Σύντομη Εισαγωγή στη Θεωρία Υπολογιστικής Πολυπλοκότητας

Στάθης Ζάχος, Άρης Παγουρτζής, and Δημήτρης Φωτάκης

Σχολή Ηλεκτρολόγων Μηχανικών και Μηχανικών Υπολογιστών
Εθνικό Μετσόβιο Πολυτεχνείο
Email: zachos@cs.ntua.gr, pagour@cs.ntua.gr, fotakis@cs.ntua.gr

1 Εισαγωγή – Ιστορική Αναδρομή

Η θεωρητική θεμελίωση του υπολογισμού έχει τις ρίζες της στην μαθηματική λογική. Η Συλλογιστική του Αριστοτέλη αποτέλεσε την πρώτη προσπάθεια θεμελίωσης της λογικής και των μαθηματικών. Πολλούς αιώνες αργότερα, ο Leibni(t)z πρότεινε το εξής πρόγραμμα:

1. Να δημιουργηθεί μια τυπική γλώσσα (formal language), με την οποία να μπορούμε να περιγράψουμε όλες τις μαθηματικές έννοιες και προτάσεις.
2. Να δημιουργηθεί μια μαθηματική θεωρία (δηλ. ένα σύνολο από αξιώματα και συμπερασματικούς κανόνες συνεπαγωγής), με την οποία να μπορούμε να αποδεικνύουμε όλες τις ορθές μαθηματικές προτάσεις.
3. Να αποδειχθεί ότι αυτή η θεωρία είναι συνεπής, (δηλ. ότι η πρόταση "A και όχι A" ($A \wedge \neg A$) δεν είναι δυνατόν να αποδειχθεί σ' αυτή τη θεωρία).

Η πραγμάτωση αυτού του προγράμματος άρχισε πολύ αργότερα, προς το τέλος του 19ου αιώνα. Πολλοί επιστήμονες ασχολήθηκαν με τον ορισμό της ενιαίας γλώσσας της μαθηματικής (ή συμβολικής) λογικής (Boole, Frege, κ.α). Άλλοι ασχολήθηκαν με τον ορισμό της ενιαίας θεωρίας των συνόλων (Cantor, κ.α.) και άλλοι με την παραγωγή όλων των αληθών μαθηματικών προτάσεων με χρήση της Συνολοθεωρίας (Frege, Russel, Whitehead, κ.α.).

Στην αρχή του 20ου αιώνα, ο Hilbert έθεσε ως στόχο την εύρεση αλγόριθμου που να αποκρίνεται για την ορθότητα κάθε μαθηματικής πρότασης. Το 1931, ο Godel απέδειξε το *Θεώρημα Μη Πληρότητας*, συνέπειες του οποίου είναι ότι:

- Είναι αδύνατον να αποδειχθεί η συνέπεια της Συνολοθεωρίας.
- Οποιαδήποτε αξιωματική θεωρία των Μαθηματικών, που περιλαμβάνει τουλάχιστον την Αριθμοθεωρία, θα περιλαμβάνει και προτάσεις, που η αλήθειά τους δεν είναι αποδείξιμη.

Το Θεώρημα Μη Πληρότητας του Godel ήταν η αιτία μιας σημαντικής κρίσης στα κλασικά μαθηματικά, μα συγχρόνως και η απαρχή των μοντέρνων δυναμικών μαθηματικών. Το κεντρικό ερώτημα δεν είναι πια απλά αν μια πρόταση είναι αληθής ή ψευδής, αλλά αν είναι "υπολογιστή (computable) ή όχι". Αυτό ακριβώς είναι και το αντικείμενο της *Θεωρίας της Υπολογιστότητας* (Computability Theory ή Theory of Computation). Το επόμενο βήμα, δηλαδή "δεδομένης μιας υπολογιστής συνάρτησης f [1], ποια είναι η ποσότητα υπολογιστικών πόρων που χρειάζονται για τον υπολογισμό της;" αποτελεί το βασικό ερώτημα της *Θεωρίας της Υπολογιστικής Πολυπλοκότητας* (Computational Complexity Theory).

[1] Συχνά χρησιμοποιείται και ο όρος υπολογίσιμος αντί του όρου υπολογιστός.

Οργάνωση. Το κεφάλαιο αυτό αποτελείται από τρία μέρη. Το πρώτο μέρος αφορά στις βασικές έννοιες της Θεωρίας Υπολογιστότητας (βλ. π.χ. [5,12,16,22,25]). Αφού ορίσουμε την έννοια του (υπολογιστικού) προβλήματος απόφασης και αντιστοιχίσουμε κάθε τέτοιο πρόβλημα σε μία τυπική γλώσσα (Ενότητα 2), θα ορίσουμε το υπολογιστικό μοντέλο των μηχανών Turing (Ενότητα 3) και τη μη ντετερμινιστική εκδοχή τους (Ενότητα 5). Με βάση το υπολογιστικό μοντέλο της μηχανής Turing, θα ορίσουμε την έννοια του υπολογιστού προβλήματος (Ενότητα 4), και θα δώσουμε ένα παράδειγμα μη υπολογιστού προβλήματος (Ενότητα 4.1). Το δεύτερο μέρος αφορά στις βασικές έννοιες της Θεωρίας Υπολογιστικής Πολυπλοκότητας (βλ. π.χ. [2,5,8,17,25]). Ειδικότερα, θα μιλήσουμε για ντετερμινιστική χρονική πολυπλοκότητα, με έμφαση στην κλάση **P** (Ενότητα 7.1), και μη ντετερμινιστική χρονική πολυπλοκότητα, με έμφαση στην κλάση **NP** (Ενότητα 7.2), και για τις θεμελιώδεις έννοιες της αναγωγής και της πληρότητας (Ενότητα 8), με έμφαση στην πολυωνυμική αναγωγή και στην πληρότητα για την κλάση **NP** (Ενότητα 9). Το τρίτο μέρος αφορά στον ορισμό άλλων σημαντικών κλάσεων υπολογιστικής πολυπλοκότητας. Θα παρουσιάσουμε τις βασικές κλάσεις χωρικής πολυπλοκότητας (Ενότητα 10.1), δύο κλάσεις πολυπλοκότητας συναρτήσεων (Ενότητα 10.2), την έννοια των συμπληρωματικών κλάσεων πολυπλοκότητας (Ενότητα 10.3), τις βασικές κλάσεις πολυπλοκότητας πιθανοτικών αλγορίθμων (Ενότητα 10.4), και την έννοια των διαλογικών συστημάτων απόδειξης και την κλάση **IP** (Ενότητα 10.6). Ακόμη θα εισαγάγουμε την ιδιαίτερα χρήσιμη (διασθητικά και πρακτικά) έννοια των πλειοψηφικών ποσοδεικτών, με τη βοήθεια των οποίων θα κωδικοποιήσουμε τους ορισμούς των κλάσεων πολυπλοκότητας πιθανοτικών αλγορίθμων (Ενότητα 10.5).

2 Υπολογιστικά Προβλήματα και Τυπικές Γλώσσες

Για τη Θεωρία Υπολογιστότητας και τη Θεωρία Υπολογιστικής Πολυπλοκότητας, τα (υπολογιστικά) προβλήματα είναι (μαθηματικές) οντότητες που καθ' εαυτές χρήζουν μελέτης ως προς τα υπολογιστικά χαρακτηριστικά τους.

Ένα υπολογιστικό πρόβλημα Π συσχετίζει κάθε έγκυρο στιγμιότυπο με τις λύσεις του. Για παράδειγμα, ας θεωρήσουμε το πρόβλημα Συντομότερου Μονοπατιού (Shortest Path Problem) μεταξύ ενός δεδομένου ζεύγους κορυφών σε ένα γράφημα. Ένα στιγμιότυπο αποτελείται από ένα κατευθυνόμενο γράφημα με μήκη (ή βάρη) στις ακμές $G(V, E, w)$, και δύο κορυφές $s, t \in V$. Κάθε τέτοια τριάδα $(G(V, E, w), s, t)$ αποτελεί ένα έγκυρο στιγμιότυπο του προβλήματος. Κάθε συντομότερο μονοπάτι από την s στην t αποτελεί μία λύση του προβλήματος. Ως λύση θεωρείται και το κενό μονοπάτι, όταν η t δεν είναι προσπελάσιμη από την s. Επειδή ένα συντομότερο $s-t$ μονοπάτι δεν είναι αναγκαστικά μοναδικό, ένα στιγμιότυπο μπορεί να έχει περισσότερες από μία λύσεις.

