Pseudorandomness

Pseudorandom Generators - Derandomisation

Παναγιώτης Γροντάς

$\mu\Pi\lambda\forall$

17.05.2012, 24.05.2012

Κλάσεις Πολυπλοκότητας

Θα χρησιμοποιήσουμε τις εξής κλάσεις πολυπλοκότητας:

P

BPP $= \{L | x \in L \Rightarrow Pr_{r \in \{0,1\}^{p(|x|)}}[M(x,r) = L(x)] \geq \frac{2}{3}\}$

QuasiP $= \cup_{c \in \mathbb{N}} DTIME(2^{(logn)^c})$

SUBEXP $= \cap_{\epsilon > 0} DTIME(2^{n^\epsilon})$ Οι λογάριθμοι του χρόνου εκτέλεσης είναι μικρότεροι από κάθε πολυώνυμο.

E $= DTIME(2^{O(n)})$ Exponential Time With Linear Exponent

EXP $= \cup_{c \in \mathbb{N}} DTIME(2^{n^c})$ Exponential Time

Τυχαιότητα

Τρεις Θεωρίες Τυχαιότητας:

- **Shannon**: Εντροπία και τυχαιότητα, Μέγιστη Περιεχόμενη Πληροφορία - Ομοιόμορφη Κατανομή $U_n : (x \in \{0,1\}^n \to 2^{-n})$
- **Kolmogorov**, **Chaitin**: Αλγοριθμική Τυχαιότητα - Το μέγεθος του μικρότερου προγράμματος που μπορεί να παράγει $(x \in \{0,1\}^n)$
- **Blum**, **Goldwasser**, **Micali**: Σχετικιστική Τυχαιότητα - Όχι εσωτερική ιδιότητα αντικειμένων, αλλά εξαρτάται από τον παρατηρητή
  - Δύο αντικείμενα είναι ίδια, αν 'φαίνονται' ίδια.
  - Φαίνονται ίδια: Δεν μπορούν να διαχωριστούν από αποδοτική διαδικασία.
  - Πλεονέκτημα: Μπορούμε να **ενισχύσουμε** την τυχαιότητα.

Τυχαιότητα και CS - (1)

Πιθανοτικοί Αλγόριθμοι: Αποδοτικές Λύσεις Σε Δύσκολα Προβλήματα:

- **Primality** Εύρεση 'πιστοποιητικών' ότι κάποιος αριθμός είναι σύνθετος.
- **Ισότητα Πολυωνύμων** Είναι δύο πολυώνυμα ίδια;
- **Reachability** Είναι συνδεδεμένες δύο κορυφές ενός γραφήματος (Random Walks)
- **Approximate Counting** Προσέγγιση πλήθους λύσεων σε συνδυαστικά προβλήματα.

Τυχαιότητα και CS - (2)

Βασικό Ερώτημα:

- $BPP =^{?} P$
- Είναι οι πιθανοτικοί αλγόριθμοι εν γένει πιο δυνατοί από τους ντετερμινιστικούς
- ή
- μπορούμε να μετατρέψουμε έναν **αποδοτικό** πιθανοτικό αλγόριθμο σε **αποδοτικό** ντετερμινιστικό (*derandomisation*).

Trivial Derandomisation

$BPP \subseteq EXP$

Έστω $A(x, r)$ ένας πιθανοτικός αλγόριθμος όπου:

- για είσοδο x μήκους n
- χρησιμοποιεί τυχαίο r μήκους m και
- τρέχει σε χρόνο $T(n)$.

Μπορεί να προσομοιωθεί από τον εξής ντετερμινιστικό αλγόριθμο:

- Για είσοδο x
- Δημιούργησε όλα τα δυνατά r (2^m)
- Για κάθε r, υπολόγισε το $A(x, r)$ σε χρόνο $T(n)$.
- Αν $\#YES > \#NO$ output "YES" αλλιώς output "NO".

Η προσομοίωση γίνεται σε εκθετικό χρόνο $2^m T(n)$

$$...\text{εκτός αν } m = logn...$$

Μπορούμε καλύτερα; ΝΑΙ (υπο συνθήκες) (Blum, GoldWasser, Micali, Yao)

Conditional Derandomisation: *Αν η συνθήκη X ισχύει τότε ο PPT αλγόριθμος A μπορεί να μετατραπεί σε ντετερμινιστικό που τρέχει σε χρόνο Y*

X Δεν υπάρχει PT αλγόριθμος που βρίσκει την παραγοντοποίηση ενός ακεραίου

Y 2^{n^ϵ} $\forall \epsilon > 0$

90s
- Η συνθήκη X να είναι εύλογη.
- Το Υ να είναι πολυωνυμικός.

