Περιεχόμενα

Κλάσεις Πολυπλοκότητας Ι

- **Θεωρία Υπολογισμού**: Μας ενδιαφέρει μόνον αν ένα πρόβλημα είναι **υπολογίσιμο ή όχι**.
- **Θεωρία Πολυπλοκότητας**: Θεωρούμε μόνο υπολογίσιμα προβλήματα και προσπαθούμε να δούμε αν μπορούν να επιλυθούν με **περιορισμούς στους διαθέσιμους υπολογιστικούς πόρους**, όπως ο χρόνος υπολογισμού, ο επιπλέον χώρος μνήμης που απαιτείται για ενδιάμεσα αποτελέσματα κατά την επίλυση, και άλλοι.
- Αυτοί οι *περιορισμοί*, καθώς και άλλα χαρακτηριστικά των υπολογισμών ορίζουν **κλάσεις πολυπλοκότητας** μέσα στις οποίες τοποθετούμε τα διάφορα προβλήματα.

Παρά τις συνεχείς και μακροχρόνιες προσπάθειες πολλών επιστημόνων, υπάρχουν αρκετά **ανοιχτά ερωτήματα**. Π.χ., υπάρχουν προβλήματα για τα οποία, αν και ανήκουν στο NP, δεν έχει βρεθεί πολυωνυμικός αλγόριθμος, αλλά ούτε απόδειξη ότι είναι NP-complete.

Κλάσεις Πολυπλοκότητας (Τι δεν ήξερε ο Karp το 1972)

- **GRAPH ISOMORPHISM** (το πιο γνωστό ανοιχτό πρόβλημα): Δεδομένων δύο γράφων είναι ισομορφικοί; (παράβαλε με το SUBGRAPH ISOMORPHISM το οποίο είναι γνωστό ότι είναι NP-complete)
- **LINEAR PROGRAMMING** (παρέμενε για χρόνια ανοιχτό): δεδομένου ενός συστήματος γραμμικών εξισώσεων και ανισοτήτων και μιας γραμμικής αντικειμενικής συνάρτησης (μεγιστοποίηση ή ελαχιστοποίηση) να ευρεθεί μια βέλτιστη εφικτή λύση;
  - **Μέθοδος Simplex** (Dantzig): Στη χειρότερη περίπτωση χρειαζόταν εκθετικό χρόνο.
  - **Ελλειψοειδής μέθοδος** (Khachiyan): Ο πρώτος πολυωνυμικός αλγόριθμος για τον γραμμικό προγραμματισμό. Δεν είχε μεγάλο πρακτικό ενδιαφέρον.
  - **Αλγόριθμος Karmarkar**: Πολυωνυμικός αλγόριθμος που είχε και πρακτικά αποτελέσματα καλύτερα από τη μέθοδο Simplex.
- **PRIMALITY**: Δίνεται ένας ακέραιος. Είναι πρώτος ή όχι;
  Πρόσφατα (2002 από τους Agrawal, Kayal, Saxena --- AKS) απεδείχθη και ότι το πρόβλημα αυτό, που παρέμενε για αρκετό καιρό ανοικτό, είναι στο P.

Βασικοί Ορισμοί I

Ορισμός

Στην κλάση TIME($t(n)$) (ή DTIME($t(n)$)) ανήκουν τα προβλήματα που μπορούν να επιλυθούν από **ντετερμινιστική** μηχανή Turing σε χρόνο $t(n)$.

Ορισμός

Στην κλάση NTIME($t(n)$) ανήκουν τα προβλήματα που μπορούν να επιλυθούν από **μη ντετερμινιστική** μηχανή Turing σε χρόνο $t(n)$.

Ορισμός

Στην κλάση SPACE($s(n)$) (ή DSPACE($s(n)$)) ανήκουν τα προβλήματα που μπορούν να επιλυθούν από **ντετερμινιστική** μηχανή Turing χρησιμοποιώντας επιπλέον χώρο $s(n)$.

Ορισμός

Στην κλάση NSPACE($s(n)$) ανήκουν τα προβλήματα που μπορούν να επιλυθούν από **μη ντετερμινιστική** μηχανή Turing χρησιμοποιώντας επιπλέον χώρο $s(n)$.

Με βάση τα παραπάνω, ορίζουμε:

Βασικοί Ορισμοί II

- $\mathrm{P} = \mathrm{PTIME} = \bigcup_{i \geqslant 1} \mathrm{DTIME}(n^i)$
- $\mathrm{NP} = \mathrm{NPTIME} = \bigcup_{i \geqslant 1} \mathrm{NTIME}(n^i)$
- $\mathrm{PSPACE} = \bigcup_{i \geqslant 1} \mathrm{DSPACE}(n^i)$
- $\mathrm{NPSPACE} = \bigcup_{i \geqslant 1} \mathrm{NSPACE}(n^i)$
- $\mathrm{L} = \mathrm{DSPACE}(\log n)$
- $\mathrm{NL} = \mathrm{NSPACE}(\log n)$
- $\mathrm{EXP} = \bigcup_{i \geqslant 1} \mathrm{DTIME}(2^{n^i})$
- $\mathrm{EXPSPACE} = \bigcup_{i \geqslant 1} \mathrm{DSPACE}(2^{n^i})$

Παρατήρηση

Μία συνάρτηση f ονομάζεται συνάρτηση πολυπλοκότητας (constructible) αν πρέπει να υπάρχει μία TM τέτοια ώστε: $\forall$ input x με $|x| = n$, αποδέχεται το input σε χρόνο $O(n + f(n))$ (time-constructible) ή working space $O(f(n))$ (space-constructible).

Βασικοί Ορισμοί III

Αν η f είναι μία συνάρτηση πολυπλοκότητας τότε ισχύουν:

- $\text{DSPACE}(f(n)) \subseteq \text{NSPACE}(f(n))$
- $\text{DTIME}(f(n)) \subseteq \text{NTIME}(f(n))$

διότι κάθε ντετερμινιστική μηχανή Turing μπορεί να θεωρηθεί ως μη ντετερμινιστική με μία μόνο επιλογή σε κάθε βήμα.

- $\text{DTIME}(f(n)) \subseteq \text{DSPACE}(f(n))$
- $\text{NTIME}(f(n)) \subseteq \text{DSPACE}(f(n))$

διότι σε χρόνο $f(n)$ δεν μπορεί να εξεταστεί χώρος (αριθμός θέσεων στην ταινία της $T.M.$) παραπάνω από $f(n)$.

Αν $f(n) > \log n$ τότε:

- $\text{DSPACE}(f(n)) \subseteq \text{DTIME}(c^{f(n)})$
- $\text{NTIME}(f(n)) \subseteq \text{DTIME}(c^{f(n)})$

Βασικοί Ορισμοί IV

- $\text{NSPACE}(f(n)) \subseteq \text{DTIME}(k^{f(n)})$

Το παρακάτω θεώρημα οφείλεται στον Savitch (1970):

Θεώρημα

Αν $f(n) \geqslant \log n$ *τότε* $\text{NSPACE}(f(n)) \subseteq \text{DSPACE}(f^2(n))$.

Άμεσα από το θεώρημα του Savitch προκύπτει ότι PSPACE = NPSPACE.

Από τις παραπάνω σχέσεις προκύπτει η εξής ιεραρχία:

$$\text{L} \subseteq \text{NL} \subseteq \text{P} \subseteq \text{NP} \subseteq \text{PSPACE} = \text{NPSPACE}$$

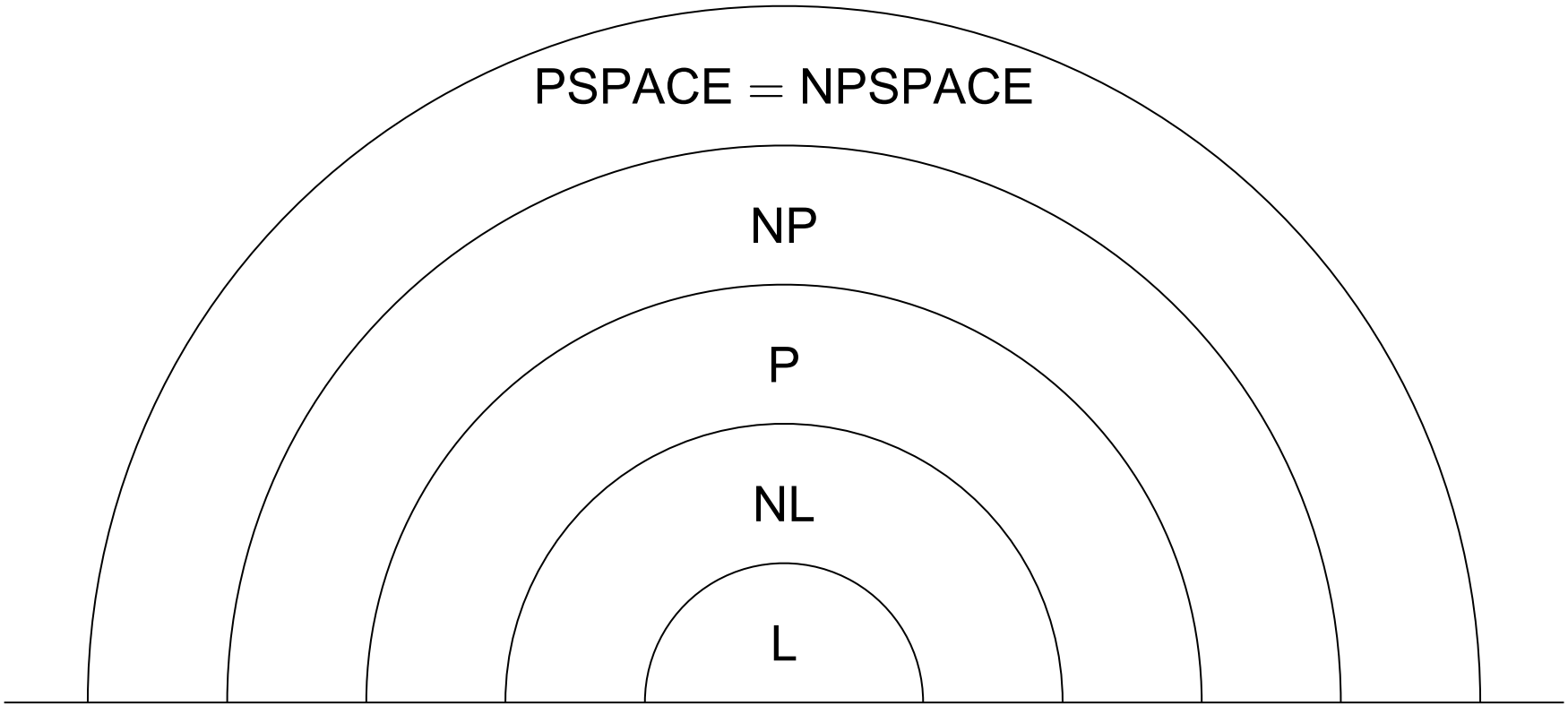
Γνωρίζουμε ότι $\text{L} \neq \text{PSPACE}$ και $\text{NL} \neq \text{PSPACE}$ (αυτό προκύπτει από το θεώρημα ιεραρχίας για χωρικές κλάσεις πολυπλοκότητας, που αναφέρεται παρακάτω).

Ανοιχτά παραμένουν τα προβλήματα:

$$\text{L} \supseteq \text{NL} \supseteq \text{P} \supseteq \text{NP} \supseteq \text{PSPACE}$$

Βασικοί Ορισμοί V

Ο κόσμος μοιάζει, ως τώρα, να είναι όπως στο παρακάτω σχήμα.