Προβλήματα Βελτιστοποίησης. Το πρόβλημα Συντομότερου Μονοπατιού μεταξύ ενός ζεύγους κορυφών αποτελεί ένα τυπικό παράδειγμα *προβλήματος βελτιστοποίησης* (optimization problem). Σε ένα πρόβλημα βελτιστοποίησης ζητάμε λύση που ελαχιστοποιεί (ή μεγιστοποιεί) κάποια αντικειμενική συνάρτηση. Ειδικότερα, για κάθε στιγμιότυπο x, υπάρχει ένα σύνολο *εφικτών* (feasible) λύσεων $F(x)$. Σε κάθε λύση $s \in F(x)$, αντιστοιχεί, μέσω μιας αντικειμενικής συνάρτησης c, ένας θετικός ακέραιος $c(s)$. Το ζητούμενο είναι ο υπολογισμός μιας *βέλτιστης λύσης* (optimal solution) $s \in F(x)$, για την οποία το $c(s)$ είναι ελάχιστο, αν πρόκειται για *πρόβλημα ελαχιστοποίησης*, ή μέγιστο, αν πρόκειται για *πρόβλημα μεγιστοποίησης*.

Στο πρόβλημα Συντομότερου Μονοπατιού μεταξύ ενός ζεύγους κορυφών, για κάθε στιγμιότυπο $(G(V, E, w), s, t)$, το σύνολο των $s-t$ μονοπατιών αποτελεί το σύνολο των εφικτών λύσεων.

Η αντικειμενική συνάρτηση αντιστοιχίζει σε κάθε $s-t$ μονοπάτι το συνολικό μήκος του. Βέλτιστη λύση είναι κάθε $s-t$ μονοπάτι με ελάχιστο συνολικό μήκος.

Ένα άλλο τυπικό παράδειγμα προβλήματος βελτιστοποίησης είναι το *πρόβλημα του Πλανόδιου Πωλητή* (Traveling Salesperson Problem, TSP). Σε αυτό, δίνεται ένα πεπερασμένο σύνολο πόλεων $C = \{c_1, \ldots, c_n\}$, και απόστασεις (ή μήκη) $d(c_i, c_j) \in \mathbb{N}$, για κάθε ζεύγος πόλεων c_i, c_j. Το ζητούμενο είναι μια περιοδεία, που περνά μια φορά από κάθε πόλη και επιστρέφει στην αρχική, ελάχιστου συνολικού μήκους (κόστους). Θέλουμε δηλαδή να υπολογίσουμε μια μετάθεση π του C που ελαχιστοποιεί το άθροισμα:

$$\sum_{i=1}^{n-1} d(c_{\pi(i)}, c_{\pi(i+1)}) + d(c_{\pi(n)}, c_{\pi(1)})$$

Προβλήματα Απόφασης. Ένας άλλος τρόπος να διατυπωθεί το πρόβλημα Συντομότερου Μονοπατιού είναι να θεωρήσουμε ένα άνω φράγμα $B \geq 0$, και να εξετάσουμε αν υπάρχει $s-t$ μονοπάτι με μήκος μικρότερο ή ίσο του B. Σε αυτή τη διατύπωση, ένα στιγμιότυπο είναι μια τετράδα $(G(V, E, w), s, t, B)$, και οι δυνατές λύσεις του προβλήματος είναι δύο: NAI και OXI. Κάθε πρόβλημα με σύνολο λύσεων $\{\text{NAI}, \text{OXI}\}$ ονομάζεται *πρόβλημα απόφασης* (decision problem).

Με παρόμοιο τρόπο μπορούμε να διατυπώσουμε το πρόβλημα του Πλανόδιου Πωλητή, αλλά και κάθε άλλο πρόβλημα βελτιστοποίησης, ως πρόβλημα απόφασης. Ειδικότερα, θεωρούμε ένα φράγμα B, και εξετάζουμε αν "υπάρχει εφικτή λύση με κόστος μικρότερο ή ίσο του B;" προκειμένου για πρόβλημα ελαχιστοποίησης, και αν "υπάρχει εφικτή λύση με κέρδος μεγαλύτερο ή ίσο του B;" προκειμένου για πρόβλημα μεγιστοποίησης. Αν ένα πρόβλημα απόφασης μπορεί να λυθεί αποδοτικά (π.χ. σε πολυωνυμικό χρόνο), τότε και το αντίστοιχο πρόβλημα βελτιστοποίησης μπορεί να λυθεί αποδοτικά (π.χ. σε πολυωνυμικό χρόνο), και αντίστροφα.

Δύο άλλα τυπικά παραδείγματα προβλημάτων απόφασης αφορούν στην ύπαρξη *κύκλου Hamilton* (Hamilton Cycle Problem) σε ένα γράφημα, και στην *Ικανοποιησιμότητα* μιας λογικής πρότασης (Satisfiability Problem, SAT) σε *Συζευκτική Κανονική Μορφή* (Conjunctive Normal Form, CNF).

Προβλήματα Απόφασης και Τυπικές Γλώσσες. Στη Θεωρία Υπολογιστότητας και τη Θεωρία Υπολογιστικής Πολυπλοκότητας, εστιάζουμε στη μελέτη προβλημάτων απόφασης. Ένας από τους λόγους είναι ότι τα προβλήματα αυτά έχουν άμεση σχέση με τις τυπικές γλώσσες.

Για να λυθεί κάποιο πρόβλημα, πρέπει τα στιγμιότυπα του προβλήματος να αναπαρασταθούν με τρόπο αντιληπτό από τον υπολογιστή. Για το σκοπό αυτό, χρησιμοποιούμε μία *κωδικοποίηση* (encoding) e που αντιστοιχίζει κάθε στιγμιότυπο x του προβλήματος σε μια μοναδική συμβολοσειρά $e(x)$ ενός αλφάβητου Σ με τουλάχιστον δύο σύμβολα[2]. Το μήκος του $e(x)$ ονομάζεται *μέγεθος* του στιγμιότυπου x. Μια (τυπική) γλώσσα L που ορίζεται σε ένα αλφάβητο Σ είναι ένα υποσύνολο του Σ^*, δηλαδή του συνόλου όλων των πεπερασμένων συμβολοσειρών (strings) του Σ συμπεριλαμβανομένης και της κενής συμβολοσειράς.

Το ισοδύναμο ενός προβλήματος απόφασης Π είναι η γλώσσα $L(\Pi, e) \subseteq \Sigma^*$. Ειδικότερα, δεδομένου ενός προβλήματος απόφασης Π και μιας κωδικοποίησης e για τα στιγμιότυπα του Π, η γλώσσα $L(\Pi, e)$ είναι το σύνολο των συμβολοσειρών $e(x) \in \Sigma^*$, όπου x είναι ένα στιγμιότυπο του Π με απάντηση NAI. Στο εξής, θα χρησιμοποιούμε τους όρους *πρόβλημα απόφασης* και *γλώσσα* ως ταυτόσημους.

[2] Θεωρούμε πάντα εύλογες κωδικοποιήσεις που χρησιμοποιούν τα σύμβολα του Σ με αποδοτικό τρόπο για να κωδικοποιήσουν τα στιγμιότυπα του προβλήματος (π.χ. κάθε αριθμός x κωδικοποιείται με $\lceil \log_{|\Sigma|} x \rceil$ ψηφία, το $e(x)$ δεν περιέχει πλεονασμό ή άχρηστη πληροφορία, κοκ.).

3 Ντετερμινιστικές Μηχανές Turing

Ως υπολογιστικό μοντέλο, υιοθετούμε την *ντετερμινιστική μηχανή Turing* (Turing Machine, TM). Ας θεωρήσουμε μια πεπερασμένη μηχανική συσκευή με μια ταινία που εκτείνεται (δυνητικά) στο άπειρο και προς τις δύο κατευθύνσεις[3], και υποδιαιρείται σε κύτταρα, που το καθένα είτε περιέχει κάποιο σύμβολο ενός αλφάβητου εισόδου Σ είτε είναι κενό (ας το συμβολίσουμε με $\sqcup$). Δηλαδή το αλφάβητο της ταινίας είναι το $\Gamma = \Sigma \cup \{\sqcup\}$. Σε κάθε χρονική στιγμή η κεφαλή της TM βρίσκεται σε ένα κύτταρο, το λεγόμενο τρέχον κύτταρο. Οι βασικές λειτουργίες μιας TM είναι:

- Διάβασε το περιεχόμενο του τρέχοντος κυττάρου.
- Γράψε κάποιο σύμβολο του Σ ή $\sqcup$ στο τρέχον κύτταρο.
- Διατήρησε το τρέχον κύτταρο (συμβολίζεται με S) ή κάνε τρέχον το αμέσως αριστερότερο (συμβολίζεται με L) ή το αμέσως δεξιότερο (συμβολίζεται με R) κύτταρο.