Conditional Derandomisation

- Μία υπόθεση X για την χειρότερη περίπτωση ενός αλγόριθμου συνεπάγεται μία ισχυρότερη υπόθεση για την μέση περίπτωση (*hardness amplification - (Impagliazzo, Wigderson)*)
- Μπορώ από την μέση περίπτωση να κατασκευάσω ψευδοτυχαία γεννήτρια (*pseudorandom generator*).
- Η ψευδοτυχαία γεννήτρια θα βοηθήσει στο derandomisation. (*Nisan, Wigderson*)

Φυσικά αν κάποια στιγμή αποδειχθεί η υπόθεση X, τότε θα έχουμε πλήρη derandomisation (P=BPP).

Ψευδοτυχαιότητα

- Σχετικιστική Θεώρηση Τυχαιότητας
  - **Ψευδοτυχαία Κατανομή**: Δεν μπορεί να διαχωριστεί αποδοτικά από την Ομοιόμορφη Κατανομή U_n.
  - **Αποδοτικός Διαχωρισμός**: Probabilistic Polynomial Time Algorithm (PPT)
- Αρχικές Εφαρμογές - Κρυπτογραφία
  - Εξαγωγή Δεδομένων Από φυσική διαδικασια που θεωρείται τυχαία.
  - Ενίσχυση της αρχικής τυχαιότητας (seed).
  - Το αποτέλεσμα πρέπει να παραμένει (ψευδο)τυχαίο.

Computational Indistinguishability

Ορισμός

Οικογένεια κατανομών $\{X_n\}$: σύνολο τυχαίων μεταβλητών

Ορισμός

Δύο οικογένειες κατανομών $\{X_n\}, \{Y_n\}$ είναι υπολογιστικά μη διαχωρίσιμες αν:

- $\forall PPT\ A, \forall positive\ polynomial\ p,\ \forall n > n_0$:
- $|Pr_{x\sim X_n}[A(x) = 1] - Pr_{y\sim Y_n}[A(y) = 1]| < \frac{1}{p(n)}$

όπου: $Pr_{x\sim X_n}[A(x) = 1]$ η πιθανότητα να επιβεβαίωσει ο αλγόριθμος ότι $x \sim X_n$

Pseudorandom Generators

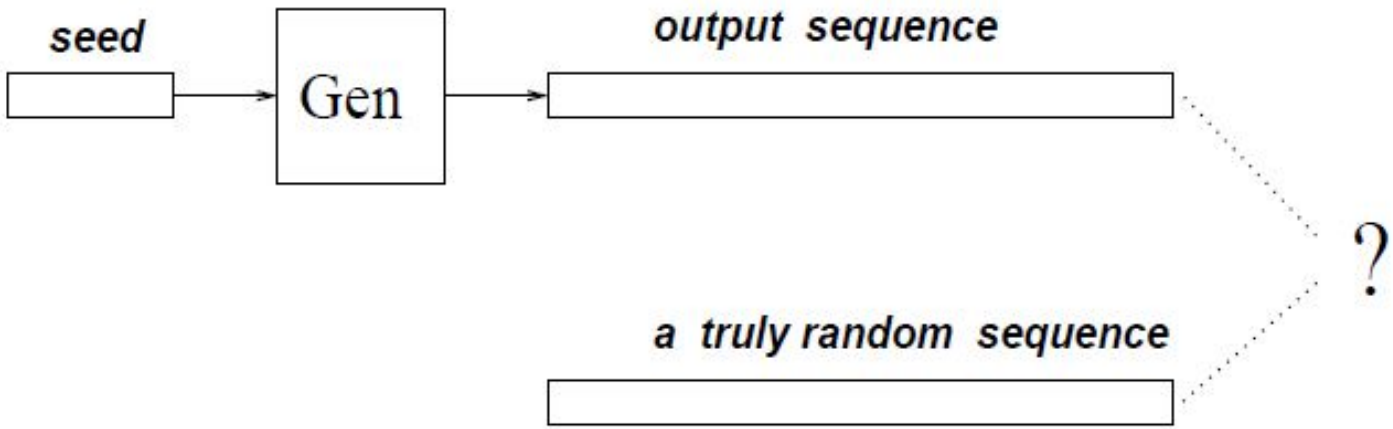
Ένας αλγόριθμος που μετατρέπει μικρά τυχαία strings (seeds) σε μεγάλες ψευδοτυχαίες ακολουθίες.

- Αποδοτικός: Πρέπει(;) να υλοποιείται από ντετερμινιστικό πολυωνυμικό αλγόριθμο.
- Επέκταση: Μετατροπή strings μεγέθους n σε ακολουθίες μεγέθους $l(n)$ με $l(n) > n$.
- Ψευδοτυχαιότητα: Δεν υπάρχει PPT που να διαχωρίζει υπολογιστικά το αποτέλεσμα από την ομοιόμορφη κατανομή.