Σχήμα: Κλάσεις πολυπλοκότητας

Βασικοί Ορισμοί VI

Επίσης, πρέπει να αναφέρουμε ότι οι παραπάνω κλάσεις πολυπλοκότητας αφορούν προβλήματα **απόφασης**. Μπορούμε επίσης να ορίσουμε κλάσεις πολυπλοκότητας για μηχανές Turing που υπολογίζουν **συναρτήσεις**. Ένα χαρακτηριστικό παράδειγμα είναι η παρακάτω κλάση:

Ορισμός

FP = το σύνολο των συναρτήσεων που υπολογίζεται από ντετερμινιστική μηχανή Turing σε πολυωνυμικό χρόνο.

Η κλάση FP θα φανεί χρήσιμη παρακάτω στον ορισμό των αναγωγών, αφού περιλαμβάνει τις ``εύκολα'' υπολογιζόμενες συναρτήσεις.

Μία άλλη επίσης χρήσιμη κλάση πολυπλοκότητας που αφορά συναρτήσεις είναι η εξής:

Ορισμός

FL = το σύνολο των συναρτήσεων που υπολογίζεται από ντετερμινιστική μηχανή Turing σε λογαριθμικό χώρο.

Θεωρήματα ιεραρχίας I

Ισχύουν τα παρακάτω θεωρήματα για το μοντέλο της ντετερμινιστικής μηχανής Turing με τρεις ταινίες (θεωρούμε πάντοτε συναρτήσεις πολυπλοκότητας t_1, t_2, s_1, s_2):

Θεώρημα (Fürer, 1982)

Έστω $t_2(n) > n$. Τότε υπάρχει γλώσσα που γίνεται αποδεκτή σε χρόνο t_2, αλλά όχι σε χρόνο t_1 για οποιοδήποτε $t_1 = o(t_2(n))$.

Θεώρημα (Hartmanis, Lewis, Stearns, 1965)

Έστω $s_2(n) > \log n$. Τότε υπάρχει γλώσσα που γίνεται αποδεκτή σε χώρο s_2, αλλά όχι σε χώρο s_1 για οποιοδήποτε $s_1 = o(s_2(n))$.

Οι τεχνικές αποδείξεις παραλείπονται.

Παρόμοια θεωρήματα ισχύουν και για μη ντετερμινιστικές μηχανές Turing. Οι αποδείξεις μάλιστα είναι πιο εύκολες.

Θεωρήματα ιεραρχίας II

Η εμμονή μας σε constructible συναρτήσεις πολυπλοκότητας οφείλεται στο γεγονός ότι αν επιτρέψουμε οποιαδήποτε συνάρτηση στην θέση των $t(n)$, $s(n)$, τότε προκύπτουν διάφορα παθολογικά φαινόμενα, όπως το παρακάτω:

Θεώρημα (Gap theorem)

Υπάρχει αναδρομική συνάρτηση $t(n)$, τέτοια ώστε $\text{TIME}(t(n)) = \text{TIME}(2^{t(n)})$.

Συμπληρωματικές κλάσεις πολυπλοκότητας I

Ορισμός

Έστω γλώσσα L. Ως γνωστόν, το συμπλήρωμα της γλώσσας συμβολίζεται και ορίζεται ως εξής: $\overline{L} = \{x \mid x \notin L\}$. Τώρα, για μία κλάση γλωσσών $\mathcal{C}$, ορίζουμε (με την βοήθεια του συμπληρώματος):

$$\text{co}\mathcal{C} = \{\overline{L} \mid L \in \mathcal{C}\}.$$

Παράδειγμα: η κλάση coNP αποτελείται από τις γλώσσες που είναι συμπληρώματα γλωσσών στο NP. Ένα πρόβλημα που ανήκει στην κλάση coNP είναι το $\overline{\text{SAT}}$ ή το στενά συσχετιζόμενο με αυτό πρόβλημα της ταυτολογίας, αν δηλαδή ένας λογικός τύπος που δίνεται είναι ταυτολογία.

Συμπληρωματικές κλάσεις πολυπλοκότητας II

Έχει ενδιαφέρον να δούμε ποιες κλάσεις πολυπλοκότητας είναι **κλειστές ως προς συμπλήρωμα**, δηλαδή για ποιες κλάσεις $\mathcal{C}$ ισχύει $\mathcal{C} = co\mathcal{C}$.

Γενικά, οι ντετερμινιστικές κλάσεις πολυπλοκότητας (είτε χρονικές, είτε χωρικές) είναι κλειστές ως προς συμπλήρωμα, δηλαδή, οι $DTIME(t(n))$ και $DSPACE(s(n))$ είναι κλειστές ως προς συμπλήρωμα.

Αν θεωρήσουμε μη ντετερμινισμό, το πρόβλημα είναι ανοιχτό στην περίπτωση της χρονικής πολυπλοκότητας. Για παράδειγμα δεν γνωρίζουμε αν coNP $\neq$ NP. Μάλιστα, το τελευταίο συνδέεται και με το πρόβλημα αν P $\neq$ NP, αφού προφανώς αν coNP $\neq$ NP, τότε P $\neq$ NP.

Συμπληρωματικές κλάσεις πολυπλοκότητας III

Θεώρημα (Immerman-Szelepcsényi)

Η κλάση NSPACE$(s(n))$ *είναι κλειστή ως προς συμπλήρωμα.*

Για $s(n) = n$ έχουμε την κλάση προβλημάτων που επιλύονται από μηχανή Turing που χρησιμοποιεί γραμμικό χώρο, αλλιώς γνωστό και ως LBA (linearly bounded automaton), οπότε το παραπάνω θεώρημα έλυσε και ένα, για πολλά χρόνια, ανοικτό πρόβλημα, αν δηλαδή η κλάση των LBA (ή ισοδύναμα η κλάση των context sensitive γλωσσών, από ένα αποτέλεσμα του Kuroda, του 1964) είναι κλειστή ως προς συμπλήρωμα.

Αναγωγές I

Η έννοια της **αναγωγής** σε πολυωνυμικό χρόνο πρέπει να συνδέει μεταξύ τους προβλήματα με υπολογιστικά ``εύκολο'' τρόπο. Θεωρούμε εύκολες συναρτήσεις (και προβλήματα) που υπολογίζονται σε πολυωνυμικό χρόνο.

Θα θέλαμε:

- Αν οι συναρτήσεις f και g είναι ``εύκολες'', τότε και η σύνθεσή τους $f \circ g$ είναι ``εύκολη''.
- Αν η f είναι υπολογίσιμη σε χρόνο $O(n^2)$, τότε θεωρείται εύκολη.

Άρα θεωρούμε εύκολα προβλήματα (και συναρτήσεις) αυτά που υπολογίζονται σε πολυωνυμικό χρόνο (έστω και σε $O(n^{1000})$). Για τους παραπάνω λόγους, ορίζουμε την αναγωγή κατά Karp:

Ορισμός (Αναγωγή κατά Karp)

$$A \leqslant_m^P B: \quad \exists f \in \mathrm{FP}, \forall x (x \in A \iff f(x) \in B)$$

Υπάρχουν και άλλες χρήσιμες αναγωγές, όπως η λεγόμενη log-space, που χρησιμοποιεί λογαριθμικό χώρο, και η οποία είναι χρήσιμη για αναγωγές προβλημάτων σε μικρότερες κλάσεις πολυπλοκότητας, όπως η P:

Αναγωγές II

Ορισμός (Log-space Αναγωγή)

$$A \leqslant_m^L B: \quad \exists f \in \text{FL}, \forall x (x \in A \iff f(x) \in B)$$

Ισχύει: $A \leqslant_m^L B \implies A \leqslant_m^P B$, αλλά όχι το αντίστροφο.

Μία επιθυμητή ιδιότητα μίας αναγωγής είναι να είναι κλειστή ως προς διάφορες κλάσεις γλωσσών:

Ορισμός

Λέμε ότι μία κλάση γλωσσών C είναι **κλειστή** ως προς μία αναγωγή $\leqslant$ αν

$$A \leqslant B \wedge B \in C \implies A \in C.$$

Μερικές από τις κλάσεις πολυπλοκότητας που είναι κλειστές ως προς την αναγωγή κατά Karp ($\leqslant_m^P$) είναι οι εξής: P, PSPACE, EXP, EXPSPACE (βλέπε παραπάνω για τους ορισμούς τους).

Αναγωγές III

Ορισμός (Hardness)

Λέμε ότι A είναι C-hard (C-δύσκολο), ως προς την $\leqslant$, αν:

$$\forall B \in C : B \leqslant A.$$

Η έννοια της hardness δίνει ένα κάτω όριο για την πολυπλοκότητα ενός προβλήματος, δεδομένου ότι το πρόβλημα A είναι τουλάχιστον τόσο δύσκολο όσο οποιοδήποτε πρόβλημα μίας κλάσης C.

Ορισμός (Completeness)

Λέμε ότι A είναι C-complete (C-πλήρες), ως προς την $\leqslant$, αν:

$$A \text{ είναι } C - \text{hard ως προς } \leqslant \quad \wedge \quad A \in C.$$

Αναγωγές IV

Παρακάτω δίνουμε πλήρη προβλήματα για μερικές από τις σημαντικότερες κλάσεις πολυπλοκότητας:

- NL: το πρόβλημα Reachability (log-space αναγωγές).
- P: Circuit-Value και Linear Programming (πάλι με log-space αναγωγές).
- NP: το 3SAT.
- PSPACE: το QBF (Quantified Boolean Formula satisfiability problem).
- EXP: το $n \times n$ Go.
- EXPSPACE: το RegExp$(\cup, \cdot, *, ^2)$, που είναι το πρόβλημα ελέγχου ισοδυναμίας regular expressions, που χρησιμοποιούν τους τελεστές $\cup$ (ένωση), $\cdot$ (παράθεση), $*$ (άστρο του Kleene) και ^{2}, όπου $\alpha^2 = \alpha \cdot \alpha$.

Παράμετροι για ορισμό κλάσεων πολυπλοκότητας I

- **Concrete Complexity**: Θεωρούμε κάποιο *συγκεκριμένο* μοντέλο υπολογισμού, κάποιο *συγκεκριμένο* πρόβλημα και κάποιον *συγκεκριμένο* αλγόριθμο για το πρόβλημα σε αυτό το μοντέλο. Έτσι καθορίζουμε την ακριβή πολυπλοκότητα του αλγορίθμου (οι σταθερές φυσικά δεν παίζουν ρόλο).
- **Abstract** ή αλλιώς **structural complexity**: Θεωρούμε κλάσεις πολυπλοκότητας με διάφορες *υπολογιστικές παραμέτρους* και συγκρίνουμε τις κλάσεις μεταξύ τους (ως προς εγκλεισμό, διαχωρισμό κ.τ.λ.). Χρήσιμο για τις συγκρίσεις είναι να βρούμε αναγωγές και προβλήματα που είναι πλήρη σε αυτές τις κλάσεις ως προς αυτές τις αναγωγές.