Μια TM έχει έναν πεπερασμένο αριθμό καταστάσεων: $Q = \{q_0, q_1, \ldots\}$. Κάποια από αυτές τις καταστάσεις, συνήθως η q_0, θεωρείται *αρχική*, και κάποιες καταστάσεις θεωρούνται *τελικές*. Ο υπολογισμός μιας TM ξεκινάει από την αρχική κατάσταση και ολοκληρώνεται όταν η μηχανή φτάσει σε κάποια από τις τελικές καταστάσεις. Η εξέλιξη του υπολογισμού προσδιορίζεται από τη *συνάρτηση μετάβασης* (transition function) δ, η οποία με βάση την παρούσα κατάσταση και το περιεχόμενο του τρέχοντος κυττάρου, καθορίζει την επόμενη κατάσταση και τις βασικές λειτουργίες που εκτελούνται. Τυπικά:

Ορισμός 1 (Ντετερμινιστική Μηχανή Turing). *Μία* ντετερμινιστική μηχανή Turing M *είναι μία διατεταγμένη πεντάδα* $M = (Q, \Sigma, \delta, q_0, F)$ *όπου:*

- Q *είναι ένα πεπερασμένο σύνολο καταστάσεων.*
- Σ *είναι το αλφάβητο εισόδου. Το αλφάβητο ταινίας είναι* $\Gamma = \Sigma \cup \{\sqcup\}$.
- $q_0 \in Q$ *είναι η αρχική κατάσταση.*
- $F \subseteq Q$ *είναι το σύνολο των τελικών καταστάσεων.*
- $\delta : (Q \setminus F) \times \Gamma \mapsto Q \times \Gamma \times \{S, L, R\}$ *είναι η συνάρτηση μετάβασης.*

Μία TM M ξεκινάει τον υπολογισμό από την αρχική κατάσταση q_0. Η ταινία περιέχει μόνο κενά και την συμβολοσειρά εισόδου $x \in \Sigma^*$, και το τρέχον κύτταρο είναι το πρώτο (αριστερότερο) σύμβολο της x. Ο υπολογισμός της M σταματάει μόνο όταν η τρέχουσα κατάσταση είναι μία από τις τελικές καταστάσεις του F. Υπάρχουν τρεις χαρακτηριστικές τελικές καταστάσεις: η YES που αντιστοιχεί στην *αποδοχή* της εισόδου, η NO που αντιστοιχεί στην *απόρριψη* της εισόδου, και η HALT που δηλώνει την ολοκλήρωση του υπολογισμού. Όταν η TM επεξεργάζεται ένα πρόβλημα απόφασης, συνήθως θεωρούμε ότι $F = \{\text{YES}, \text{NO}\}$. Όταν η μηχανή Turing υπολογίζει μια συνάρτηση, συνήθως θεωρούμε ότι $F = \{\text{HALT}\}$.

Αφού τα υπολογιστικά βήματα μιας TM M ορίζονται μονοσήμαντα από τη συνάρτηση μετάβασης, ο υπολογισμός και τελικά η απάντηση / έξοδος της M προσδιορίζεται *μονοσήμαντα* (ή *ντετερμινιστικά*) από τη συνάρτηση μετάβασης και την είσοδο x. Αν η τελική κατάσταση της M με είσοδο x είναι YES, γράφουμε $M(x) = \text{YES}$ και λέμε ότι η M *αποδέχεται* το x. Αν η τελική κατάσταση είναι NO, γράφουμε $M(x) = \text{NO}$ και λέμε ότι η M *απορρίπτει* το x. Αν η τελική κατάσταση είναι HALT, θεωρούμε ότι το αποτέλεσμα του υπολογισμού έχει καταγραφεί στην ταινία ως μια συμβολοσειρά $y \in \Sigma^*$ που αρχίζει από το τρέχον κύτταρο και οριοθετείται από κενά σύμβολα δεξιά και αριστερά. Σε αυτή την περίπτωση γράφουμε ότι $M(x) = y$. Υπάρχει ακόμη

[3] Δηλαδή η ταινία είναι, ανά πάσα στιγμή, πεπερασμένη, αλλά απεριόριστη. Αυτή η ιστορική περιγραφή του Turing ισοδυναμεί με κάτι που σήμερα καλούμε πεπερασμένη συμβολοσειρά (string).

περίπτωση ο υπολογισμός $M(x)$ να μην φτάνει ποτέ σε τελική κατάσταση, οπότε λέμε ότι η $M(x)$ δεν σταματάει.

Παράδειγμα 1. Σε κάθε TM μπορούμε να αντιστοιχήσουμε μια μερική συνάρτηση από το $\mathbb{N}$ στο $\mathbb{N}$, όπου η είσοδος και η έξοδος κωδικοποιούνται στο δυαδικό σύστημα αρίθμησης (ή γενικότερα στο σύστημα αρίθμησης με $|\Sigma|$ σύμβολα). Με βάση αυτό, θα κατασκευάσουμε μια TM που με είσοδο x, υπολογίζει το $x+1$. Η TM θα εργάζεται σύμφωνα με τον παρακάτω αλγόριθμο (τον οποίο πρώτα γράφουμε σε ψευδοκώδικα):

Κάνε τρέχον το κύτταρο με το τελευταίο σύμβολο της εισόδου x;
repeat
  Αν το τρέχον κύτταρο έχει $\sqcup$, γράψε 1 και σταμάτησε;
  Αν το τρέχον κύτταρο έχει 1, γράψε 0, κάνε τρέχον το αμέσως
    αριστερότερο κύτταρο, και κρατούμενο := 1;
  Αν το τρέχον κύτταρο έχει 0, γράψε 1, κάνε τρέχον το αμέσως
    αριστερότερο κύτταρο, και κρατούμενο := 0;
until κρατούμενο $= 0$;
Κάνε τρέχον το κύτταρο με το πρώτο σύμβολο του $x+1$ και σταμάτησε;

Ο παραπάνω αλγόριθμος υλοποιείται από μια TM με αλφάβητο εισόδου το $\Sigma = \{0, 1\}$, σύνολο καταστάσεων $Q = \{q_0, q_1, q_2, \text{HALT}\}$, αρχική κατάσταση την q_0, σύνολο τελικών καταστάσεων $F = \{\text{HALT}\}$, και συνάρτηση μετάβασης δ που περιγράφεται στον παρακάτω πίνακα[4]:

	0	1	$\sqcup$
q_0	$(q_0, 0, R)$	$(q_0, 1, R)$	$(q_1, \sqcup, L)$
q_1	$(q_2, 1, L)$	$(q_1, 0, L)$	$(\text{HALT}, 1, S)$
q_2	$(q_2, 0, L)$	$(q_2, 1, L)$	$(\text{HALT}, \sqcup, R)$

Παρατηρούμε ότι στην αρχική κατάσταση q_0, η TM θέτει ως τρέχον το κύτταρο με το τελευταίο (δεξιότερο) σύμβολο της εισόδου x, στην κατάσταση q_1 υλοποιείται η επανάληψη που αυξάνει το x κατά 1, και στην κατάσταση q_2, η TM θέτει ως τρέχον το κύτταρο με το πρώτο σύμβολο του $x+1$ και ολοκληρώνει τον υπολογισμό της.

Ακολουθεί ένα παράδειγμα λειτουργίας με είσοδο τη συμβολοσειρά 1011, όπου σε κάθε βήμα αναγράφεται μια περιγραφή της στιγμιαίας συνολικής κατάστασης (configuration) της TM. Αρχικά αναφέρεται η τρέχουσα κατάσταση της μηχανής και στη συνέχεια το περιεχόμενο της ταινίας. Το υπογραμμισμένο σύμβολο δηλώνει το τρέχον κύτταρο, ενώ τα κενά αριστερά και δεξιά δεν αναφέρονται εκτός αν το περιεχόμενο του τρέχοντος κυττάρου είναι κενό. Το σύμβολο $\vdash$ λέγεται μπάρα (turnstile) και αντιστοιχεί σε μια εφαρμογή της συνάρτησης μετάβασης δ που μεταφέρει από μια συνολική κατάσταση στην επόμενη. Ένας υπολογισμός (computation) είναι μια έγκυρη ακολουθία συνολικών καταστάσεων.

$$\begin{aligned}(q_0, \underline{1}011) &\vdash (q_0, 1\underline{0}11) \vdash (q_0, 10\underline{1}1) \vdash (q_0, 101\underline{1}) \vdash \\ (q_0, 1011\underline{\sqcup}) &\vdash (q_1, 101\underline{1}) \vdash (q_1, 10\underline{1}0) \vdash (q_1, 1\underline{0}00) \vdash \\ (q_2, \underline{1}100) &\vdash (q_2, \underline{\sqcup}1100) \vdash (\text{HALT}, \underline{1}100)\end{aligned}$$

[4] Η TM που περιγράφουμε δεν ελέγχει αν η συμβολοσειρά εισόδου είναι κενή. Η μετατροπή της TM ώστε να ελέγχει (και να σταματάει απευθείας) αν η είσοδος είναι κενή αφήνεται ως άσκηση.