Γενικός Ορισμός

Ένας αλγόριθμος G είναι *ψευδοτυχαία γεννήτρια* αν υπάρχει μία συνάρτηση $l : \mathbb{N} \to \mathbb{N}$ για την οποία να ισχύει $\forall\, k \;\; l(k) > k$ - *stretch function* - *συνάρτηση έκτασης* - τέτοια ώστε για κάθε PPT D (distinguisher), για κάθε θετικό πολυώνυμο p, και για αρκετά μεγάλα k

$$|Pr[D(G(U_k)) = 1] - Pr[D(U_{l(k)}) = 1] < \frac{1}{p(k)}$$



Αποδοτικότητα PRG

- Στην κρυπτογραφία πρέπει να είναι πολυωνυμικός γιατί τον χρησιμοποιούν οι νόμιμοι χρήστες του κρυπτοσυστήματος, οι οποίοι έχουν περιορισμένους πόρους.
- Στο derandomisation μπορεί να είναι εκθετικός, γιατί θα χρησιμοποιηθεί από εκθετικό αλγόριθμο (trivial derandomisation).

Εναλλακτικοί ορισμοί (με κυκλώματα):

- Μία κατανομή R στο $\{0,1\}^m$ είναι (S,ϵ) ψευδοτυχαία αν για κάθε κύκλωμα C μεγέθους το πολύ $S \in \mathbb{N}$

$$|Pr[C(R)=1] - Pr[C(U_m)=1]| < \epsilon$$

- Μία 2^n υπολογίσιμη συνάρτηση $G : \{0,1\}^* \to \{0,1\}^*$ είναι $S(l) - PRG$ όπου $S : \mathbb{N} \to \mathbb{N}$ αν:
  - $|G(z)| = S(|z|), z \in \{0,1\}^*$
  - Η κατανομή $G(U_l)$ είναι $(S(l)^3, 0.1)$ ψευδοτυχαία $l \in \mathbb{N}$

Παρατήρηση: Τα νούμερα είναι αυθαίρετα.

Derandomisation με PRG

Theorem

Αν $S : \mathbb{N} \to \mathbb{N}$ *και υπάρχει* $S(l) - PRG$ *τότε* $BPTIME(S(l(n))) \subseteq DTIME(2^{cl(n)}))$ *για* c *σταθερά και* $l : \mathbb{N} \to \mathbb{N}$

Δηλαδή:

- $BPP = P$ αν υπάρχει $2^{\epsilon l}$ PRG.
- $BPP \subseteq DTIME(2^{polylog})]$ αν υπάρχει $2^{l^{\epsilon}}$ PRG.
- $BPP \subseteq DTIME(2^{n^{\epsilon}})]$ αν υπάρχει l^{c} PRG.

Derandomisation με PRG - Απόδειξη

Βασική Ιδέα

- Για την προσομοίωση του αλγόριθμου $A(x, r)$ χρησιμοποιούμε ψευδοτυχαίο r.
- Διαλέγουμε τυχαίο $z \in \{0,1\}^{l(n)}$ (το οποίο θα έχει λιγότερα bits) από το r
- Εκτελούμε το A(x,G(z)), όπου G ο PRG
- $Pr[A(x, G(z)) = L(x)] \geq \frac{2}{3} - 0.1 > 0.5$

Άρα αν κάνουμε trivial derandomisation δεν χρειαζόμαστε χρόνο (2^m) αλλά $(2^{l(n)})$.

Hardness and Derandomisation

- Average Case Hardness
- $H_{avg}(f) = max\{S | Pr[C(x) = f(x)] < \frac{1}{2} + \frac{1}{S}\}$
- όπου
  - x τυχαίο $\in \{0,1\}^n$
  - f συνάρτηση $\{0,1\}^n \to \{0,1\}$
  - C κύκλωμα n εισόδων μεγέθους το πολύ S.

Παρά το γεγονός ότι δίνουμε στο κύκλωμα την ίδια είσοδο με την συνάρτηση, του φαίνεται τυχαία.

RPGs from average-case hardness

Theorem

Αν υπάρχει $f \in DTIME(2^{O(n)})$ *τέτοια ώστε* $H_{avg}(f)(n) \geq S(n)$ *τότε υπάρχει* $S(\delta l)^{\delta}$ *PRG για* $\delta > 0$.

Proof.

Σταδιακή κατασκευή PRG από hard function - (Nisan - Wigderson Construction) □

Επέκταση κατά 1 bit

Theorem (RPG από hard function)

Αν υπάρχει συνάρτηση $f \in E$ με $H_{avg}(f) \geq n^4$, τότε υπάρχει $S(l) = l + 1$ PRG G.

Proof.