Παράμετροι για ορισμό κλάσεων πολυπλοκότητας II

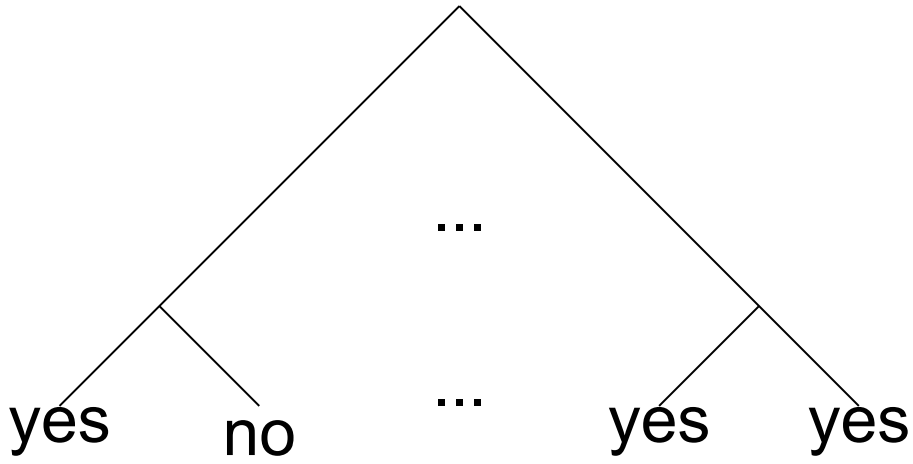
Αναφέρουμε επιγραμματικά μερικές παραμέτρους με τις οποίες ορίζονται κλάσεις πολυπλοκότητας:

- **μοντέλο υπολογισμού**: Μηχανή Turing (TM), Random Access Machine (RAM), πεπερασμένο αυτόματο, Linearly Bounded Automaton (LBA), Παράλληλη RAM (PRAM), μονότονα κυκλώματα (monotone circuits)·
- **μέθοδος λειτουργίας/αποδοχής**: ντετερμινιστική, μη ντετερμινιστική, πιθανοτική, εναλλασσόμενη (alternating), παράλληλη·
- **είδος μοντέλου/λειτουργίας**: αποφασιστής (decider), αποδέκτης (acceptor), γεννήτρια (generator), μετατροπέας (transducer)·
- **αγαθά**: αριθμός βημάτων, αριθμός συγκρίσεων, αριθμός πολλαπλασιασμών, χρόνος, χώρος μνήμης, πλήθος επεξεργαστών, αριθμός εναλλαγών στο υπολογιστικό δένδρο, μέγεθος (size) κυκλώματος, βάθος (depth) κυκλώματος·
- **άλλα εργαλεία**: τυχαιότητα (randomness), μαντεία (oracles), διαλογική αλληλεπίδραση (interactivity), υπόσχεση (promise), τελεστές (operators)·
- **φράγματα (bounds) ως προς το μήκος της εισόδου**: για παράδειγμα $O(n^3)$ ή πολυωνυμικό, time/space $(t(n), s(n))$ tradeoff (αντιστάθμισμα), Probabilistic Checkable Proofs: PCP$(r(n), q(n))$ (με χρήση $r(n)$ τυχαίων bits και $q(n)$ queries, ερωτήσεων, στην απόδειξη).

Μοντέλα δένδρων υπολογισμού για TM I

- Για να μελετήσουμε την συμπεριφορά μηχανών Turing, θα κωδικοποιήσουμε τους υπολογισμούς μίας μηχανής Turing με ένα **δέντρο υπολογισμού**.
- Ο υπολογισμός ξεκινά στην **ρίζα** του δένδρου.
- Θεωρούμε ότι αν σε κάποιο σημείο του υπολογισμού έχουμε μία μη ντετερμινιστική επιλογή τότε έχουμε μία **διακλάδωση** στο δένδρο.
- Στα **φύλλα** της μηχανής TM έχουμε τις απαντήσεις της μηχανής Turing. Κάθε μονοπάτι από την ρίζα του δένδρου μέχρι κάποιο φύλλο επομένως κωδικοποιεί έναν πιθανό υπολογισμό
- Χωρίς βλάβη της γενικότητας, υποθέτουμε ότι το δένδρο είναι δυαδικό, πλήρες και γεμάτο. όλα τα φύλλα του είναι στο ίδιο επίπεδο.

Μοντέλα δένδρων υπολογισμού για TM II



Σχήμα: Μοντέλο δένδρων υπολογισμού

Επίσης, έχει ενδιαφέρον το μήκος του υπολογιστικού μονοπατιού από την ρίζα μέχρι το φύλλο να έχει **πολυωνυμικό μήκος** ως προς το μήκος της εισόδου (να αντιστοιχεί δηλαδή το κάθε μονοπάτι σε κάποιον <<εύκολο>>, δηλαδή πολυωνυμικό, υπολογισμό).

Θεωρώντας το παραπάνω μοντέλο, θα ορίσουμε μερικές από τις γνωστές κλάσεις υπολογισμού, καθώς και μερικές καινούριες. Πιο συγκεκριμένα, Θα χρησιμοποιήσουμε ποσοδείκτες ($\exists$, $\forall$) στα μονοπάτια. Επειδή εννοείται πάντοτε ο περιορισμός του μήκους των μονοπατιών, θα γράφουμε π.χ. $\exists y$, αντί για $\exists y\colon |y| \leqslant p(|x|)$, όπου y: μεταβλητή για τα μονοπάτια, x: μεταβλητή για την είσοδο, p: πολυώνυμο.

Μοντέλα δένδρων υπολογισμού για TM III

Για παράδειγμα, η κλάση P μπορεί να περιγραφεί ως εξής:

$$L \in \mathrm{P} \iff \exists R \in \mathrm{P}: \begin{cases} x \in L & \implies \forall y R(x,y) \\ x \notin L & \implies \forall y \neg R(x,y) \end{cases}$$

Περισσότερο ενδιαφέρον έχει η κλάση NP που μπορεί να περιγραφεί ως εξής:

$$L \in \mathrm{NP} \iff \exists R \in \mathrm{P}: \begin{cases} x \in L & \implies \exists y R(x,y) \\ x \notin L & \implies \forall y \neg R(x,y) \end{cases}$$

Δηλαδή, αν $x \in L$ υπάρχει τουλάχιστον ένας υπολογισμός που αποδέχεται, ενώ αν $x \notin L$ κανένας υπολογισμός δεν αποδέχεται.

Μοντέλα δένδρων υπολογισμού για TM IV

Παρομοίως, η κλάση coNP περιγράφεται ως εξής:

$$L \in \text{coNP} \iff \exists R \in \text{P}: \begin{cases} x \in L & \implies \forall y R(x,y) \\ x \notin L & \implies \exists y \neg R(x,y) \end{cases}$$

Παρατηρούμε ότι οι ποσοδείκτες που χρησιμοποιούνται και αντιστοιχούν στο $<<x \in L>>$ και στο $<<x \notin L>>$ καθορίζουν πλήρως την αντίστοιχη κλάση πολυπλοκότητας. Έτσι, εισάγουμε τον παρακάτω συμβολισμό:

$$\text{P} = (\forall, \forall), \quad \text{NP} = (\exists, \forall), \quad \text{coNP} = (\forall, \exists).$$

Τυχαιότητα (Randomness) I

- Χρησιμοποιώντας το μοντέλο **δένδρων υπολογισμού**, θα ορίσουμε κλάσεις πολυπλοκότητας που βασίζονται στις **πιθανότητες**, με βάση τυχαίες επιλογές.
- Αυτή η προσέγγιση είναι πολύ χρήσιμη από πρακτική άποψη, αφού σε πολλές εφαρμογές, είναι ικανοποιητικός ένας αλγόριθμος ο οποίος κάνοντας κάποιες τυχαίες επιλογές, δίνει στις περισσότερες των περιπτώσεων το σωστό αποτέλεσμα.
- Ένας **πιθανοκρατικός αλγόριθμος** είναι συνήθως πιο απλός στην διατύπωσή του και στην πράξη πιο αποδοτικός από έναν αντίστοιχο ντετερμινιστικό που επιλύει το ίδιο πρόβλημα. Για παράδειγμα, απλοί πιθανοκρατικοί αλγόριθμοι για τον έλεγχο αν ένας αριθμός είναι πρώτος υπάρχουν από την δεκαετία του 1970 και χρησιμοποιούνται στην πράξη έναντι πιο περίπλοκων ντετερμινιστικών τύπου AKS.
- Στα πλαίσια του μοντέλου δένδρων υπολογισμού, θα θεωρήσουμε ότι η επιλογή σε κάθε κόμβο του δένδρου γίνεται τυχαία με πιθανότητα $1/2$ για κάθε παιδί του κόμβου. Για να δείξουμε ότι η *<<συντριπτική>> πλειοψηφία* των υπολογισμών δίνει το σωστό αποτέλεσμα, εισάγουμε έναν νέο ποσοδείκτη, τον $\exists^+$.

Τυχαιότητα (Randomness) II

- Με την βοήθεια του $\exists^+$, ορίζουμε την κλάση BPP, από το **Bounded error Probabilistic Polynomial**:

Ορισμός ($\mathrm{BPP} = (\exists^+, \exists^+)$)

$$L \in \mathrm{BPP} \iff \exists R \in \mathrm{P}: \begin{cases} x \in L & \implies \exists^+ y R(x,y) \\ x \notin L & \implies \exists^+ y \neg R(x,y) \end{cases}$$

Τυχαιότητα (Randomness) III

- Με άλλα λόγια, σε ένα δέντρο για την κλάση BPP έχουμε την <<συντριπτική>> πλειοψηφία των φύλλων να δίνει το σωστό αποτέλεσμα. Στον παραπάνω ορισμό, δεν έχει μεγάλη σημασία ο ακριβής ορισμός της <<συντριπτικής>> πλειοψηφίας, αλλά πρέπει να είναι οπωσδήποτε *φραγμένος* (εξ ου και το `bounded' του BPP) πάνω από το $1/2$. Το ποσοστό της πλειοψηφίας μπορεί να είναι, ενδεικτικά, μεγαλύτερο από $1/2+\varepsilon$, $1/2+1/p(|x|)$, $2/3$, 99%, $1-2^{-p(|x|)}$ (όπου $p(|x|)>1$) ($2^{-p|(x|)}$ καλείται αμελητέα ποσότητα). Αυτή η δυνατότητα επιλογής υπάρχει, επειδή με πολυωνυμικές επαναλήψεις του αντίστοιχου αλγορίθμου, είναι δυνατόν να αυξήσουμε την πιθανότητα επιτυχίας, όσο θέλουμε. Αλγόριθμοι BPP ονομάζονται **Monte Carlo** ή αλλιώς two-sided error, επειδή ανεξάρτητα από το αποτέλεσμα (ναι ή όχι), υπάρχει κάποια πιθανότητα λάθους. Είναι προφανές ότι η κλάση BPP είναι κλειστή ως προς συμπλήρωμα.
- Ας θεωρήσουμε τώρα αλγορίθμους οι οποίοι κάνουν λάθος μόνον για την μία απάντηση (one sided error). Έτσι, προκύπτει η κλάση RP (**Randomized Polynomial**):

Ορισμός (RP $= (\exists^+, \forall)$)

$$L \in \mathrm{RP} \iff \exists R \in \mathrm{P}: \begin{cases} x \in L & \implies \exists^+ y R(x,y) \\ x \notin L & \implies \forall y \neg R(x,y) \end{cases}$$

Τυχαιότητα (Randomness) IV

Σε αυτήν την κλάση, αν ο αντίστοιχος RP αλγόριθμος δώσει απάντηση <<ναι>> (δηλαδή το κατηγόρημα R υπολογιστεί αληθές), είμαστε σίγουροι ότι $x \in L$. Αντίθετα, η απάντηση <<όχι>> του RP αλγορίθμου δεν είναι <<σίγουρη>>.