Καθολική Μηχανή Turing. Σε αντίθεση με τους σύγχρονους ηλεκτρονικούς υπολογιστές, που εκτελούν διάφορα προγράμματα, μια TM εκτελεί μόνο το "πρόγραμμα" που καθορίζεται από τη συνάρτηση μετάβασης. Ο Turing όμως περιέγραψε μία *καθολική μηχανή Turing* (Universal Turing Machine), που προσομοιώνει τη λειτουργία κάθε άλλης TM.

Συγκεκριμένα, η καθολική TM U δέχεται σαν είσοδο την κωδικοποίηση μιας TM M και την είσοδο x της M. Αν η $M(x)$ σταματάει, η $U(M;x)$ σταματάει στην ίδια κατάσταση και με τα ίδια περιεχόμενα στην ταινία της όπως η $M(x)$. Ισχύει δηλαδή $U(M;x) = M(x)$. Αν η $M(x)$ δεν σταματάει, ούτε η $U(M;x)$ σταματάει.

4 Υπολογιστότητα

Μια γλώσσα L είναι *αποκρίσιμη* (decidable, ή *υπολογιστή*, computable) αν υπάρχει μια TM που αποδέχεται όλες τις συμβολοσειρές της L και απορρίπτει όλες τις συμβολειρές της $\overline{L}$[5]. Δηλαδή, για κάθε $x \in L$, $M(x) =$ YES, και για κάθε $x \notin L$, $M(x) =$ NO.

Μετά το Θεώρημα Μη Πληρότητας του Godel, πολλοί επιστήμονες (Turing, Church, Kleene, Post, Markov, κ.α.) προσπάθησαν να ξεκαθαρίσουν τις έννοιες της αποκρίσιμης γλώσσας και του υπολογιστού προβλήματος. Σε αυτή την προσπάθεια, κατέληξαν σε διαφορετικά υπολογιστικά μοντέλα (δηλ. απλούς ιδεατούς υπολογιστές), τα οποία όμως αποδείχθηκαν όλα ισοδύναμα μεταξύ τους. Αυτό οδήγησε στη διατύπωση της περίφημης *Θέσης των Church-Turing* (1936), η οποία λέει απλουστευμένα ότι *όλα τα γνωστά και τα "άγνωστα" μοντέλα της έννοιας "υπολογιστός" ή "αποκρίσιμος" είναι μηχανιστικά ισοδύναμα*. Δηλαδή δοθέντος ενός αλγορίθμου σε ένα μοντέλο για ένα πρόβλημα Π ή μια γλώσσα L, μπορούμε μηχανιστικά να κατασκευάσουμε αλγόριθμο σε ένα άλλο μοντέλο για το Π ή την L.

4.1 Μη-Υπολογιστότητα: Το Πρόβλημα Τερματισμού

Υπάρχουν προβλήματα που δεν είναι υπολογιστά. Ο λόγος είναι απλός: υπάρχουν απείρως περισσότερα προβλήματα από τις μηχανές Turing που τα υπολογίζουν. Συγκεκριμένα, οι διαφορετικές μηχανές Turing είναι αριθμήσιμες, ενώ οι διαφορετικές γλώσσες (ή ισοδύναμα, τα διαφορετικά προβλήματα) είναι μη-αριθμήσιμα. Με βάση τη Θέση των Church-Turing, οι *μη-αποκρίσιμες* (undecidable) γλώσσες αντιστοιχούν σε προβλήματα που δεν λύνονται σε κανένα υπολογιστικό μοντέλο (ανεξαρτήτως των διαθέσιμων υπολογιστικών πόρων).

Κεντρική θέση ανάμεσα σε αυτά τα προβλήματα κατέχει το *Πρόβλημα Τερματισμού* μιας TM (Halting Problem), όπου δίνονται μία TM M και μια συμβολοσειρά εισόδου x, και πρέπει να αποφανθούμε αν η $M(x)$ σταματάει.

Θεώρημα 1. *Το Πρόβλημα Τερματισμού είναι μη-αποκρίσιμο.*

Απόδειξη. Η απόδειξη χρησιμοποιεί την τεχνική της διαγωνιοποίησης. Ας υποθέσουμε ότι υπάρχει μία TM H που δέχεται σαν είσοδο μια TM M και μια συμβολοσειρά x, και σταματάει πάντα απαντώντας $H(M;x) =$ YES αν η $M(x)$ σταματάει, και $H(M;x) =$ NO διαφορετικά. Ορίζουμε μία TM K που λειτουργεί ως εξής:

– Η $K(M)$ προσομοιώνει την $H(M;M)$ ως το σημείο που η τελευταία βρίσκεται ένα βήμα πριν τον τερματισμό.

[5] Η γλώσσα $\overline{L}$ ονομάζεται *συμπλήρωμα* (complement) της L, και περιέχει όλες τις συμβολοσειρές του Σ^* που δεν ανήκουν στην L.

– Αν $H(M;M) = \text{YES}$, η $K(M)$ μπαίνει σε ατέρμονα βρόγχο (και δεν σταματάει).
– Αν $H(M;M) = \text{NO}$, η $K(M)$ σταματάει.

Εξ' ορισμού, η $K(M)$ σταματάει αν και μόνο αν η $M(M)$ δεν σταματάει. Εξετάζοντας τη λειτουργία της K με είσοδο την περιγραφή της (δηλαδή εξετάζοντας τον υπολογισμό της $K(K)$) προκύπτει αντίφαση, αφού η $K(K)$ σταματάει αν και μόνο αν η $K(K)$ δεν σταματάει. □

5 Μη Ντετερμινιστικές Μηχανές Turing

Αντί του μονοσήμαντα ορισμένου τρόπου με τον οποίο εξελίσσεται ο υπολογισμός μίας (ντετερμινιστικής) TM για δεδομένη είσοδο, ο υπολογισμός μιας *μη ντετερμινιστικής μηχανής Turing* (Nondeterministic Turing Machine, NTM) μπορεί να επιλέγει ανάμεσα σε διαφορετικές υπολογιστικές εκδοχές. Τυπικά, μια NTM N είναι μία διατεταγμένη πεντάδα $M = (Q, \Sigma, \Delta, q_0, F)$. Όπως στον Ορισμό 1, το Q είναι το (πεπερασμένο) σύνολο καταστάσεων, το $q_0 \in Q$ είναι η αρχική κατάσταση, το $F \subseteq Q$ είναι το σύνολο των τελικών καταστάσεων, και το Σ είναι το αλφάβητο εισόδου. Η ουσιαστική διαφορά είναι ότι ο υπολογισμός της N καθορίζεται από τη *σχέση μετάβασης*

$$\Delta \subseteq ((Q \setminus F) \times \Gamma) \times (Q \times \Gamma \times \{L, R, S\})$$

Για κάθε συνδυασμό κατάστασης και συμβόλου στο τρέχον κύτταρο στο $(Q \setminus F) \times \Gamma$, η σχέση μετάβασης Δ ορίζει τους επιτρεπτούς επόμενους συνδυασμούς κατάστασης, συμβόλου στην ταινία, και μετακίνησης (ή μη μετακίνησης) σε γειτονικό κύτταρο (μπορεί να υπάρχει κανένας, ένας, ή περισσότεροι τέτοιοι συνδυασμοί). Η NTM επιλέγει μη ντετερμινιστικά τον επόμενο συνδυασμό από το σύνολο των επιτρεπτών συνδυασμών που καθορίζει η σχέση μετάβασης Δ. Ισοδύναμα, η λειτουργία μιας NTM μπορεί να περιγραφεί με μία *συνάρτηση μετάβασης*

$$\delta : (Q \setminus F) \times \Gamma \mapsto \text{Pow}(Q \times \Gamma \times \{L, R, S\}),$$

η οποία παίρνει τιμές στο δυναμοσύνολο του $Q \times \Gamma \times \{L, R, S\}$.