Συνένωση της εισόδου με το αποτέλεσμα της συνάρτησης - $G(z) = z \bullet f(z)$
Πρέπει να δειχθεί ότι ο G είναι όντως PRG, δηλ.
Η κατανομή $G(U_l)$ είναι $(l+1)^3, 0.1$ ψευδοτυχαία. □

Unpredictability → Pseudorandomness

Theorem (Yao)

Έστω $Pr_{r\in_r Y}[C(r_1,...,r_{i-1}) = r_i] \leq \frac{1}{2} + \frac{\epsilon}{m}$ *όπου:*

- $S > 10n$
- C *κύκλωμα μεγέθους* $2S$

Τότε η κατανομή Y *είναι* (S,ϵ) *ψευδοτυχαία.*

- Υπενθυμίζουμε ότι $G(z) = z \bullet f(z)$.
- Αφού η $H_{avg}(f) \geq n^4$, ισχύει:
- $Pr[C(x) = f(x)] < \frac{1}{2} + \frac{1}{n^4}$ για κάθε κύκλωμα μεγέθους n.

Απόδειξη του 1-bit PRG

- Το z προέρχεται από την ομοιόμορφη κατανομή.
- Άρα κάθε bit του z δεν μπορει να υπολογιστεί από τα υπόλοιπα.
- Αρκεί λοιπόν να αποδείξουμε ότι το τελευταίο bit (δηλ. $f(z)$) είναι μη προβλέψιμο από όλα τα υπόλοιπα.
- Από ορισμό, πρέπει για κανένα κύκλωμα C μεγέθους $2(l+1)^3$ να ισχύει:
- $Pr_{z\in_r\{0,1\}^l}[C(z) = f(z)] > \frac{1}{2} + \frac{1}{20(l+1)} > \frac{1}{2} + \frac{1}{l^4}$
- Δεν είναι δυνατόν λόγω της δυσκολίας της f.

Επέκταση κατά 2 bits

Theorem (RPG από hard function)

Αν υπάρχει συνάρτηση $f \in E$ με $H_{avg}(f) \geq n^4$. Τότε υπάρχει $l+2$ PRG G.

- Εφαρμόζουμε την συνάρτηση δύο φορές στο πρώτο μισό και στο δεύτερο μισό.
- Συνενώνουμε το αποτέλεσμα.
- Δηλαδη
  $G(z) = z_1 ... z_{l/2} \bullet f(z_1, ..., z_{l/2}) \bullet z_{l/2+1} ... z_l \bullet f(z_{l/2+1}, ..., z_l)$
- Πρέπει να δειχθεί ότι ο G είναι όντως PRG.
- Όπως και πριν χρησιμοποιούμε το θεώρημα του Yao και την δυσκολία της f.

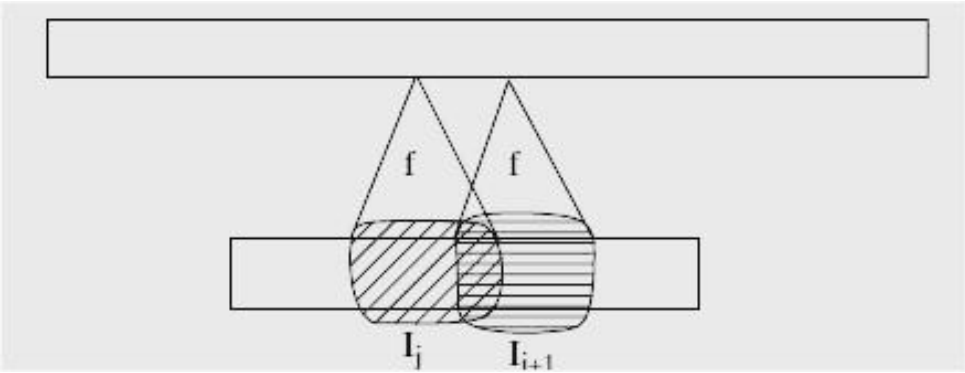
Επιπλέον επέκταση

- Με την διαδικασια αυτή **δεν** μπορούμε να επεκτεινουμε περισσοτερο από το διπλάσιο της εισόδου.
- $G(z) = z_1 \bullet f(z_1) \bullet \ldots \bullet z_l \bullet f(z_l)$
- Θέλουμε εκθετικά μεγαλύτερη έξοδο.
- Θα συνδυάσουμε κομμάτια της εισόδου αντί να τα χρησιμοποιούμε ανεξάρτητα (combinatorial design)
- Αναγκαστικά τα κομμάτια της εισόδου που θα συνδυάσουμε δεν θα είναι ξένα μεταξύ τους.
- Η επικάλυψη πρέπει να είναι μικρή, ώστε να μην υπάρχει πρόβλεψη.
- Τελικά η έξοδος θα αυξηθεί τόσο πολύ που θα μπορούμε να αγνοήσουμε εντελώς την είσοδο και να συνενώσουμε απλά τα αποτελέσματα της συνάρτησης.