Προφανώς, ισχύουν: $\text{RP} \subseteq \text{BPP}$, $\text{coRP} \subseteq \text{BPP}$, αλλά δεν γνωρίζουμε αν $\text{RP} = \text{coRP}$.

- Μία άλλη πολύ χρήσιμη κλάση, είναι αυτή που ορίζεται με τομή των RP και coRP, η $\text{ZPP} = \text{RP} \cap \text{coRP}$. Η ονομασία προέρχεται από το **Zero error Probabilistic Polynomial**, γιατί μπορεί εύκολα να δειχτεί ότι ένα πρόβλημα είναι στο ZPP αν υπάρχει πιθανοκρατικός αλγόριθμος ο οποίος τρέχει σε αναμενόμενο πολυωνυμικό χρόνο και δίνει πάντοτε σωστή απάντηση. Πράγματι, αν ένα πρόβλημα είναι στο ZPP, σημαίνει ότι έχουμε ένα RP και έναν coRP αλγόριθμο για αυτό, οπότε αρκεί να τρέχουμε εναλλακτικά τους δύο αλγορίθμους, μέχρι ο ένας να δώσει την <<σίγουρή>> του απάντηση. Βέβαια, μπορεί να χρειαστεί να τρέχουμε εναλλακτικά τους δύο αλγορίθμους για πάντα, αλλά με μεγάλη πιθανότητα θα έχουμε μία <<σίγουρη>> απάντηση, μετά από μερικές επαναλήψεις. Εναλλακτικά, μπορούμε να πούμε ότι ένας ZPP αλγόριθμος έχει τρεις εξόδους: <<ναι>>, <<όχι>> (για τις <<σίγουρες>> απαντήσεις), και <<δεν ξέρω>> (για τις όχι <<σίγουρες>>).

Οι αλγόριθμοι στο ZPP ονομάζονται **Las Vegas**.

Τυχαιότητα (Randomness) V

- Δεδομένου ότι υπάρχουν αρκετοί πιθανοκρατικοί αλγόριθμοι ευρείας χρήσης για πρακτικά προβλήματα, πολλοί τοποθετούν τους εφικτούς (feasible) υπολογισμούς πάνω από το P, στις πιθανοτικές κλάσεις BPP, RP, ZPP.

  Πάντως, δεν γνωρίζουμε αν υπάρχουν πλήρη προβλήματα για τις κλάσεις που ορίστηκαν παραπάνω (BPP, RP, ZPP).
- Αν τώρα το ποσοστό λάθους ενός πιθανοκρατικού αλγορίθμου δεν φραχθεί μακριά από το $1/2$, τότε έχουμε απλώς την βεβαιότητα ότι στο μοντέλο δένδρων υπολογισμού παραπάνω από τα μισά υπολογιστικά μονοπάτια δίνουν την σωστή απάντηση. Για να δηλώσουμε το παραπάνω χρησιμοποιούμε τον ποσοδείκτη $\exists_{1/2}$. Για unbounded two-sided error, έχουμε την κλάση PP (Probabilistic Polynomial):

Ορισμός ($\mathrm{PP} = (\exists_{1/2}, \exists_{1/2})$)

$$L \in \mathrm{PP} \iff \exists R \in \mathrm{P}: \begin{cases} x \in L & \implies \exists_{1/2} y R(x,y) \\ x \notin L & \implies \exists_{1/2} y \neg R(x,y) \end{cases}$$

Τυχαιότητα (Randomness) VI

- Λόγω της έλλειψης φράγματος για την πιθανότητα λάθους, δεν μπορούμε να χρησιμοποιήσουμε την τεχνική της επανάληψης για να βελτιώσουμε την πιθανότητα σωστού αποτελέσματος από έναν PP αλγόριθμο. Μία άλλη ένδειξη για το ανέφικτο της κλάσης PP σε σχέση με τις BPP, RP, ZPP, προκύπτει από το παρακάτω αποτέλεσμα:

Πρόταση

NP $\subseteq$ PP.

- Πρέπει επίσης να σημειώσουμε ότι δεν λάβαμε καθόλου υπ' όψιν μας, ως υπολογιστικό πόρο, των αριθμό των τυχαίων bits που χρησιμοποιεί ένας πιθανοκρατικός αλγόριθμος. Στην πράξη, κάθε <<τυχαίο>> bit που χρειαζόμαστε δεν είναι χωρίς τίμημα, αφού το λαμβάνουμε από κάποια γεννήτρια *ψευδοτυχαίων* bits.
- Τέλος, αναφέρουμε και την κλάση RL (**Randomized Logspace**) που περιέχει τα προβλήματα που έχουν one-sided error αλγόριθμο που χρησιμοποιεί λογαριθμικό χώρο και πολυωνυμικό ως προς το μήκος της εισόδου αριθμό τυχαίων bits.

Πολυωνυμική Ιεραρχία Ι

Αντίστοιχα με προηγούμενο κεφάλαιο, όπου ορίσαμε μαντεία και την αριθμητική ιεραρχία, θα ορίσουμε την **πολυωνυμική ιεραρχία**, η οποία έχει παρόμοια δομή, αλλά βρίσκεται πολύ χαμηλότερα από άποψη υπολογιστικής πολυπλοκότητας. Υπενθυμίζουμε την έννοια του υπολογισμού με μαντείο: Ένας αλγόριθμος χρησιμοποιεί ένα μαντείο για το πρόβλημα Π, αν έχει την δυνατότητα κατά την διάρκεια του υπολογισμού, να ρωτάει το μαντείο για κάποιο στιγμιότυπο x του προβλήματος Π, αν $x \in \Pi$, και το μαντείο να του απαντά άμεσα με ένα <<ναι>> ή με ένα <<όχι>>. Όσο δύσκολο και να είναι το πρόβλημα Π, ο αλγόριθμος δεν σπαταλά επιπλέον υπολογιστικούς πόρους.

Πολυωνυμική Ιεραρχία ΙΙ

Ορισμός (Κλάσεις με μαντεία)

- $\mathcal{C}^{\Pi}$: η κλάση των προβλημάτων τα οποία λύνονται με αλγόριθμο στην κλάση $\mathcal{C}$ ο οποίος χρησιμοποιεί μαντείο για το πρόβλημα Π
- $\mathcal{C}^{\mathcal{C}_o} = \bigcup_{\Pi \in \mathcal{C}_o} \mathcal{C}^{\Pi}$

Για παράδειγμα, η κλάση $\mathrm{P}^{\mathrm{SAT}}$ αποτελείται από τα προβλήματα που λύνονται με ντετερμινιστικό πολυωνυμικό αλγόριθμο ο οποίος χρησιμοποιεί μαντείο για το πρόβλημα SAT. Άλλη περιγραφή της ίδιας κλάσης είναι: P^{NP} (αφού το SAT είναι NP-πλήρες).

Πολυωνυμική Ιεραρχία ΙΙΙ

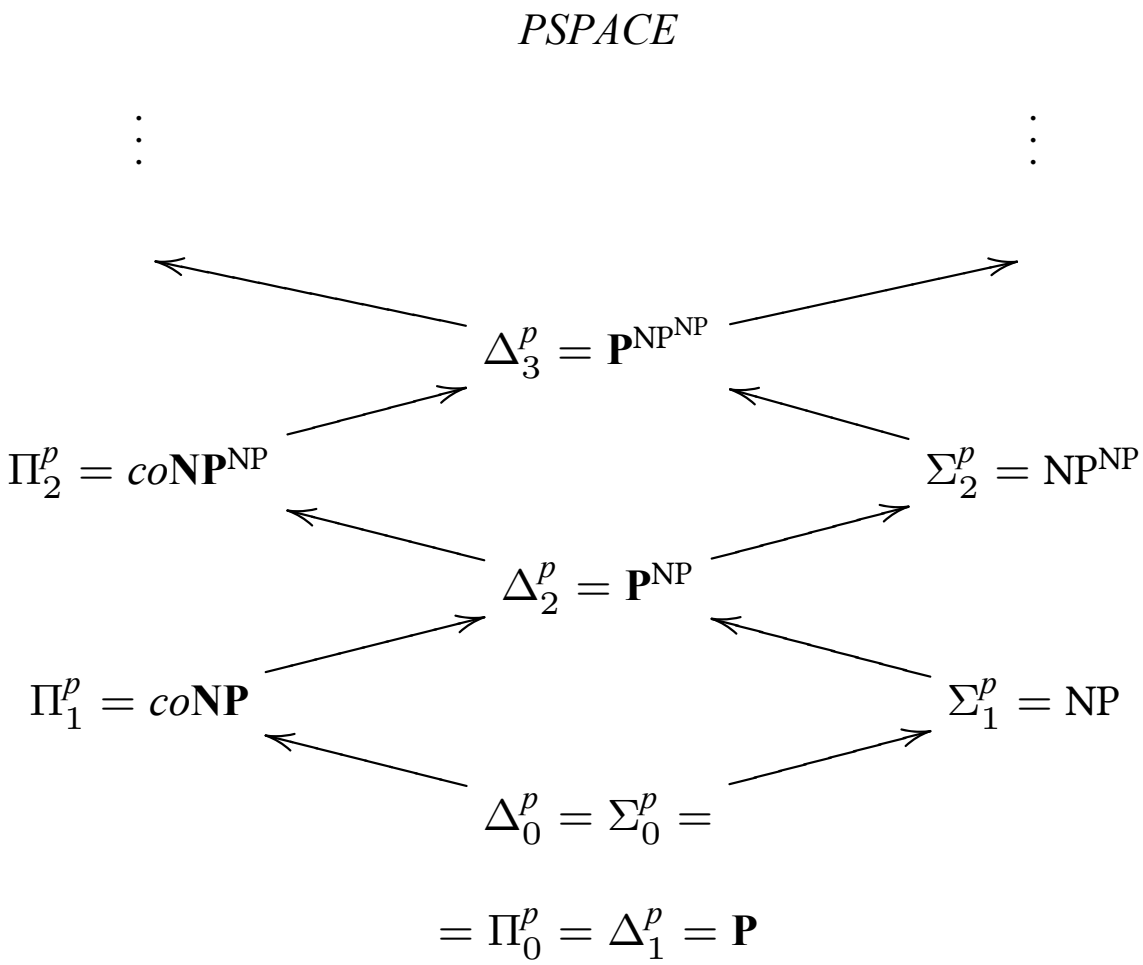
Ορισμός

($k \geqslant 0$)

- $\Sigma_0^p = \Pi_0^p = \Delta_0^p = \mathrm{P}$
- $\Sigma_{k+1}^p = \mathrm{NP}^{\Sigma_k^p}$, $\Pi_{k+1}^p = \mathrm{co}\Sigma_{k+1}^p$, $\Delta_{k+1}^p = \mathrm{P}^{\Sigma_k^p}$, $\Delta\Sigma_k^p = \Sigma_k^p \cap \Pi_k^p$
- Πολυωνυμική ιεραρχία: $\mathrm{PH} = \bigcup_{k\in\mathbb{N}} \Sigma_k^p$

Ισχύουν τα παρακάτω: $\Sigma_1^p = \mathrm{NP}$, $\Pi_1^p = \mathrm{coNP}$ και για κάθε $k \geqslant 0$: $\Sigma_k^p \subseteq \Sigma_{k+1}^p$ και $\Pi_k^p \subseteq \Sigma_{k+1}^p$. Αν και δεν έχει αποδειχθεί το αυστηρό των παραπάνω εγκλεισμών (όπως στην αριθμητική ιεραρχία), εν τούτοις πιστεύουμε ότι η ιεραρχία είναι *αυστηρή* (strict). Αν η PH δεν είναι αυστηρή, τότε θα υπάρχει κάποιο k για το οποίο $\mathrm{PH} = \Sigma_k^p$, οπότε λέμε ότι *η πολυωνυμική ιεραρχία καταρρέει στο k-οστό επίπεδο* (collapses at the k-th level).