Έτσι ο υπολογισμός μιας NTM N με είσοδο x εξελίσσεται με βάση τις υπολογιστικές εκδοχές που προκύπτουν από τη σχέση μετάβασης Δ και το x, και μπορεί να καταλήξει σε πολλές διαφορετικές απαντήσεις. Ένας τρόπος αναπαράστασης του υπολογισμού της $N(x)$ είναι το *δέντρο υπολογισμού* (computation tree). Ο υπολογισμός ξεκινά στη ρίζα του δέντρου, που αντιστοιχεί στην αρχική διαμόρφωση της $N(x)$ με κατάσταση q_0 και είσοδο x. Αν σε κάποιο σημείο του υπολογισμού, η σχέση μετάβασης Δ δίνει τη δυνατότητα μη ντετερμινιστικής επιλογής, τότε εμφανίζεται διακλάδωση στο δέντρο. Τα φύλλα του δέντρου υπολογισμού αντιστοιχούν στις απαντήσεις της NTM. Έτσι κάθε διαφορετική υπολογιστική εκδοχή που μπορεί να ακολουθήσει η $N(x)$ αντιστοιχεί σε έναν κλάδο του δέντρου υπολογισμού.

Η N *αποδέχεται* την είσοδο x αν κάποια από τις απαντήσεις της $N(x)$ είναι YES (δηλαδή υπάρχει κλάδος υπολογισμού που καταλήγει σε κατάσταση YES). Η N *απορρίπτει* την είσοδο x αν όλες οι απαντήσεις της $N(x)$ είναι NO (δηλαδή όλοι οι κλάδοι υπολογισμού καταλήγουν σε κατάσταση NO). Σε περίπτωση αποδοχής, γράφουμε $N(x) = \text{YES}$, και σε περίπτωση απόρριψης, γράφουμε $N(x) = \text{NO}$. Μια γλώσσα $L \subseteq \Sigma^*$ είναι αποκρίσιμη από μία NTM N αν για κάθε $x \in \Sigma^*$, όλοι οι κλάδοι υπολογισμού της $N(x)$ σταματούν, και $x \in L$ αν και μόνο αν $N(x) = \text{YES}$.

Εξ' ορισμού, οι TM αποτελούν μια ειδική κατηγορία NTM (όπου το δέντρο υπολογισμού εκφυλίζεται σε μία αλυσίδα). Αντίστροφα, κάθε NTM προσομοιώνεται από μία (ντετερμινιστική) TM (αλλά με εκθετική επιβάρυνση στο χρόνο εκτέλεσης, βλ. Θεώρημα 4). Έτσι, όπως άλλωστε πρεσβεύει η Θέση των Church-Turing, κάθε γλώσσα αποκρίσιμη από μια NTM είναι αποκρίσιμη (από μία ντετερμινιστική TM).

6 Υπολογιστική Πολυπλοκότητα

Η Θεωρία Υπολογιστικής Πολυπλοκότητας εστιάζει σε υπολογιστά προβλήματα, και διερευνά αν μπορούν να επιλυθούν με περιορισμένους υπολογιστικούς πόρους, όπως χρόνο υπολογισμού, χώρο μνήμης (επιπλέον του χώρου για την είσοδο και την έξοδο) για την αποθήκευση ενδιάμεσων αποτελεσμάτων, αριθμό επεξεργαστών σε παράλληλο υπολογισμό, μέγεθος και πλήθος μηνυμάτων σε κατανεμημένο υπολογισμό, κ.α. Με βάση τέτοιους περιορισμούς και άλλες υπολογιστικές παραμέτρους, ορίζονται *κλάσεις πολυπλοκότητας* (complexity classes), στις οποίες εντάσσονται προβλήματα. Στόχος είναι ο εντοπισμός αντιπροσωπευτικών προβλημάτων, που συνοψίζουν την υπολογιστική δυσκολία των προβλημάτων κάθε κλάσης (τα γνωστά και ως *πλήρη* προβλήματα), και η σύγκριση των κλάσεων πολυπλοκότητας μεταξύ τους (ως προς εγκλεισμό, διαχωρισμό, κλπ.). Αξίζει να αναφέρουμε ενδεικτικά μερικές παραμέτρους ως προς τις οποίες ορίζονται κλάσεις πολυπλοκότητας:

- *Μοντέλο υπολογισμού*: Μηχανή Turing, Μηχανή Τυχαίας Προσπέλασης (RAM), Παράλληλη Μηχανή Τυχαίας Προσπέλασης (PRAM), μονότονα κυκλώματα·
- *Λειτουργία*: ντετερμινιστική, μη ντετερμινιστική, πιθανοτική, παράλληλη·
- *Υπολογιστικοί πόροι*: αριθμός βημάτων, αριθμός συγκρίσεων, αριθμός πολλαπλασιασμών, υπολογιστικός χρόνος, μνήμη επιπλέον της εισόδου και της εξόδου, πλήθος επεξεργαστών, μέγεθος κυκλώματος, βάθος κυκλώματος·
- *Άλλα εργαλεία*: τυχαιότητα, διαλογική αλληλεπίδραση.

7 Χρονική Πολυπλοκότητα

Η υπολογιστική πολυπλοκότητα ενός προβλήματος ορίζεται με βάση την υπολογιστική πολυπλοκότητα του πιο αποδοτικού αλγόριθμου (ή ισοδύναμα, ντετερμινιστικής μηχανής Turing) που λύνει το πρόβλημα. Το βασικότερο ίσως κριτήριο για την αποδοτικότητα ενός αλγόριθμου είναι ο υπολογιστικός χρόνος.

7.1 Ντετερμινιστική Χρονική Πολυπλοκότητα και Κλάση P

Στην περίπτωση μιας ντετερμινιστικής TM M, η μονάδα μέτρησης του υπολογιστικού χρόνου είναι το στοιχειώδες υπολογιστικό βήμα της M, το οποίο συνίσταται σε μία εφαρμογή της συνάρτησης μετάβασης. Συγκεκριμένα, έστω μια TM M που σταματάει για κάθε είσοδο. Η *χρονική πολυπλοκότητα* (time complexity) της M είναι μία αύξουσα συνάρτηση $t : \mathbb{N} \mapsto \mathbb{N}$ που ορίζεται ως εξής: Για κάθε $n \in \mathbb{N}$, το $t(n)$ είναι ο μέγιστος αριθμός στοιχειωδών βημάτων που χρειάζεται η M για να σταματήσει με είσοδο μια συμβολοσειρά x μήκους n. Τότε λέμε ότι η M έχει χρονική πολυπλοκότητα (ή χρόνο εκτέλεσης) $t(n)$ ή ότι είναι μία TM $t(n)$-χρόνου.

Η χρονική πολυπλοκότητα ενός προβλήματος Π (ή μιας γλώσσας L) είναι η χρονική πολυπλοκότητα της πιο αποδοτικής (χρονικά) TM που υπολογίζει το Π (αντίστοιχα, την L). Αν υπάρχει μια TM $t(n)$-χρόνου που υπολογίζει το Π (αντίστοιχα, την L), λέμε ότι το Π (αντίστοιχα, η L) υπολογίζεται σε χρόνο $O(t(n))$.

Δεδομένης μιας αύξουσας[6] συνάρτησης $t : \mathbb{N} \mapsto \mathbb{N}$, ορίζουμε την κλάση *ντετερμινιστικής χρονικής πολυπλοκότητας* $\mathbf{DTIME}[t(n)]$ που περιλαμβάνει όλα τα υπολογιστικά προβλήματα με

[6] Εντελώς τυπικά, δεν αρκεί η $t(n)$ να είναι αύξουσα. Πρέπει να ικανοποιεί και άλλες ιδιότητες, η συζήτηση των οποίων υπερβαίνει τους σκοπούς αυτού του κεφαλαίου. Στο εξής θα αναφερόμαστε στις συναρτήσεις που ικανοποιούν όλες τις απαιτούμενες ιδιότητες ως *συναρτήσεις πολυπλοκότητας*. Οι πολυωνυμικές, οι εκθετικές, και οι λογαριθμικές συναρτήσεις είναι συναρτήσεις πολυπλοκότητας.