Nisan - Wigderson construction

$NW_f^I(z) = f(z_{I_1}) \bullet f(z_{I_2}) \bullet \ldots \bullet f(z_{I_m})$ όπου

- $f : \{0,1\}^n \to \{0,1\}$ Πρέπει να επιδεικνύει κάποια δυσκολία
- $\{I_1, I_2, \ldots I_m\}$: μία οικογένεια υποσυνόλων του $[l]$ με n στοιχεία το καθένα. Τα υποσύνολα πρέπει να παράγονται από μία combinatorial design.
- z_{I_l}: διαλέγουμε από το z εκείνα τα bits το οποία αντιστοιχούν στο υποσύνολο $_l$.



Combinatorial Designs - Ορισμός

(l, n, d) Combinatorial Design

Μία οικογένεια $\mathfrak{I} = \{I_1, I_2, ...I_m\}$ υποσυνόλων του $\{1..l\}$ όπου:

- Κάθε ένα έχει έχει μέγεθος n.
- $|I_j \cap I_k| \leq d, j \neq k$
- $d < n < l$

Combinatorial Designs - Κατασκευή

(Greedy-Εκθετικός) Αλγόριθμος Κατασκευής

1. $\mathfrak{I} \leftarrow \emptyset$
2. Κατασκεύασε το $\{I_1, I_2, ...I_m\}$:
   - Για όλα τα υποσύνολα του $[1..l]$ πρόσθεσε το πρώτο I για το οποίο
   - $|I \cap I_j| \leq d$ όπου $j \in 1..m$
3. Τερματισμός όταν $m = 2^{d/10}$

Combinatorial Designs - Ορθότητα Κατασκευής

- Πρέπει να αποδείξουμε ότι υπάρχει υποσύνολο μεγέθους n ώστε $|I \cap I_j| \leq d$
- Διαλέγουμε ανεξάρτητα τα στοιχεία με πιθανότητα $\frac{2n}{l}$.
- Αναμενόμενο μέγεθος : $2n$.
- Αναμενόμενο μέγεθος $I \cap I_j$: $\frac{2n^2}{l}$
- Από Chernoff Bound:
  - $Pr[|I| \geq n] \geq 0.9$
  - $Pr[|I \cap I_j| \geq d] \geq 0.5 \times 2^{\frac{-d}{10}}$
- $Pr[(|I| \geq n) \wedge (|I \cap I_j| \leq d)] \geq 0.4$
- Μπορούμε να αφαιρέσουμε στοιχεια από το I, χωρίς να χαλάσουμε την κατασκευή.

Ψευδοτυχαιότητα NW generator

Theorem

Η κατανομή $NW^f_{\mathfrak{I}}(U_l)$ *είναι* $\frac{H_{avg}(f)}{10}, \frac{1}{10}$ *ψευδοτυχαία, όπου:*

- $\mathfrak{I}$ *είναι* (l, n, d) *με* $|\mathfrak{I}| = 2^{\frac{d}{10}}$
- $f : \{0,1\}^n \to \{0,1\}$ *με* $H_{avg}(f) > 2^{2d}$

Proof.

- Απαγωγή σε άτοπο.
- Χρήση θεωρήματος Yao.
- Παρά το γεγονός ότι τα διάφορα I_m είναι αλληλοεξαρτώμενα, η δυσκολία της f δεν επιτρέπει την πρόβλεψη.

□

Ψευδοτυχαιότητα NW generator - Απόδειξη (1)

- $H_{avg}(f) > 2^{2d} \rightsquigarrow Pr[C(x) = f(x)] < \frac{1}{2} + \frac{1}{2^{2d}}$, όπου C κύκλωμα με:$|C| \leq 2^{2d}$
- Από θεώρημα Yao: Για να είναι η R $(2^{2d}/10, 1/10)$ ψευδοτυχαία πρέπει για κάθε κύκλωμα C με $|C| \leq 2^d$ ισχύει ότι