Πολυωνυμική Ιεραρχία IV



Σχήμα: Πολυωνυμική ιεραρχία

Πολυωνυμική Ιεραρχία --- Εναλλαγή Ποσοδεικτών I

Ένας εναλλακτικός τρόπος ορισμού της πολυωνυμικής ιεραρχίας είναι με την βοήθεια εναλλαγής ποσοδεικτών (∃ και ∀). Επισημαίνουμε ότι, σε κάθε περίπτωση, οι ποσοδείκτες αναφέρονται σε αντικείμενα το μέγεθος των οποίων είναι φραγμένο από κάποιο πολυώνυμο p ως προς το μήκος της εισόδου.

Πρόταση

$L \in \Sigma_k^p$ ανν υπάρχει κατηγόρημα R υπολογιζόμενο σε πολυωνυμικό χρόνο και πολυώνυμο p που φράσσει το μέγεθος των αντικειμένων των ποσοδεικτών, τέτοια ώστε:

$$x \in L \iff \exists y_1 \forall y_2 \ldots Q y_k \, R(x, y_1, y_2, \ldots, y_k),$$

όπου $Q = \begin{cases} \exists, & k \text{ περιττό} \\ \forall, & k \text{ άρτιο} \end{cases}$.

Πολυωνυμική Ιεραρχία --- Εναλλαγή Ποσοδεικτών

Παρομοίως, για την κλάση Π_k^p μόνο που τώρα η ακολουθία των ποσοδεικτών αρχίζει από $\forall$:

Πρόταση

$L \in \Pi_k^p$ ανν υπάρχει κατηγόρημα R υπολογιζόμενο σε πολυωνυμικό χρόνο και πολυώνυμο p που φράσσει το μέγεθος των αντικειμένων των ποσοδεικτών, τέτοια ώστε:

$$x \in L \iff \forall y_1 \exists y_2 \ldots Q y_k \, R(x, y_1, y_2, \ldots, y_k),$$

όπου $Q = \begin{cases} \forall, & k \text{ περιττό} \\ \exists, & k \text{ άρτιο} \end{cases}$.

Πολυωνυμική Ιεραρχία --- Alternating TM I

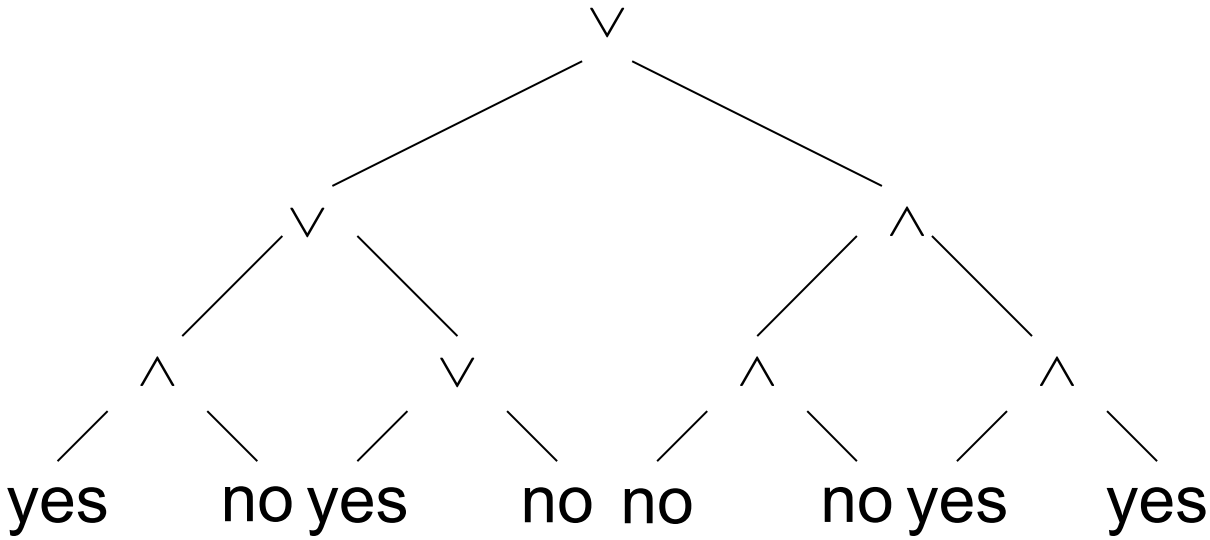
Η εναλλαγή των ποσοδεικτών στην πολυωνυμική ιεραρχία δίνει το έναυσμα για τον ορισμό της μηχανής Turing με εναλλασσόμενη λειτουργία. Αν θεωρήσουμε την δενδρική αναπαράσταση των υπολογισμών μίας NP μηχανής Turing, η μηχανή απαντά ναι, αν υπάρχει ένα τουλάχιστον φύλλο που λέει <<ναι>>. Μπορούμε να θεωρήσουμε ότι κάθε κόμβος στο δένδρο υπολογίζει την διάζευξη ($\vee$) των αποτελεσμάτων από τα παιδιά του και την προωθεί στον γονέα του (τα φύλλα απλώς προωθούν προς τα πάνω), μέχρι το αποτέλεσμα να φτάσει στην ρίζα. Αντίστοιχα, μία coNP μηχανή Turing αποδέχεται όταν στην δενδρική αναπαράσταση όλα τα φύλλα λένε <<ναι>>, οπότε μπορούμε να θεωρήσουμε ότι ο κάθε κόμβος συλλέγει τα αποτέλεσμα των παιδιών του και προωθεί στον γονέα του την σύζευξη ($\wedge$) των αποτελεσμάτων από τα παιδιά του, πάλι μέχρι το σωστό αποτέλεσμα να φτάσει στην ρίζα. Λέμε ότι όλοι οι κόμβοι στο δένδρο μίας NP μηχανής είναι τύπου $\vee$, ή $\exists$, ή υπαρξιακού. Λέμε ότι όλοι οι κόμβοι στο δένδρο μίας coNP μηχανής είναι τύπου $\wedge$, ή $\forall$, ή καθολικού.

Πολυωνυμική Ιεραρχία --- Alternating TM II

Μία **εναλλασσόμενη μηχανή Turing** είναι μία μηχανή Turing στην οποία το αντίστοιχο υπολογιστικό δένδρο έχει εσωτερικούς κόμβους τύπου $\vee$ ή $\wedge$. Σημασία έχει το **πλήθος εναλλαγών τύπου**. Το μέγιστο **πλήθος εναλλαγών** (σε ένα μονοπάτι), που πιθανώς να είναι φραγμένο, αποτελεί μέτρο των δυνατοτήτων μίας τέτοιας μηχανής.

Για παράδειγμα, το υπολογιστικό δένδρο του παρακάτω σχήματος έχει πλήθος εναλλαγών τύπου ίσο με 2.

Πολυωνυμική Ιεραρχία --- Alternating TM III



Σχήμα: Υπολογιστικό δένδρο με εναλλαγές

Πολυωνυμική Ιεραρχία --- Alternating TM IV

Μπορεί να δειχθεί ότι η πολυωνυμική ιεραρχία είναι ακριβώς η κλάση των γλωσσών που γίνεται αποδεκτή από μηχανές Turing που έχουν φραγμένο πλήθος εναλλαγών. Πιο συγκεκριμένα:

- $L \in \Sigma_k^p$ ανν η L γίνεται αποδεκτή από μηχανή Turing που έχει το πολύ k εναλλαγές τύπου και αρχίζει με τύπο $\vee$.
- $L \in \Pi_k^p$ ανν η L γίνεται αποδεκτή από μηχανή Turing που έχει το πολύ k εναλλαγές τύπου και αρχίζει με τύπο $\wedge$.

Παραλληλοποιήσιμα προβλήματα I

Για να μελετήσουμε παράλληλους υπολογισμούς θα εισάγουμε ένα νέο μοντέλο υπολογισμού, το **κύκλωμα**.

Ένα κύκλωμα είναι ένας κατευθυνόμενος ακυκλικός γράφος, στον οποίο έχουμε ένα σύνολο κόμβων **εισόδου** και έναν κόμβο που είναι η **έξοδος**. Θεωρούμε ότι οι είσοδοι στο κύκλωμα είναι αληθοτιμές (ή αλλιώς οι τιμές 0 και 1) και κάθε εσωτερικός κόμβος αντιστοιχεί σε μία λογική συνάρτηση (ή αλλιώς πύλη --- gate) με πλήθος εισόδων, όσες οι ακμές που καταλήγουν σε αυτόν. Αν ένα κύκλωμα C έχει n εισόδους του ενός bit: $x_1, x_2, ..., x_n$, τότε για κάθε $x \in \{0,1\}^n$, υπολογίζει μία μοναδική τιμή στην έξοδο, την $C(x)$. Αν $C(x) = 1$, λέμε ότι το κύκλωμα C αποδέχεται την είσοδο των n bit: x.

Παραλληλοποιήσιμα προβλήματα II

Η αναντιστοιχία του παραπάνω ορισμού αποδοχής, σε σχέση με την αποδοχή, ας πούμε, σε μία μηχανή Turing, είναι ότι ένα κύκλωμα, λόγω της αμετάβλητης φύσης του, μπορεί να απαντά για εισόδους μήκους ακριβώς n, ενώ μία μηχανή Turing (ή ένας αλγόριθμος γενικότερα) δέχεται εισόδους οποιουδήποτε μεγέθους. Για τον λόγο αυτό θα θεωρούμε μία οικογένεια κυκλωμάτων $\{C_1, C_2, \dots\}$, όπου κάθε C_n έχει n κόμβους εισόδου. Η **γλώσσα** που αποδέχεται μία οικογένεια κυκλωμάτων είναι η

$$L(C) = \{x \mid C_{|x|}(x) = 1\}.$$

Το πρόβλημα είναι ότι οι οικογένειες κυκλωμάτων (σε αντίθεση με τις μηχανές Turing) δεν είναι αριθμήσιμες.