χρονική πολυπλοκότητα $O(t(n))$. Συγκεκριμένα, ορίζουμε:

$$\mathbf{DTIME}[t(n)] \equiv \{\Pi : \Pi \text{ πρόβλημα υπολογιστό από TM } O(t(n))\text{-χρόνου}\}$$

Ιεραρχία Κλάσεων Χρονικής Πολυπλοκότητας. Μια σημαντική παρατήρηση σχετικά με τις κλάσεις $\mathbf{DTIME}[t(n)]$ είναι ότι διευρύνονται καθώς αυξάνει η τάξη μεγέθους του $t(n)$. Συγκεκριμένα:

Θεώρημα 2 (Ιεραρχία Χρονικής Πολυπλοκότητας). *Για όλες τις συναρτήσεις πολυπλοκότητας* $t_1(n), t_2(n) \geq n$, *αν* $t_1(n)\log t_1(n) = o(t_2(n))$, *τότε*

$$\mathbf{DTIME}[t_1(n)] \subset \mathbf{DTIME}[t_2(n)]$$

Μάλιστα για την περίπτωση των TM με $k \geq 2$ ταινίες, όπου k μια ακέραια σταθερά, ο M. Furer απέδειξε μια ισχυρότερη διατύπωση του Θεωρήματος 2, η οποία ειναι γνωστή ως *Ακριβής Ιεραρχία Χρονικής Πολυπλοκότητας* (Tight Time Hierarchy). Ο Furer [6] απέδειξε ότι η ασθενέστερη συνθήκη $t_1(n) = o(t_2(n))$ είναι επαρκής για να ισχύει ο γνήσιος εγκλεισμός $\mathbf{DTIME}[t_1(n)] \subset \mathbf{DTIME}[t_2(n)]$.

΄Ενα ενδιαφέρον πόρισμα του Θεωρήματος 2 είναι ο γνήσιος εγκλεισμός των κλάσεων $\mathbf{DTIME}[n^k]$, για $k = 1, 2, 3, \ldots$. Συγκεκριμένα, ισχύει ότι

$$\mathbf{DTIME}[n] \subset \mathbf{DTIME}[n^2] \subset \mathbf{DTIME}[n^3] \subset \cdots$$

Κλάσεις $\mathbf{P}$ *και* $\mathbf{EXP}$. Δύο σημαντικές κλάσεις ντετερμινιστικής χρονικής πολυπλοκότητας είναι οι $\mathbf{P}$ και $\mathbf{EXP}$. Η κλάση $\mathbf{P}$ αποτελείται από όλα τα προβλήματα με πολυωνυμική χρονική πολυπλοκότητα, ενώ η κλάση $\mathbf{EXP}$ από όλα τα προβλήματα με εκθετική χρονική πολυπλοκότητα. Συγκεκριμένα:

$$\mathbf{P} \equiv \bigcup_{k\geq 0} \mathbf{DTIME}[n^k] \quad \text{και} \quad \mathbf{EXP} \equiv \bigcup_{k\geq 0} \mathbf{DTIME}[2^{n^k}]$$

Αφού κάθε πολυώνυμο φράσσεται άνω (ως προς την τάξη μεγέθους) από μια εκθετική συνάρτηση, τα προβλήματα που λύνονται σε πολυωνυμικό χρόνο αποτελούν υποσύνολο αυτών που λύνονται σε εκθετικό χρόνο, δηλαδή $\mathbf{P} \subseteq \mathbf{EXP}$. Μια άμεση συνέπεια του Θεωρήματος 2 είναι ότι $\mathbf{P} \subset \mathbf{EXP}$, δηλαδή υπάρχουν προβλήματα που λύνονται σε εκθετικό χρόνο αλλά δεν μπορούν να λυθούν σε πολυωνυμικό χρόνο.

Ευεπίλυτα Προβλήματα και Κλάση $\mathbf{P}$. Από τα υπολογιστά προβλήματα, κάποια απαιτούν μια εύλογη ποσότητα υπολογιστικών πόρων για να λυθούν, και θεωρούνται *ευεπίλυτα* (tractable), και κάποια όχι, και θεωρούνται *δυσεπίλυτα* (intractable). Όταν υπάρχει ένας *αποδοτικός* αλγόριθμος, δηλαδή ένας αλγόριθμος που απαιτεί εύλογη ποσότητα υπολογιστικών πόρων, για κάποιο πρόβλημα, τότε αυτό εντάσσεται στην κατηγορία των ευεπίλυτων προβλημάτων. Αντίθετα, όταν για κάποιο πρόβλημα δεν είναι γνωστός κανένας αποδοτικός αλγόριθμος, αυτό δεν εντάσσεται αυτόματα στα δυσεπίλυτα προβλήματα. Ο λόγος είναι ότι μπορεί να υπάρχει αποδοτικός αλγόριθμος, αλλά να μην έχει ανακαλυφθεί ακόμα (π.χ. ο πρώτος αλγόριθμος πολυωνυμικού χρόνου για το πρόβλημα του Γραμμικού Προγραμματισμού ανακαλύφθηκε στο τέλος της δεκαετίας του 1970 [14], και ο πρώτος αλγόριθμος πολυωνυμικού χρόνου για τον έλεγχο του αν ένας αριθμός είναι πρώτος (Primality Test) μόλις το 2002 [1]).

Η κρατούσα αντίληψη ταυτίζει την κλάση των ευεπίλυτων προβλημάτων με την κλάση $\mathbf{P}$. Αυτή η πεποίθηση αναφέρεται συχνά ως *Θέση των Cook-Karp*. Υπάρχουν αρκετά σημαντικά επιχειρήματα που στηρίζουν τη Θέση των Cook-Karp. Από πρακτική άποψη, ο ρυθμός αύξησης ενός πολυωνύμου μικρού βαθμού επιτρέπει την επίλυση αρκετά μεγάλων στιγμιότυπων ενός προβλήματος

με αντίστοιχη πολυπλοκότητα σε εύλογο χρονικό διάστημα. Επιπλέον, η ύπαρξη ενός αλγόριθμου πολυωνυμικού χρόνου επιτρέπει τη σημαντική αύξηση του μεγέθους των στιγμιότυπων που λύνει ο αλγόριθμος σε δεδομένο χρονικό διάστημα, όταν αυξάνεται η διαθέσιμη υπολογιστική ισχύς. Τα παραπάνω δεν ισχύουν για συναρτήσεις που αυξάνουν πολύ πιο γρήγορα από ένα πολυώνυμο (π.χ. εκθετικές συναρτήσεις). Επιπλέον, το σύνολο των πολυωνύμων είναι κλειστό ως προς τις πράξεις της πρόσθεσης, του πολλαπλασιασμού, και της σύνθεσης. Επομένως η κλάση $\mathbf{P}$ έχει τις αντίστοιχες ιδιότητες κλειστότητας (βλ. Άσκηση 1). Για παράδειγμα, αν συνθέσουμε δύο αλγόριθμους πολυωνυμικού χρόνου, θέτοντας την έξοδο του ενός ως είσοδο του άλλου, θα έχουμε έναν αλγόριθμο πολυωνυμικού χρόνου.

7.2 Μη Ντετερμινιστική Χρονική Πολυπλοκότητα και Κλάση NP

Η χρονική πολυπλοκότητα μιας NTM N με είσοδο x καθορίζεται από την πιο απαιτητική υπολογιστική εκδοχή, ή ισοδύναμα από το ύψος του δέντρου υπολογισμού της $N(x)$. Συγκεκριμένα, η χρονική πολυπλοκότητα μιας NTM N είναι $t(n)$, όπου t μια συνάρτηση πολυπλοκότητας, αν για κάθε συμβολοσειρά x μήκους n, το μήκος κάθε κλάδου υπολογισμού της $N(x)$ είναι μικρότερο ή ίσο του $t(n)$. Η μη ντετερμινιστική χρονική πολυπλοκότητα ενός προβλήματος απόφασης Π (αντίστοιχα, μιας γλώσσας L) είναι η χρονική πολυπλοκότητα της πιο αποδοτικής (χρονικά) NTM που υπολογίζει το Π (αντίστοιχα, την L).

Το σύνολο των προβλημάτων με μη ντετερμινιστική χρονική πολυπλοκότητα $O(t(n))$ αποτελούν την κλάση $\mathbf{NTIME}[t(n)]$. Συγκεκριμένα, ορίζουμε:

$$\mathbf{NTIME}[t(n)] \equiv \{\Pi : \Pi \text{ πρόβλημα υπολογιστό από NTM } O(t(n))\text{-χρόνου}\}$$

Όπως και οι κλάσεις $\mathbf{DTIME}[t(n)]$, οι κλάσεις μη ντετερμινιστικής χρονικής πολυπλοκότητας $\mathbf{NTIME}[t(n)]$ διευρύνονται καθώς αυξάνει η τάξη μεγέθους του $t(n)$.