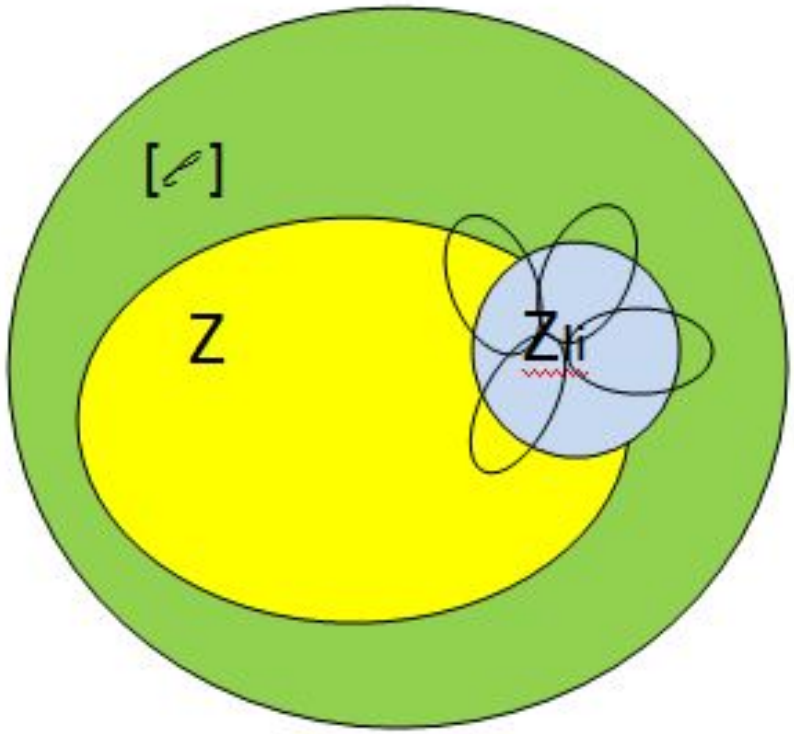
$$Pr[C(R_1, ..., R_{i-1}) = R_i] < \frac{1}{2} + \frac{1}{10\,2^{d/10}}, i \in \{1..2^{d/10}\}].$$

- Έστω ότι υπάρχει τέτοιο C ώστε:

$$Pr[C(f(Z_{I_1}), ..., f(Z_{I_{i-1}})) = f(Z_{I_i})] \geq \frac{1}{2} + \frac{1}{10\,2^{d/10}}.$$

Ψευδοτυχαιότητα NW generator - Απόδειξη (2)

- Ορίζουμε τις τυχαίες μεταβλητές
  - Z_1 : συνιστώσες(bits) του z στο I_i και
  - Z_2 : συνιστώσες(bits) του z στο δεν ανήκουν στο I_i δηλ. $[l] - I_i$.
- Ορίζουμε $f_j(Z_1, Z_2) = f(Z_1 < I_j \cap I_i > \bullet Z_2 < I_j - I_i >)$



Ψευδοτυχαιότητα NW generator - Απόδειξη (3)

- Τότε $Pr[C(f_1(Z_1, Z_2), ..., f_{i-1}(Z_1, Z_2)) = f_i(Z_1)] \geq \frac{1}{2} + \frac{1}{10\, 2^{d/10}}$
- Υπάρχει $z_2 \in \{0,1\}^{l-n}$ ώστε:
- Τότε $Pr[C(f_1(Z_1, z_2), ..., f_{i-1}(Z_1, z_2)) = f_i(Z_1)] \geq \frac{1}{2} + \frac{1}{10\, 2^{d/10}}$
- Μήπως μπορεί να προβλεφτεί το $f_i(Z_1)$ αφού έχουμε δεί τμήματα του μέσω των $f_i(Z_1, z_2)$.
- Όχι, λόγω του ορισμού της combinatorial design και της δυσκολίας της f.

Ψευδοτυχαιότητα NW generator - Απόδειξη (4)

- Αφού έχουμε combinatorial design $|I \cap I_k| \leq d$ η συνάρτηση $Z_1 \mapsto f(Z_1, z_2)$ εξαρτάται από d συντεταγμένες του $_1$.
- Άρα μπορεί να υπολογιστει από $d2^d$ κύκλωμα B.
- Τότε $Pr[B(Z_1) = f(Z_1)] \geq \frac{1}{2} + \frac{1}{10\, 2^{d/10}}$
- Άτοπο λογω της δυσκολίας της f.
- $H_{avg}(f) > 2^{2d} \rightsquigarrow Pr[C(x) = f(x)] < \frac{1}{2} + \frac{1}{2^{2d}}$, όπου C κύκλωμα με:$|C| \leq 2^{2d}$

Συμπεράσματα ... (μέχρι τώρα)

- Ψευδοτυχαιότητα:Υπολογιστική ομοιότητα με ομοιόμορφη κατανομή.
- PRG: Κατασκευές που επεκτείνουν μια τυχαία ακολουθία διατηρώντας την ψευδοτυχαιότητα.
- Χρήση:
  - Ενίσχυση τυχαιότητας για χρήση στην κρυπτογραφία (πολυωνυμικοί PRG).
  - Derandomisation με ισχυρές υποθέσεις (ύπαρξη one-way functions, $NP \neq P$).
- NW: Κατασκευή εκθετικού PRG με την υπόθεση ότι υπάρχει μία οποιαδήποτε δύσκολη συνάρτηση.
- Δύσκολες συναρτήσεις: Πιο εύκολο να μαντέψεις την τιμή τους.
- Πιο ρεαλιστική υπόθεση για derandomisation.

