_t) of length t.
(3) Return ⋀_{i=0}^{t} A(x, u_i).
```

(α′) Υπολογίστε την πιθανότητα λάθους του αλγορίθμου A' και συγκρίνετέ την με αυτή του A. (*5 μον.*)

(β′) Υπολογίστε τον αριθμό των random bits που χρησιμοποιεί ο A' και συγκρίνετέ τον με αυτόν του A. (*5 μον.*)

(*Expander Graphs*)