Θεώρημα 3 (Ιεραρχία Μη Ντετερμινιστικής Χρονικής Πολυπλοκότητας, [20]). *Για όλες τις συναρτήσεις πολυπλοκότητας $t_1(n), t_2(n) \geq n$, αν $t_1(n+1) = o(t_2(n))$, τότε $\mathbf{NTIME}[t_1(n)] \subset \mathbf{NTIME}[t_2(n)]$.*

Αφού οι TM αποτελούν ειδική κατηγορία των NTM, ισχύει ότι $\mathbf{DTIME}[t(n)] \subseteq \mathbf{NTIME}[t(n)]$ για κάθε $t(n)$. Για κάποιες συναρτήσεις πολυπλοκότητας, έχει αποδειχθεί ότι η κλάση $\mathbf{DTIME}$ αποτελεί γνήσιο υποσύνολο της αντίστοιχης κλάσης $\mathbf{NTIME}$ (π.χ. $\mathbf{DTIME}[n] \subset \mathbf{NTIME}[n]$ [18]). Από την άλλη, ισχύει ότι (για την απόδειξη, βλ. Άσκηση 2):

Θεώρημα 4. *Κάθε πρόβλημα που υπολογίζεται από μια NTM $t(n)$-χρόνου, μπορεί να υπολογιστεί από μια TM $O(c^{t(n)})$-χρόνου, για κάποια σταθερά $c > 1$.*

Η Κλάση $\mathbf{NP}$. Μια από τις σημαντικότερες κλάσεις πολυπλοκότητας είναι η κλάση των προβλημάτων που υπολογίζονται σε πολυωνυμικό μη-ντετερμινιστικό χρόνο, γνωστή ως κλάση $\mathbf{NP}$. Τυπικά,

$$\mathbf{NP} = \bigcup_{k \geq 0} \mathbf{NTIME}[n^k]$$

Υπάρχει ένας ισοδύναμος ορισμός του $\mathbf{NP}$ που προκύπτει από το γεγονός ότι το δέντρο υπολογισμού για ένα πρόβλημα στο $\mathbf{NP}$ έχει πολυωνυμικό ύψος. Έστω μια διμελής σχέση $R \subseteq \Sigma^* \times \Sigma^*$. Η R καλείται *πολυωνυμικά αποκρίσιμη* (polynomially decidable) αν υπάρχει μία (ντετερμινιστική) TM πολυωνυμικού χρόνου που για κάθε $x, y \in \Sigma^*$, αποφασίζει αν $(x, y) \in R$. Η R καλείται *πολυωνυμικά ισορροπημένη* (polynomially balanced) αν για κάθε $(x, y) \in R$, το μήκος του y είναι πολυωνυμικό στο μήκος του x (δηλ. υπάρχει σταθερά $k \geq 1$ τέτοια ώστε $|y| \leq |x|^k$).

Θεώρημα 5 (Χαρακτηρισμός NP). *Μια γλώσσα L ανήκει στο* **NP** *αν και μόνο αν υπάρχει μια πολυωνυμικά αποκρίσιμη και πολυωνυμικά ισορροπημένη σχέση R τέτοια ώστε*

$$L = \{x \in \Sigma^* : \exists y \in \Sigma^*, (x, y) \in R\}$$

Το Θεώρημα 5 οδηγεί σε μια ενδιαφέρουσα θεώρηση του **NP**. Κάθε γλώσσα $L \in \mathbf{NP}$ έχει μια αξιοσημείωτη ιδιότητα: Όταν $x \in L$, υπάρχει ένα "πιστοποιητικό" πολυωνυμικού μήκους y που μπορεί να ελεγχθεί από μία (ντετερμινιστική) TM πολυωνυμικού χρόνου και να πιστοποιήσει ότι $x \in L$. Αντίθετα, κανένα τέτοιο "πιστοποιητικό" y δεν υπάρχει όταν $x \notin L$. Ουσιαστικά το "πιστοποιητικό" y κωδικοποιεί τις επιλογές που οδηγούν σε κάποιο κλάδο υπολογισμού με τελική κατάσταση YES. Για παράδειγμα, το "πιστοποιητικό" ότι ένα στιγμιότυπο του προβλήματος του Πλανόδιου Πωλητή έχει απάντηση NAI (και άρα ανήκει στην αντίστοιχη γλώσσα) είναι μία περιοδεία με μήκος που δεν ξεπερνά το B (βλ. Ασκήσεις 3 και 4).

Η κλάση **UP**. Με τρόπο παρόμοιο με το **NP**, μπορούμε να ορίσουμε το **UP**, μια κλάση πολυπλοκότητας που έχει άμεση σχέση με την κρυπτογραφία.

Ορισμός 2. *Μία* NTM N *χαρακτηρίζεται ως* μονοσήμαντη *αν για κάθε είσοδο x, το δέντρο υπολογισμού της $N(x)$ έχει το πολύ έναν κλάδο υπολογισμού που καταλήγει σε κατάσταση* YES. **UP** *είναι η κλάση των προβλημάτων που υπολογίζονται από μονοσήμαντες* NTM *πολυωνυμικού χρόνου.*

Εξ' ορισμού, ισχύει ότι $\mathbf{P} \subseteq \mathbf{UP} \subseteq \mathbf{NP}$. Το παρακάτω θεώρημα των J. Grollman και A.L. Selman [10], του οποίου η απόδειξη παραλείπεται, συσχετίζει την κλάση **UP** με τις *συναρτήσεις μονής κατεύθυνσης* (one-way-functions).

Θεώρημα 6 (Grollman-Selman). $\mathbf{UP} = \mathbf{P}$ *αν και μόνον αν υπάρχουν συναρτήσεις μονής κατεύθυνσης.*

Κωδικοποίηση Κλάσεων με Ποσοδείκτες. Με βάση την ιδέα του δέντρου υπολογισμού, μπορούμε να κωδικοποιήσουμε τους ορισμούς πολλών γνωστών κλάσεων πολυπλοκότητας με τους ποσοδείκτες $\exists$ και $\forall$ (και τα "πλειοψηφικά" τους ισοδύναμα, που θα ορίσουμε στην Ενότητα 10.5). Θα ξεκινήσουμε κωδικοποιώντας τους ορισμούς των κλάσεων **P** και **NP** με αυτό τον τρόπο. Το Θεώρημα 5 δείχνει ότι η κλάση **NP** ορίζεται ως:

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

όπου το $R \in \mathbf{P}$ δηλώνει ότι η $R \subseteq \Sigma^* \times \Sigma^*$ είναι μια πολυωνυμικά αποκρίσιμη και πολυωνυμικά ισορροπημένη διμελής σχέση, και το $R(x, y)$ (αντίστοιχα, το $\neg R(x, y)$) δηλώνει ότι $(x, y) \in R$ (αντίστοιχα, ότι $(x, y) \notin R$). Αφού το δέντρο υπολογισμού μιας TM εκφυλίζεται σε μονοπάτι, η κλάση **P** ορίζεται ως:

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

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 συμβολισμό $\mathbf{P} = (\forall, \forall)$ και $\mathbf{NP} = (\exists, \forall)$.

8 Αναγωγή και Πληρότητα

Οι έννοιες της *αναγωγής* (reduction) και της *πληρότητας* (completeness) είναι κεντρικές στη Θεωρία Υπολογιστικής Πολυπλοκότητας. Διαισθητικά, ένα πρόβλημα είναι *δύσκολο* (hard) για μια κλάση πολυπλοκότητας αν είναι τουλάχιστον τόσο δύσκολο να λυθεί όσο κάθε άλλο πρόβλημα της κλάσης. Ένα πρόβλημα είναι *πλήρες* (complete) για μια κλάση πολυπλοκότητας αν είναι δύσκολο για την κλάση και ταυτόχρονα μέλος της. Έτσι τα πλήρη προβλήματα συνοψίζουν την υπολογιστική δυσκολία των προβλημάτων μιας κλάσης.

Οι παραπάνω περιγραφές της δυσκολίας και της πληρότητας βασίζονται στη σύγκριση της υπολογιστικής δυσκολίας δύο προβλημάτων. Η έννοια της αναγωγής επιτρέπει αυτή τη σύγκριση. Η απλούστερη αναγωγή που θα χρησιμοποιήσουμε εδώ είναι ένας απλός μετασχηματισμός της εισόδου ενός προβλήματος σε είσοδο άλλου προβλήματος. Ένα πρόβλημα απόφασης Π_1 *ανάγεται* σε ένα πρόβλημα απόφασης Π_2 αν υπάρχει ένας (υπολογιστός) μετασχηματισμός f που για κάθε συμβολοσειρά x παράγει μια συμβολοσειρά $f(x)$ τέτοια ώστε $x \in \Pi_1$ αν και μόνο αν $f(x) \in \Pi_2$. Ο μετασχηματισμός f ονομάζεται *αναγωγή* του Π_1 στο Π_2 (συνήθως γράφουμε $\Pi_1 \leq \Pi_2$). Δηλαδή η αναγωγή f απεικονίζει τα NAI-στιγμιότυπα του Π_1 σε NAI-στιγμιότυπα του Π_2 και τα OXI-στιγμιότυπα του Π_1 σε OXI-στιγμιότυπα του Π_2. Η σύνθεση της αναγωγής f με οποιονδήποτε αλγόριθμο που υπολογίζει το Π_2 δίνει έναν αλγόριθμο που υπολογίζει το Π_1.