Συνέπειες στην Πολυπλοκότητα

- Υπό προϋποθέσεις...
  - $BPP = P$
  - $BPP \subset SUBEXP$
  - $BPP \subset QuasiP$
  - Parallel Computation
- Χωρίς προϋποθέσεις...
  - Constant Depth Circuits
  - $AM = almostNP$
  - $BPP \subset \Sigma_2 \cap \Pi_2$
  - $PH = almostPH$

$BPP = P$ ...υπό προϋποθέσεις

Theorem

Αν υπάρχει συνάρτηση $f \in E$ με $H_{wrs}(f) = 2^{\epsilon n}$ για κάποιο $\epsilon > 0$ τότε $BPP = P$.

Proof.

- Από NW, μπορεί να κατασκευαστεί PRG $logn \to n$
- Άρα το derandomisation γίνεται σε $2^{logn}T(n)$.
- Δηλ. $BPP \subseteq P$ και $BPP = P$.

□

Παρατήρηση: Από worst case hardness μπορούμε να πάρουμε average case hardness.

$BPP \subset SUBEXP$ ...υπό προϋποθέσεις

Theorem

Αν υπάρχει συνάρτηση που δεν μπορεί να προσεγγιστεί με πολυωνυμικά κυκλώματα τότε $BPP \subset SUBEXP$.

Proof.

- Έστω f δεν μπορεί να προσεγγιστεί με κυκλώματα n^c
- υπάρχει συνάρτηση $\in E$ η οποία έχει δυσκολία n^c
- υπάρχει PRG: $n^\epsilon \to n$
- Άρα το derandomisation γίνεται σε χρόνο $2^{n^\epsilon} T(n)$
- $A \in BPP \Rightarrow A \in SUBEXP$

□

$BPP \subset QuasiP$ ...υπό προϋποθέσεις

Theorem

Αν υπάρχει συνάρτηση που να μην μπορεί να προσεγγιστεί με 2^{n^ϵ} κυκλώματα τότε $BPP \subset QuasiP$.

Proof.

- Έστω f δεν μπορεί να υπολογιστεί με κυκλώματα 2^{n^ϵ}
- υπάρχει συνάρτηση $f_1 \in E$ η οποία έχει δυσκολία 2^{n^ϵ}
- υπάρχει PRG: $(logn)^c \to n$
- Άρα το derandomisation γίνεται σε χρόνο $2^{(logn)^c} T(n)$
- $A \in BPP \Rightarrow A \in QuasiP$

□

Parallel Computation

Αφού κατασκευαστούν τα υποσύνολα ο PRG $NW_f^I(z) = f(z_{I_1}) \bullet f(z_{I_2}) \bullet \ldots \bullet f(z_{I_m})$ είναι παράλληλος.

Theorem

- *Αν υπάρχει συνάρτηση στο PSPACE που να μην μπορεί να προσεγγιστεί με NC κυκλώματα τότε* $RNC \subset \bigcap_{\epsilon>0} DSPACE(n^\epsilon)$.
- *Αν υπάρχει συνάρτηση στο PSPACE που να μην μπορεί να προσεγγιστεί με κυκλώματα βάθους* n^ϵ *τότε* $RNC \subset DSPACE(polylog)$.

Parallel Computation - (2)

Proof.

- $NC = PT/WK(log^k n, n^k)$ - (polylog depth, polynomial size)
- RNC: προσθήκη randomisation
- Αντικαθιστά προηγούμενο θεώρημα: Αν η εύρεση αντιστροφου $modp$ δεν μπορεί να προσεγγιστεί με NC κυκλώματα τότε $RNC \subset \bigcap_{\epsilon>0} DSPACE(n^\epsilon)$.

□

Constant Depth Circuits

Theorem

$|Pr[C_n(x) = parity(x)] - \frac{1}{2}| \leq 2^{n^{1/(d+1)}}$ *όπου*

- C_n *οικογένεια κυκλωμάτων με βάθος* d *και μέγεθος* $\leq 2^{n^{1/(d+1)}}$.
- $x \in_R \{0,1\}^n$

- Η συνάρτηση parity = ’περιττό πλήθος 1΄ δεν μπορεί να υπολογιστεί από πολυωνυμικά κυκλώματα σταθερού βάθους (AC^0)
- Άρα η συνάρτηση parity μας κάνει για δύσκολη
- Μπορούμε να κατασκευάσουμε οικογένειες από NW-constructions

Constant Depth Circuits-(2)

$BPAC^0$: Πιθανοτικά, πολυωνυμικά κυκλώματα σταθερού βάθους με two-sided error

Theorem

- $BPAC^0, RAC^0 \subset \cup_c DSPACE((logn)^c)$
- $BPAC^0, RAC^0 \subset QuasiP$