Παραλληλοποιήσιμα προβλήματα III

Για να ξεπεράσουμε την παραπάνω δυσκολία, θα περιοριστούμε σε **ομοιόμορφες οικογένειες κυκλωμάτων** (uniform circuit families). Για αυτές υπάρχει αλγόριθμος και μάλιστα αποδοτικός ο οποίος δεδομένου n κατασκευάζει την αναπαράσταση του κυκλώματος C_n της οικογένειας. Μία επιλογή είναι οι P-ομοιόμορφες οικογένειες, που χρησιμοποιούν πολυωνυμικό αλγόριθμο κατασκευής. Επειδή όμως τα κυκλώματα χρησιμοποιούνται συνήθως για ορισμό κλάσεων χαμηλότερα από το P, θα χρησιμοποιήσουμε μία άλλη πιο περιορισμένη έννοια ομοιομορφίας:

Ορισμός

Μία οικογένεια κυκλωμάτων είναι DLOGTIME-ομοιόμορφη αν υπάρχει μηχανή Turing (με δυνατότητα τυχαίας προσπέλασης στην ταινία εισόδου) που απαντά τις παρακάτω ερωτήσεις σε χρόνο $O(\log n)$:

- Υπάρχει σύνδεση από τον κόμβο u στον κόμβο v στο C_n;
- Τι είδους πύλη έχει ο κόμβος u;

Παραλληλοποιήσιμα προβλήματα IV

Το *μέγεθος* (size) ενός κυκλώματος είναι το πλήθος των κόμβων που περιέχει ο αντίστοιχος γράφος. Το μέγεθος αποτελεί μέτρο του κόστους κατασκευής του κυκλώματος και συνήθως δεν θεωρείται περισσότερο από πολυωνυμικό ως προς το μήκος της εισόδου. Το μέγεθος, όμως, δεν αποτελεί και πολύ καλό μέτρο του χρόνου υπολογισμού σε ένα κύκλωμα, επειδή εν γένει πολλές λογικές πύλες λειτουργούν παράλληλα. Οι πύλες που πρέπει να περιμένουν διαδοχικά ενδιάμεσα αποτελέσματα είναι αυτές που βρίσκονται σε κάθε μονοπάτι από μία είσοδο στην έξοδο. Για τον λόγο αυτό, πιο σημαντικό είναι το *βάθος* (depth) ενός κυκλώματος, που ορίζεται ως το μήκος του μακρύτερου μονοπατιού από είσοδο στην έξοδο.

Παραλληλοποιήσιμα προβλήματα V

Επίσης, σημαντικό είναι το είδος των λογικών πυλών που χρησιμοποιούνται σε κάθε κύκλωμα. Πιο συγκεκριμένα, θεωρούμε τα παρακάτω είδη λογικών πυλών:

1. Λογικές πύλες με περιορισμένο αριθμό εισόδων (bounded fan-in), καθώς και εναδικές πύλες $\neg$. Αρκεί να θεωρήσουμε δυαδικές πύλες $\wedge$ και $\vee$ (μαζί με τις εναδικές πύλες $\neg$).
2. Λογικές πύλες $\wedge$ και $\vee$ με απεριόριστο αριθμό εισόδων (unbounded fan-in), καθώς και εναδικές πύλες $\neg$.
3. Πύλες κατωφλίου (threshold) με απεριόριστο αριθμό εισόδων, καθώς και εναδικές πύλες $\neg$. Αρκεί, αντί για γενικές πύλες κατωφλίου, να χρησιμοποιήσουμε την πύλη πλειοψηφίας (majority gate), που δίνει στην έξοδο 1 αν και μόνον αν τουλάχιστον $r/2$ από τις r εισόδους της είναι 1.

Παραλληλοποιήσιμα προβλήματα VI

Με βάση τα παραπάνω μπορούμε να ορίσουμε τις παρακάτω κλάσεις:

Ορισμός

($k \geqslant 0$):

1. NC^k: η κλάση των γλωσσών που γίνονται αποδεκτές από DLOGTIME-ομοιόμορφες οικογένειες κυκλωμάτων πολυωνυμικού μεγέθους και βάθους $O(\log^k n)$, με χρήση πυλών του 1ου είδους (bounded fan-in).
2. AC^k: η κλάση των γλωσσών που γίνονται αποδεκτές από DLOGTIME-ομοιόμορφες οικογένειες κυκλωμάτων πολυωνυμικού μεγέθους και βάθους $O(\log^k n)$, με χρήση πυλών του 2ου είδους (unbounded fan-in).
3. TC^k: η κλάση των γλωσσών που γίνονται αποδεκτές από DLOGTIME-ομοιόμορφες οικογένειες κυκλωμάτων πολυωνυμικού μεγέθους και βάθους $O(\log^k n)$, με χρήση πυλών του 3ου είδους (threshold gates).
4. SC^k: η κλάση των γλωσσών που γίνονται αποδεκτές από DTM σε πολυωνυμικό χρόνο και σε $O(\log^k n)$ χώρο.

Παραλληλοποιήσιμα προβλήματα VII

Επίσης, ορίζεται $\mathrm{NC} = \bigcup_{k \in \mathbb{N}} \mathrm{NC}^k$. Η τελευταία κλάση λέγεται και Nick's Class, από τον Nicholas Pippenger, που ήταν από τους πρώτους που μελέτησε τέτοια κυκλώματα. Στην πραγματικότητα, πολλά άλλα μοντέλα παράλληλου υπολογισμού (π.χ. PRAM), εκτός από τα κυκλώματα, μπορούν να χρησιμοποιηθούν για τον ορισμό της κλάσης NC, κάτι που αποτελεί ένδειξη για την ευρωστία της κλάσης και την στενή σχέση της με τα παραλληλοποιήσιμα προβλήματα.

Το <<A>> στην AC^k οφείλεται στην εναλλαγή (alternation), αφού αποδεικνύεται ότι η κλάση AC^k, για $k \geqslant 1$, είναι ακριβώς οι γλώσσες που γίνονται αποδεκτές από εναλλασσόμενη μηχανή Turing που χρησιμοποιεί $O(\log n)$ χώρο και κάνει το πολύ $O(\log^k n)$ εναλλαγές. Το <<T>> στην TC^k προέρχεται από το threshold. Η ονομασία SC, ``Steve's Class'', προέρχεται από τον Steve Cook.

Παραλληλοποιήσιμα προβλήματα VIII

Πιο συγκεκριμένα οι κλάσεις σχετίζονται μεταξύ τους ως εξής:

Θεώρημα

Για κάθε $k \geqslant 0$, $\mathrm{NC}^k \subseteq \mathrm{AC}^k \subseteq \mathrm{TC}^k \subseteq \mathrm{NC}^{k+1}$.

Σε σχέση με άλλες γνωστές κλάσεις, ισχύει:

Θεώρημα

$\mathrm{Regular} \subseteq \mathrm{NC}^1 \subseteq \mathrm{L} = \mathrm{SC}^1 \subseteq \mathrm{NL} \subseteq \mathrm{AC}^1$.
$\mathrm{Regular} \subset \mathrm{CF} \subset \mathrm{AC}^1$.

Δηλαδή το πρόβλημα του ελέγχου αν μία συμβολοσειρά παράγεται από δεδομένη γλώσσα χωρίς συμφραζόμενα (context free) ανήκει στην κλάση NC^2.

Διαλογική αλληλεπίδραση (interactivity) I

Διαλογικά συστήματα αποδείξεων (IP)

Ας θεωρήσουμε έναν **αποδείκτη (prover)** που προσπαθεί να αποδείξει την αλήθεια μίας πρότασης του τύπου $<<x \in L>>$ σε κάποιον άλλο, που τον ονομάζουμε **επαληθευτή (verifier)**.

Ο αποδείκτης είναι παντοδύναμος, με την έννοια ότι είναι ένας αλγόριθμος χωρίς περιορισμούς στο μέγεθος των αγαθών που χρησιμοποιεί (χρόνος, χώρος). Αντίθετα, ο επαληθευτής είναι απλώς ένας πιθανοτικός αλγόριθμος πολυωνυμικού χρόνου.

Ο επαληθευτής και ο αποδείκτης συμμετέχουν σε ένα πρωτόκολλο επικοινωνίας στέλνοντας μηνύματα. Ανάλογα με τα μηνύματα που λαμβάνει ο V από τον P, ο V αποδέχεται την απόδειξη, αλλιώς την απορρίπτει. Ο αποδείκτης μπορεί να μην είναι έντιμος, και να θέλει να πείσει τον επαληθευτή ότι $<<x \in L>>$, ακόμη και για x για τα οποία $<<x \notin L>>$. Ο επαληθευτής, απέναντι στον παντοδύναμο αποδείκτη, μπορεί να χρησιμοποιήσει εκτός του πολυωνυμικού χρόνου, κυρίως την τυχαιότητα που διαθέτει.

Διαλογική αλληλεπίδραση (interactivity) II

Διαλογικά συστήματα αποδείξεων (IP)

Η κλάση IP ορίστηκε από τους Goldwasser, Micali, Rackoff:

Ορισμός

$L \in$ IP:

- $x \in L \implies$ υπάρχει αποδείκτης (prover) P, ώστε ο επαληθευτής (verifier) V πάντοτε αποδέχεται (δηλαδή έχουμε πιθανότητα αποδοχής ίση με 1)·
- $x \notin L \implies$ για κάθε αποδείκτη (prover) P, ο επαληθευτής V δεν αποδέχεται με συντριπτική πιθανότητα.

Ας θεωρήσουμε το **πρόβλημα μη ισομορφισμού γράφων**: <<Δίνονται δύο γράφοι. Είναι μη ισομορφικοί;>>. Αυτό το πρόβλημα ανήκει στο coNP. Θα δώσουμε ένα πρωτόκολλο για το πρόβλημα μη ισομορφισμού γράφων, που θα δείχνει ότι το πρόβλημα είναι στο IP.

Αρχικά, ο επαληθευτής έχει τους δύο γράφους G_1 και G_2. Επιλέγει τυχαία έναν από τους δύο, έστω των G_i, και υπολογίζει έναν τυχαίο ισομορφικό γράφο του G_i, έστω τον H (αυτό γίνεται διαλέγοντας τυχαία μία μετάθεση των n κορυφών του γράφου G_i). Στέλνει τον γράφο H στον αποδείκτη, ζητώντας ένα j

Διαλογική αλληλεπίδραση (interactivity) III

Διαλογικά συστήματα αποδείξεων (IP)

τέτοιο ώστε ο G_j να είναι ισομορφικός του H. Ο αποδείκτης απαντά με ένα $j \in \{1, 2\}$. Ο επαληθευτής αποδέχεται αν όντως $i = j$, αλλιώς απορρίπτει.

Στην περίπτωση που όντως οι G_1, G_2 είναι μη ισομορφικοί, ο P, αφού είναι παντοδύναμος, βρίσκει με ποιον (μοναδικό) γράφο είναι ισομορφικός ο H που του έστειλε ο V και δίνει την σωστή τιμή για να αποδεχθεί ο V. Αν τώρα οι G_1, G_2 είναι ισομορφικοί, ο P αδυνατεί να συμπεράνει από ποιον γράφο προήλθε ο ισομορφικός H, άρα δεν μπορεί να κάνει κάτι καλύτερο από το να στείλει τυχαία ένα από τα $\{1, 2\}$ στον V. Έτσι, αν οι δύο γράφοι είναι μη ισομορφικοί ο V δεν αποδέχεται με πιθανότητα $1/2$.

Τα παραπάνω σκιαγραφούν μία απόδειξη ότι το πρόβλημα μη ισομορφισμού γράφων ανήκει στην IP.

Στην πραγματικότητα, κάθε γλώσσα στην πολυωνυμική ιεραρχία έχει πρωτόκολλο IP. Μάλιστα, έχει αποδειχθεί το ακόμη ισχυρότερο αποτέλεσμα:

Θεώρημα (Shamir)

IP = PSPACE

Διαλογική αλληλεπίδραση (interactivity) IV

Διαλογικά συστήματα αποδείξεων (IP)

Τι γίνεται όμως στην περίπτωση που ο επαληθευτής μπορεί να διαλέγεται με δύο ή περισσότερους αποδείκτες; Αν οι αποδείκτες επικοινωνούν μεταξύ τους, τότε παραμένουμε στην κλάση IP (πρακτικά, ένας αποδείκτης, ως παντοδύναμος αλγόριθμος, μπορεί να εξομοιώνει οσουσδήποτε άλλους). Αν όμως, οι αποδείκτες δεν έχουν επικοινωνία μεταξύ τους, τότε προκύπτει η ισχυρότερη κλάση MIP (Multi IP). Μάλιστα ισχύει: MIP $=$ NEXP.

