Κλάσεις Πολυπλοκότητας I

- **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 *περιοριμοί*, 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 II

- **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): O 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 $s(n)$.

Βασικοί Ορισμοί II

Με βάση τα παραπάνω, ορίζουμε:

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

Βασικοί Ορισμοί III

Παρατήρηση

Μία συνάρτηση f ονομάζεται συνάρτηση πολυπλοκότητας (constructible) αν 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).

Αν η f είναι μία συνάρτηση πολυπλοκότητας τότε ισχύουν:

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

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

Βασικοί Ορισμοί IV

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

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

- $\mathrm{NSPACE}(f(n)) \subseteq \mathrm{DTIME}(k^{\log n + f(n)})$

Αν $f(n) > \log n$ τότε:

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

Βασικοί Ορισμοί V

Το παρακάτω θεώρημα οφείλεται στον Savitch (1970):

Θεώρημα

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

Άμεσα από το θεώρημα του Savitch προκύπτει ότι $\mathrm{PSPACE} = \mathrm{NPSPACE}$.

Από τις παραπάνω σχέσεις προκύπτει η εξής ιεραρχία:

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

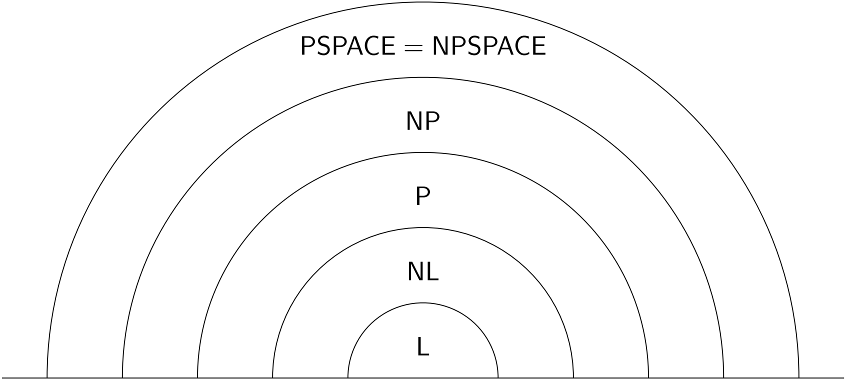
Γνωρίζουμε ότι $\mathrm{L} \neq \mathrm{PSPACE}$ και $\mathrm{NL} \neq \mathrm{PSPACE}$ (αυτό προκύπτει από το θεώρημα ιεραρχίας για χωρικές κλάσεις πολυπλοκότητας, που αναφέρεται παρακάτω).

Ανοιχτά παραμένουν τα προβλήματα:

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

Βασικοί Ορισμοί VI

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



Σχήμα: Κλάσεις πολυπλοκότητας

Βασικοί Ορισμοί VII

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

Οι 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* $\mathrm{TIME}(t(n)) = \mathrm{TIME}(2^{t(n)})$.

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):

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

Παράδειγμα: η κλάση coNP αποτελείται από τις γλώσσες που είναι συμπληρώματα γλωσσών στοNP. Ένα πρόβλημα που ανήκει στην κλάση coNP είναι το $\overline{\mathrm{SAT}}$ 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 $\mathrm{coNP} \neq \mathrm{NP}$. Μάλιστα, το τελευταίο συνδέεται και με το πρόβλημα αν $\mathrm{P} \neq \mathrm{NP}$, αφού προφανώς αν $\mathrm{coNP} \neq \mathrm{NP}$, τότε $\mathrm{P} \neq \mathrm{NP}$.

Συμπληρωματικές κλάσεις πολυπλοκότητας III

Θεώρημα (Immerman-Szelepcsényi)

Η κλάση $\mathrm{NSPACE}(s(n))$ *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)

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

Αναγωγές II

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:

Ορισμός (Log-space Αναγωγή)

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

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

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

Αναγωγές III

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).

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

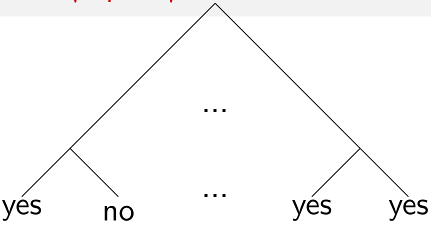
Αναφέρουμε επιγραμματικά μερικές παραμέτρους με τις οποίες ορίζονται κλάσεις πολυπλοκότητας:

- **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 οριστεί 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 οριστεί 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 ορίζεται 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, εισαγάγουμε τον παρακάτω συμβολισμό:

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

Προσεγγιστικοί Αλγόριθμοι Ι

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 $\mathrm{OPT}(x)$ και είναι ίση με $\mathrm{goal}\{v(y) \mid y \in S(x)\}$.

Προσεγγιστικοί Αλγόριθμοι II

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 III

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 IV

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:

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

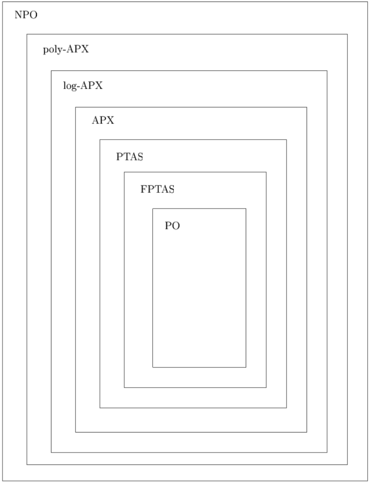
Αντίστοιχα ορίζεται ρ-προσεγγιστικός αλγόριθμος για πρόβλημα ελαχιστοποίησης.

Προσεγγιστικοί Αλγόριθμοι V

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 VI



Σχήμα: Κλάσεις προβλημάτων βελτιστοποίησης