RAC^0 : Πιθανοτικά, πολυωνυμικά κυκλώματα σταθερού βάθους με one-sided error

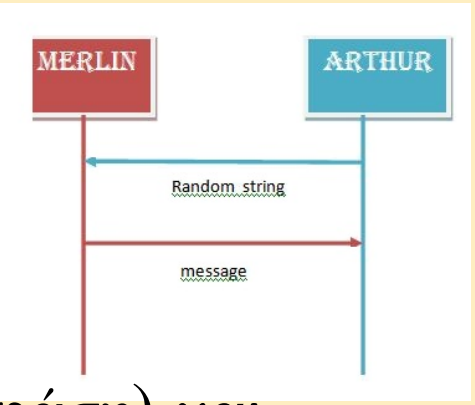
$AM = almost - NP$

AM

$L \in AM$

$x \in L : Pr[ACCEPTS_{arthur}(x) = 1] \geq \frac{2}{3}$

$x \notin L : Pr[ACCEPTS_{arthur}(x) = 1] \leq \frac{1}{3}$



Ισοδύναμα: χρήση τυχαιότητας(Arthur φάση) και μη-ντετερμινισμού(Merlin φάση) με αυτή τη σειρά.

almost-NP

$almost - NP = \{L | Pr[L \in NP^A] = 1\}$, όπου A random oracle.

Απόδειξη

Proof.

$AM \subset almost - NP$ Το random oracle είναι η φάση Arthur. Μία μηχανή AM μπορεί να θεωρηθεί ως μία μηχανή NP, όπου πριν την μη ντετερμινιστική επιλογή ρωτάει ένα random oracle. □

Proof.

$almost - NP \subset AM$

- Προσομοίωση συγκεκριμένης NDTM M που χρησιμοποιεί random oracle με AM.
- **Πρόβλημα**: Η M μπορεί μη ντετερμινιστικά να έχει πρόσβαση στο random oracle εκθετικό αριθμό φορών.
- Η AM όμως έχει πολυωνυμική τυχαιότητα.

□

Απόδειξη - *almost* $-$ $NP \subset AM$

Proof.

...συνέχεια

- Χρηση NW generator: Μπορεί από πολυωνυμικό αριθμό bits να παράγει εκθετικό αριθμό που να φαίνεται τυχαίος.
- Μετατροπή υπολογισμού της M σε κύκλωμα βάθους 2
- Φάση Arthur: Δημιουργία πολυωνυμικών τυχαιών bits.
- Φάση Merlin: Προσομοίωση της M.
- Οποτεδήποτε η M έχει πρόσβαση στο Oracle, χρήση generator.

□

$BPP \subset \Sigma_2 \cap \Pi_2$

- Αφού $BPP = coBPP$, αρκεί $BPP \subset \Sigma_2$
- Μπορεί να φτιαχτεί PRG $logn \to n$ στο Σ_2 ως εξής:
  - Αρκεί να βρούμε συνάρτηση με εκθετική δυσκολία.
  - Μάντεψε συνάρτηση με το κατάλληλο πλήθος bits $\exists$
  - Επαλήθευση ότι είναι όντως δύσκολη, ελέγχοντας όλες τις καταχωρήσεις του πίνακα της (πολυωνυμικές). $\forall - coNP$
  - Χρηση στον NW PRG
  - Δοκιμή όλων των seeds.

$almost - PH = PH$

Proof.

- Ένα random oracle μπορει να προσομοιωθεί από μία φάση Artur.
- Η φάση Arthur μπορεί να προσομοιωθεί από μία επιπλέον εναλλαγή (μάντεψε-επαλήθευσε).

□

Βιβλιογραφία

1. Luca Trevisan, Pseudorandomness and Combinatorial Constructions, CoRR abs/cs/0601100, 2006, `http://arxiv.org/abs/cs/0601100`
2. Κεφ.20 από S. Arora and B. Barak, Complexity Theory: A Modern Approach, 2008, Princeton University.
3. Srikanth Srinivasan, The Nisan-Wigderson Pseudorandom Generator, IITK Theory Meeting Notes, `http://www.cmi.ac.in/~ramprasad/theorymeet/01_srikanth_NW.pdf`
4. O. Goldreich, Computational Complexity: A Conceptual Perspective, 2008, Cambridge University Press.
5. Luca Trevisan, Lecture Notes on Pseudorandomness - Part II (derandomization), 2000, `http://www.wisdom.weizmann.ac.il/~oded/PS/ln00b.ps`
6. Oded Goldreich, Lecture Notes on Pseudorandomness - Part I (polynomial-time generators), 2000 `http://www.wisdom.weizmann.ac.il/~oded/PS/ln00a.ps`
7. N. Nisan, A. Wigderson, Hardness vs Randomness, J. Comput. Syst. Sci., 49(2):149-167, 1994 88-103, 2002