Διαλογική αλληλεπίδραση (interactivity) I

Κλάσεις Arthur-Merlin

Στην κλάση IP ο επαληθευτής κρατά <<κρυφά>> τα τυχαία bits που χρησιμοποιεί. Μάλιστα, στην απόδειξη ότι το πρόβλημα μη ισομορφισμού γράφων είναι στην IP, αυτό αποτελεί βασικό συστατικό της απόδειξης. Φαίνεται ότι αν ο επαληθευτής είναι υποχρεωμένος να αποκαλύπτει τα bits του, προκύπτει μία μικρότερη κλάση γλωσσών από την IP. Σε αυτήν την κλάση γλωσσών ο αποδείκτης ονομάζεται Merlin και ο επαληθευτής Arthur (αυτή η περιγραφή οφείλεται στον Babai). Μάλιστα, μπορούμε να θεωρήσουμε ότι τα μηνύματα του Arthur είναι ακόμα πιο περιορισμένα: απλώς στέλνει τα τυχαία bits στον Merlin. Ανάλογα με τις απαντήσεις του Merlin, ο Arthur αποφασίζει αν θα αποδεχθεί.

Διαλογική αλληλεπίδραση (interactivity) II

Κλάσεις Arthur-Merlin

Λέμε ότι οι Arthur και Merlin παίζουν ένα παιχνίδι k κινήσεων μεταξύ τους (κάθε κίνηση αντιστοιχεί σε ένα μήνυμα): αν ο Arthur κινείται πρώτος το παιχνίδι συμβολίζεται με $\mathrm{AM}(k)$, ενώ αν κινείται πρώτος ο Merlin με $\mathrm{MA}(k)$. Για παράδειγμα, $\mathrm{AM}(1) = \mathrm{A}$, $\mathrm{AM}(2) = \mathrm{AM}$, $\mathrm{AM}(3) = \mathrm{AMA}$, $\mathrm{MA}(1) = \mathrm{M}$, $\mathrm{MA}(2) = \mathrm{MA}$, $\mathrm{MA}(3) = \mathrm{MAM}$. Μία άλλη διαφορά σε σχέση με την κλάση IP είναι ότι χρειάζεται να φράξουμε τις πιθανότητες μακριά από το $1/2$ (πάλι δεν έχει μεγάλη σημασία η ακριβής τιμή). Τυπικά, για την κλάση $\mathrm{AM}(k)$, έχουμε:

Ορισμός

$L \in \mathrm{AM}(k)$ ανν υπάρχει παιχνίδι k κινήσεων όπου παίζει πρώτος ο Arthur και στο οποίο αν:

- $x \in L \implies$ ο Arthur πείθεται με πιθανότητα μεγαλύτερη από $2/3$ ότι $x \in L$·
- $x \notin L \implies$ ο Arthur πείθεται με πιθανότητα μικρότερη από $1/3$ ότι $x \in L$.

Με την βοήθεια των γενικευμένων ποσοδεικτών οι κλάσεις μπορούν να γραφτούν ως εξής (Zachos):

$$\mathrm{AM} = \mathrm{AM}(2) = (\exists^{+}\exists, \exists^{+}\forall), \quad \mathrm{MA} = \mathrm{MA}(2) = (\exists\exists^{+}, \forall\exists^{+}),$$

Διαλογική αλληλεπίδραση (interactivity) III

Κλάσεις Arthur-Merlin

και για άρτιο k, αν $\mathrm{AM}(k) = (\mathbf{Q}_1, \mathbf{Q}_2)$, όπου $\mathbf{Q}_1, \mathbf{Q}_2$ ακολουθίες ποσοδεικτών:

$$\mathrm{AM}(k+1) = (\mathbf{Q}_1 \exists^+, \mathbf{Q}_2 \exists^+), \quad \mathrm{AM}(k+2) = (\mathbf{Q}_1 \exists^+ \exists, \mathbf{Q}_2 \exists^+ \forall).$$

Η παραπάνω περιγραφή, μπορεί να απλοποιηθεί ως εξής (Zachos):

$$\mathrm{AM} = \mathrm{AM}(2) = (\forall \exists, \exists^+ \forall), \quad \mathrm{MA} = \mathrm{MA}(2) = (\exists \forall, \forall \exists^+),$$

και για άρτιο k, αν $\mathrm{AM}(k) = (\mathbf{Q}_1, \mathbf{Q}_2)$, όπου $\mathbf{Q}_1, \mathbf{Q}_2$ ακολουθίες ποσοδεικτών:

$$\mathrm{AM}(k+1) = (\mathbf{Q}_1 \forall, \mathbf{Q}_2 \exists^+), \quad \mathrm{AM}(k+2) = (\mathbf{Q}_1 \forall \exists, \mathbf{Q}_2 \exists^+ \forall).$$

Χρησιμοποιώντας ιδιότητες των ποσοδεικτών προκύπτουν τα παρακάτω αποτελέσματα:

Πρόταση

$\mathrm{MA} \subseteq \mathrm{AM}$.

Διαλογική αλληλεπίδραση (interactivity) IV

Κλάσεις Arthur-Merlin

Πρόταση

Η ιεραρχία των παιχνιδιών Arthur-Merlin καταρρέει, δηλαδή:

$$\mathrm{AM} = \mathrm{AM}(k) = \mathrm{MA}(k+1), \quad \text{για κάθε } k \geqslant 2.$$

Αν και όπως είπαμε, η κλάση Arthur-Merlin με πολυωνυμικό πλήθος μηνυμάτων αλληλεπίδρασης φαίνεται ασθενέστερη (λόγω δημοσιοποίησης των τυχαίων bits) σε σχέση με την IP, εν τούτοις οι Goldwasser, Sipser απέδειξαν ότι είναι ισοδύναμες.

Διαλογική αλληλεπίδραση (interactivity) I

Probabilistic Checable Proofs --- PCP

Αν αντικαταστήσουμε στις διαλογικές αποδείξεις, τον αποδείκτη με μία απλή απόδειξη, έχουμε την κλάση PCP. Ας πούμε ότι στην PCP, ο αποδείκτης δεν έχει καμμία άλλη επικοινωνία, εκτός από το να γράψει στην αρχή της αλληλεπίδρασης με τον επαληθευτή V μίαν απόδειξη και να την στείλει στον V. Πρέπει να σημειώσουμε ότι οι αποδείξεις αυτές ελέγχονται πιθανοτικά από τον V. Τυπικά:

Ορισμός

$L \in$ PCP:

- $x \in L \implies$ υπάρχει απόδειξη Π τέτοια ώστε ο επαληθευτής (verifier) V πάντοτε αποδέχεται (δηλαδή έχουμε πιθανότητα αποδοχής ίση με 1)·
- $x \notin L \implies$ για κάθε <<απόδειξη>> Π, ο επαληθευτής V δεν αποδέχεται με συντριπτική πιθανότητα.

Αυτή η κλάση φαίνεται πολύ ισχυρότερη από την IP γιατί πλέον ο επαληθευτής έχει να <<αντιμετωπίσει>> ένα στατικό αντικείμενο (την απόδειξη) και όχι ένα προσαρμοζόμενο στις ερωτήσεις του (τον αποδείκτη). Και πράγματι αποδεικνύεται ότι PCP $=$ MIP$(=$ NEXP$)$. Για τον λόγο αυτό, θα

Διαλογική αλληλεπίδραση (interactivity) II

Probabilistic Checable Proofs --- PCP

θεωρήσουμε περιορισμούς της κλάσης PCP. Θα θεωρήσουμε δύο είδη αγαθών που δεν μπορεί να χρησιμοποιεί αφειδώς ο επαληθευτής:

- τυχαιότητα (με την μορφή τυχαίων bits)·
- bits της απόδειξης που εξετάζονται (ερωτήσεις, ή αλλιώς queries, στην απόδειξη).

Ορισμός

Η κλάση $\mathrm{PCP}(r(n), q(n))$ αποτελείται από τις γλώσσες $L \in \mathrm{PCP}$ για τις οποίες ο πιθανοτικός πολυωνυμικού χρόνου επαληθευτής V χρησιμοποιεί $O(r(n))$ τυχαία bits και ελέγχει $O(q(n))$ bits στην απόδειξη.

Για παράδειγμα, ήδη γνωστές κλάσεις πολυπλοκότητας μπορούν να οριστούν με την βοήθεια των παραπάνω: $\mathrm{PCP} = \mathrm{PCP}(\mathrm{poly}(n), \mathrm{poly}(n))$, $\mathrm{P} = \mathrm{PCP}(0, 0)$, $\mathrm{NP} = \mathrm{PCP}(0, \mathrm{poly}(n))$, $\mathrm{coRP} = \mathrm{PCP}(\mathrm{poly}(n), 0)$.

Ένα πολύ σημαντικό αποτέλεσμα (Arora, Lund, Motwani, Sudan, Szegedy) είναι το εξής:

Διαλογική αλληλεπίδραση (interactivity) III

Probabilistic Checable Proofs --- PCP

Θεώρημα (PCP)

$\mathrm{NP} = \mathrm{PCP}(\log n, 1)$.

Μία εφαρμογή του θεωρήματος PCP είναι σε αποδείξεις μη προσεγγισιμότητας.

Το βασικό εργαλείο στην απόδειξη του προηγουμένου θεωρήματος είναι μία μέθοδος (PCP encoding) που διαχέει ένα πιθανό λάθος μίας απόδειξης σε όλα τα κομμάτια της απόδειξης, έτσι ώστε ο επαληθευτής να έχει συντριπτική πιθανότητα να διαγνώσει το λάθος. Η μέθοδος αυτή βασίζεται σε τεχνικές κωδίκων διόρθωσης λαθών (error correcting codes).

Μετρητικές Κλάσεις I

Μετρητικές κλάσεις ορίζονται με βάση το πλήθος των λύσεων που έχει κάποιο πρόβλημα. Πρόκειται δηλαδή για κλάσεις συναρτήσεων (όπως η FP). Ενδιαφέρον έχουν οι παρακάτω δύο κλάσεις:

Ορισμός

#P είναι η κλάση των συναρτήσεων f για τις οποίες υπάρχει μη ντετερμινιστική μηχανή Turing πολυωνυμικού χρόνου, το υπολογιστικό δέντρο της οποίας έχει ακριβώς $f(x)$ υπολογιστικά μονοπάτια που αποδέχονται (για είσοδο x).

Ορισμός

#L είναι η κλάση των συναρτήσεων f για τις οποίες υπάρχει μη ντετερμινιστική μηχανή Turing λογαριθμικού χώρου, το υπολογιστικό δέντρο της οποίας έχει ακριβώς $f(x)$ υπολογιστικά μονοπάτια που αποδέχονται (για είσοδο x).

Σε μετρητικές κλάσεις χρήσιμες είναι αναγωγές που διατηρούν το πλήθος των λύσεων.

Μετρητικές Κλάσεις II

Ένα χαρακτηριστικό παράδειγμα πλήρους προβλήματος για την #P είναι το πρόβλημα #SAT: <<Δίνεται τύπος σε συζευκτική κανονική μορφή. Πόσες διαφορετικές αναθέσεις υπάρχουν που ικανοποιούν τον τύπο;>>. Είναι προφανές ότι $\varphi \in \text{SAT}$ ανν $\#\text{SAT}(\varphi) \neq 0$.

Ο *Valiant* έδειξε ότι υπάρχουν προβλήματα απόφασης στο P (π.χ. ύπαρξη τέλειου ταιριάσματος σε γραφήματα) των οποίων το αντίστοιχο μετρητικό πρόβλημα (π.χ. *#PERFECT MATCHINGS*) είναι #P-πλήρες.

Μερικά αποτελέσματα για αυτές τις κλάσεις:

$$\text{FP} \subseteq \#\text{P} \subseteq \text{FPSPACE}, \quad \text{P}^{\text{PP}} = \text{P}^{\#\text{P}}, \quad \text{FL} \subseteq \#\text{L} \subseteq \text{FNC}^2.$$

Θεώρημα (Toda)

$\text{PH} \subseteq \text{P}^{\#\text{P}}$.

Η απόδειξη αποτελείται από δύο βασικούς εγκλεισμούς, όπως φαίνεται στα παρακάτω λήμματα:

Μετρητικές Κλάσεις III

Lemma

$\mathbf{PH} \subseteq \mathbf{BPP}^{\oplus\mathbf{P}}$.

Απόδειξη:

1. $\oplus\mathbf{P}^{\oplus\mathbf{P}} = \oplus\mathbf{P}$ (Papadimitriou-Zachos)
2. $\mathbf{NP} \subseteq \mathbf{BPP} \Rightarrow \mathbf{PH} \subseteq \mathbf{BPP}$ (Zachos)
3. $\mathbf{NP} \subseteq \mathbf{RP}^{\oplus\mathbf{P}}$ (Valiant-Vazirani) $\subseteq \mathbf{BPP}^{\oplus\mathbf{P}}$
4. $\mathbf{NP}^{\oplus\mathbf{P}} \subseteq \mathbf{BPP}^{\oplus\mathbf{P}^{\oplus\mathbf{P}}}$ (3. με μαντείο $\oplus\mathbf{P}$) $\overset{1.}{\Rightarrow} \mathbf{NP}^{\oplus\mathbf{P}} \subseteq \mathbf{BPP}^{\oplus\mathbf{P}}$
5. $\mathbf{NP}^{\oplus\mathbf{P}} \subseteq \mathbf{BPP}^{\oplus\mathbf{P}} \Rightarrow \mathbf{PH}^{\oplus\mathbf{P}} \subseteq \mathbf{BPP}^{\oplus\mathbf{P}}$ (2. με μαντείο $\oplus\mathbf{P}$)
6. $\mathbf{PH}^{\oplus\mathbf{P}} = \mathbf{PH} \subseteq \mathbf{BPP}^{\oplus\mathbf{P}}$

□

Μετρητικές Κλάσεις IV

Lemma

$\mathbf{BPP}^{\oplus\mathbf{P}} \subseteq \mathbf{P}^{\#\mathbf{P}}$

Χωρίς απόδειξη.

□

Προσεγγιστικοί Αλγόριθμοι Ι

Γνωρίζουμε ότι τα *NP*-δύσκολα προβλήματα δεν μπορούμε να τα λύσουμε:

1. Ακριβώς
2. Για όλα τα στιγμιότυπα
3. Σε πολυωνυμικό χρόνο

Προσεγγιστικοί Αλγόριθμοι ΙΙ

- Αν αγνοήσουμε την συνθήκη (1), τότε έχουμε **Προσεγγιστικούς Αλγορίθμους**.
- Αν αγνοήσουμε την συνθήκη (2), τότε **μπορούμε να βρούμε μεγάλες υποκλάσεις στιγμιοτύπων του προβλήματος, στις οποίες το πρόβλημα να λύνεται σε πολυωνυμικό χρόνο**, αλλά επίσης είναι δυνατόν να αποφασίσουμε ντετερμινιστικά σε πολυωνυμικό χρόνο εάν η είσοδος ανήκει στην εν λόγω υποκλάση.
  - Ψευδοπολυωνυμικοί, Ισχυρά Πολυωνυμικοί
  - Παραμετροποίηση (πχ VERTEXCOVER(n, k))
    Παραμετρική Πολυπλοκότητα ($2^k n^c$, n^k κλπ)
- Αν αγνοήσουμε την συνθήκη (3), τότε κατατάσσουμε υπερπολυωνυμικές λύσεις με μεγαλύτερη ευκρίνεια.
  $1.003^n \leqslant 1.5^n \leqslant 2^n \leqslant 5^n \leqslant n! \leqslant n^n$
  $n^{\log\log n} \leqslant n^{\log n} \leqslant n^{\log^{13} n} \leqslant n^n$.

Προσεγγιστικοί Αλγόριθμοι ΙΙΙ

Ένα πρόβλημα βελτιστοποίησης είναι: (I, S, v, goal).

- I: τα στιγμιότυπα του προβλήματος·
- S: μία συνάρτηση που αντιστοιχίζει σε κάθε στιγμιότυπο τις εφικτές λύσεις·
- v: η αντικειμενική συνάρτηση· αντιστοιχίζει σε κάθε εφικτή λύση, έναν θετικό ακέραιο·
- goal: min ή max, για πρόβλημα ελαχιστοποίησης ή μεγιστοποίησης της αντικειμενικής συνάρτησης, αντίστοιχα.

Η τιμή της αντικειμενικής συνάρτησης για την βέλτιστη λύση με είσοδο x συμβολίζεται με $\text{OPT}(x)$ και είναι ίση με $\text{goal}\{v(y) \mid y \in S(x)\}$.

Προσεγγιστικοί Αλγόριθμοι IV

Επίσης, ορίζουμε για κάθε πρόβλημα βελτιστοποίησης το *αντίστοιχο* (underlying) πρόβλημα απόφασης ως εξής:

Δίδεται επιπλέον της εισόδου x ένα φράγμα k.
Ερώτηση: είναι $\mathrm{OPT}(x) \geqslant k$;
(για πρόβλημα μεγιστοποίησης -- ανάλογα για πρόβλημα ελαχιστοποίησης.)

Παράδειγμα

Στο πρόβλημα MAX-CLIQUE το στιγμιότυπο είναι ένας γράφος x, οι εφικτές λύσεις είναι όλοι οι πλήρεις υπογράφοι του x (κλίκες), η αντικειμενική συνάρτηση είναι το πλήθος των κόμβων της κλίκας και goal = max. Το αντίστοιχο πρόβλημα απόφασης είναι το γνωστό CLIQUE.

Προσεγγιστικοί Αλγόριθμοι V

Ορίζουμε τις παρακάτω βασικές κλάσεις πολυπλοκότητας για προβλήματα βελτιστοποίησης:

Ορισμός

NPO: η κλάση των προβλημάτων βελτιστοποίησης, για τα οποία το αντίστοιχο πρόβλημα απόφασης είναι στο NP (με την προϋπόθεση ότι υπάρχουν εφικτές λύσεις για κάθε στιγμιότυπο).

Ορισμός

PO: η κλάση των προβλημάτων βελτιστοποίησης, για τα οποία το αντίστοιχο πρόβλημα απόφασης είναι στο P.

Προσεγγιστικοί Αλγόριθμοι VI

Πολλά προβλήματα βελτιστοποίησης είναι NP-δύσκολα. Για αυτό αναζητούμε προσεγγιστικούς πολυωνυμικούς αλγόριθμους που επιλύουν τέτοια προβλήματα.

Ορισμός

Ένας πολυωνυμικός αλγόριθμος M είναι ρ-προσεγγιστικός για ένα πρόβλημα μεγιστοποίησης αν για κάθε $x \in I$ επιστρέφει μια λύση $M(x) \in S(x)$ τέτοια ώστε:

$$\frac{v(M(x))}{\mathrm{OPT}(x)} \leqslant \rho.$$

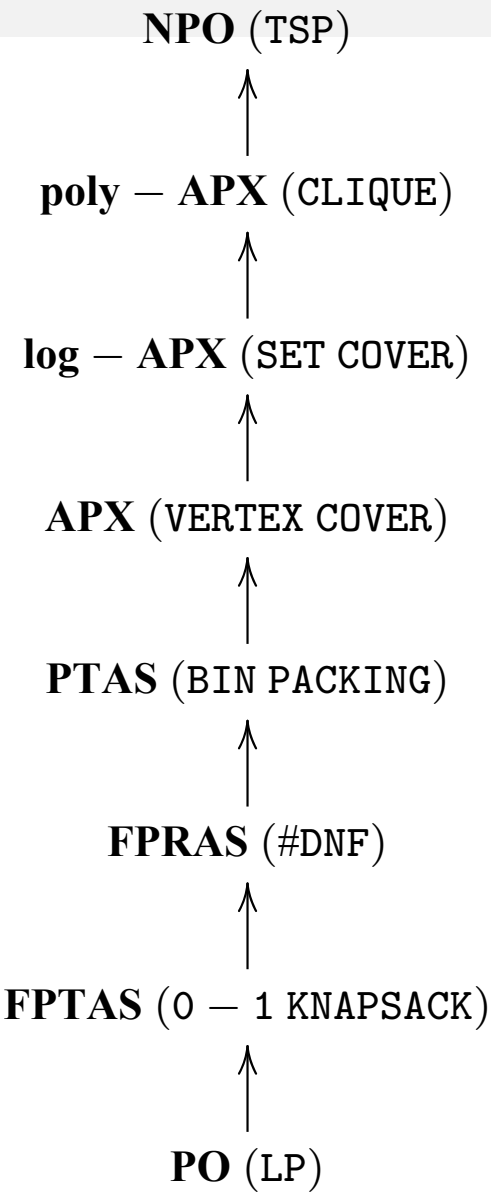
Αντίστοιχα ορίζεται ρ-προσεγγιστικός αλγόριθμος για πρόβλημα ελαχιστοποίησης.

Προσεγγιστικοί Αλγόριθμοι VII

Οι πιο γνωστές υποκλάσεις της NPO, εκτός της PO, είναι οι εξής:

- poly-APX: περιέχει προβλήματα για τα οποία υπάρχει $p(n)$-προσεγγιστικός αλγόριθμος για κάποιο πολυώνυμο p (όπου n είναι το μήκος της εισόδου: $n = |x|$).
- log-APX: περιέχει προβλήματα για τα οποία υπάρχει $\log n$-προσεγγιστικός αλγόριθμος (όπου n είναι το μήκος της εισόδου: $n = |x|$).
- APX: περιέχει προβλήματα για τα οποία υπάρχει ρ-προσεγγιστικός αλγόριθμος για κάποια σταθερά $\rho > 0$.
- PTAS: περιέχει προβλήματα για τα οποία υπάρχει *πολυωνυμικού χρόνου προσεγγιστικό σχήμα*, δηλαδή $(1+\varepsilon)$-προσεγγιστικός αλγόριθμος *για κάθε* σταθερά $\varepsilon > 0$.
- FPTAS: περιέχει προβλήματα για τα οποία υπάρχει *πλήρως πολυωνυμικού χρόνου προσεγγιστικό σχήμα*, δηλαδή $(1+\varepsilon)$-προσεγγιστικός αλγόριθμος για κάθε σταθερά $\varepsilon > 0$, που επιπλέον ο χρόνος που χρειάζεται είναι πολυωνυμικός και ως προς το $1/\varepsilon$.

Προσεγγιστικοί Αλγόριθμοι VIII



Σχήμα: Κλάσεις προβλημάτων βελτιστοποίησης