Οι συνέπειες μιας αναγωγής εξαρτώνται από το αν (και πόσο) αποδοτικά υπολογιστή είναι. Ειδικότερα, η αναγωγή σε *πολυωνυμικό χρόνο* του προβλήματος Π_1 στο πρόβλημα Π_2 αποτελεί έναν υπολογιστικά αποδοτικό τρόπο να μετασχηματίσουμε το Π_1 στο Π_2.

Ορισμός 3 (Πολυωνυμική Αναγωγή ή Αναγωγή κατά Karp). *Ένα πρόβλημα Π_1* ανάγεται πολυωνυμικά *σε ένα πρόβλημα Π_2 (γράφουμε $\Pi_1 \leq_{\text{poly}} \Pi_2$) αν υπάρχει συνάρτηση f υπολογιστή σε πολυωνυμικό χρόνο τέτοια ώστε για κάθε είσοδο x, $x \in \Pi_1$ αν και μόνο αν $f(x) \in \Pi_2$.*

Αντίστοιχα μπορούμε να ορίσουμε και άλλα είδη αναγωγών, όπως την *αναγωγή σε λογαριθμικό χώρο* (log-space reduction), όπου η αναγωγή f υπολογίζεται από μια TM λογαριθμικού χώρου (βλ. Ενότητα 10.1 για τον ακριβή ορισμό της χωρικής πολυπλοκότητας μιας TM).

Αν το ζητούμενο είναι ο υπολογισμός σε πολυωνυμικό χρόνο, η πολυωνυμική αναγωγή επιτρέπει τη σύγκριση της υπολογιστικής δυσκολίας δύο προβλημάτων. Αν $\Pi_1 \leq_{\text{poly}} \Pi_2$ και $\Pi_2 \in \mathbf{P}$, τότε $\Pi_1 \in \mathbf{P}$. Από την άλλη πλευρά, αν $\Pi_1 \notin \mathbf{P}$, τότε και $\Pi_2 \notin \mathbf{P}$. Μπορούμε λοιπόν να πούμε ότι το Π_2 είναι τουλάχιστον τόσο δύσκολο όσο το Π_1 (για τον υπολογισμό σε πολυωνυμικό χρόνο, βλ. Ασκήσεις 6 και 7).

Ιδιαίτερο ενδιαφέρον παρουσιάζουν τα προβλήματα μιας κλάσης που είναι τουλάχιστον τόσο δύσκολο να υπολογιστούν όσο κάθε άλλο πρόβλημα στην κλάση.

Ορισμός 4. *Έστω $\mathbf{C}$ μια κλάση πολυπλοκότητας. Ένα πρόβλημα Π ονομάζεται* $\mathbf{C}$-δύσκολο *($\mathbf{C}$-hard) ως προς την αναγωγή $\leq$ αν για κάθε πρόβλημα $\Pi' \in \mathbf{C}$, $\Pi' \leq \Pi$. Αν επιπλέον $\Pi \in \mathbf{C}$, το Π ονομάζεται* $\mathbf{C}$-πλήρες *($\mathbf{C}$-complete).*

Τα πλήρη προβλήματα αποτελούν ένα πολύ σημαντικό εργαλείο για τη Θεωρία Υπολογιστικής Πολυπλοκότητας. Η πολυπλοκότητα ενός προβλήματος θεωρείται καθορισμένη όταν αυτό αποδειχθεί πλήρες για κάποια κλάση ως προς κάποια εύλογη αναγωγή. Τα πλήρη προβλήματα μια κλάσης $\mathbf{C}$ συνοψίζουν την υπολογιστική δυσκολία των προβλημάτων της $\mathbf{C}$ και αποτελούν το σύνδεσμο της Θεωρίας Υπολογιστικής Πολυπλοκότητας με τη Θεωρία Αλγορίθμων και τις εφαρμογές (βλ. ακόμη Άσκηση 9). Μάλιστα, η ύπαρξη σημαντικών πρακτικών προβλημάτων που είναι πλήρη για κάποια κλάση προσδίδει στην κλάση πρακτική αξία, που δεν είναι πάντα σαφής από τον ορισμό της.

9 NP-Πληρότητα

Η κλάση **P** είναι υποσύνολο της κλάσης **NP** επειδή οι TM είναι ειδική περίπτωση των NTM. Από την άλλη, υπάρχουν πολλά και σημαντικά προβλήματα στο **NP** για τα οποία δεν είναι γνωστό αν ανήκουν στο **P** ή όχι, όπως το πρόβλημα του Πλανόδιου Πωλητή και το πρόβλημα της Ικανοποιησιμότητας λογικών προτάσεων (SAT) σε Συζευκτική Κανονική Μορφή (CNF). Το σημαντικότερο ανοικτό πρόβλημα στη Επιστήμη των Υπολογιστών αφορά στο αν η κλάση **P** αποτελεί γνήσιο υποσύνολο της κλάσης **NP**. Το ερώτημα αυτό έχει τεράστια πρακτική και θεωρητική σημασία, αλλά προς το παρόν δεν μοιάζει ώριμο να απαντηθεί. Παρόλα αυτά, αποτελεί κοινή επιστημονική πεποίθηση ότι ο εγκλεισμός είναι γνήσιος.

Αν όντως το **P** διαφέρει από το **NP**, υπάρχουν κάποια προβλήματα στο **NP** που δεν λύνονται σε πολυωνυμικό χρόνο. Μεταξύ αυτών, θα είναι όλα τα προβλήματα που είναι **NP**-πλήρη ως προς την πολυωνυμική αναγωγή (τα οποία είναι πολλά και σημαντικά).

Θεώρημα 7. *Έστω Π ένα πρόβλημα* **NP***-πλήρες ως προς την πολυωνυμική αναγωγή. Ισχύει ότι* $\Pi \in \mathbf{P}$ *αν και μόνο αν ισχύει ότι* $\mathbf{P} = \mathbf{NP}$.

Δυσεπίλυτα Προβλήματα και Κλάση **NP**. Μια άμεση συνέπεια του Θεωρήματος 7 είναι ότι αν $\mathbf{P} \neq \mathbf{NP}$, κανένα **NP**-πλήρες πρόβλημα δεν λύνεται σε (ντετερμινιστικό) πολυωνυμικό χρόνο. Δηλαδή αν, όπως πιστεύεται, $\mathbf{P} \neq \mathbf{NP}$, όλα τα **NP**-πλήρη προβλήματα ανήκουν στο $\mathbf{NP} \setminus \mathbf{P}$. Επιπλέον, γνωρίζουμε ότι αν $\mathbf{P} \neq \mathbf{NP}$, υπάρχουν κάποια προβλήματα, που λέγονται **NP**-*ενδιάμεσα* (**NP**-intermediate) προβλήματα, τα οποία επίσης ανήκουν στο $\mathbf{NP} \setminus \mathbf{P}$ αλλά δεν είναι **NP**-πλήρη (βλ. [15]). Από την άλλη, αν βρεθεί αλγόριθμος πολυωνυμικού χρόνου για κάποιο **NP**-πλήρες πρόβλημα, από αυτόν θα προκύψουν (με εφαρμογή των αντίστοιχων πολυωνυμικών αναγωγών) αλγόριθμοι πολυωνυμικού χρόνου για όλα τα προβλήματα στο **NP**. Για αυτούς τους λόγους, τα **NP**-πλήρη προβλήματα θεωρούνται δυσεπίλυτα (με την έννοια ότι κατά κοινή πεποίθηση δεν ανήκουν στο **P**), αν και κάτι τέτοιο δεν έχει αποδειχθεί.

Αυτό εξηγεί σε ένα βαθμό γιατί η Θεωρία της **NP**-πληρότητας έχει πολλές και σημαντικές εφαρμογές. Έστω ότι προσπαθούμε να σχεδιάσουμε έναν αποδοτικό αλγόριθμο για κάποιο πρόβλημα που ανήκει στο **NP** (πολλά σημαντικά προβλήματα ανήκουν στο **NP**). Έχοντας ταυτίσει τους αποδοτικούς αλγόριθμους με τους αλγόριθμους πολυωνυμικού χρόνου, είναι σημαντικό να αποφασίσουμε αν το πρόβλημα λύνεται σε πολυωνυμικό χρόνο (δηλ. ανήκει στο **P**) ή είναι **NP**-πλήρες (οπότε κατά κοινή επιστημονική πεποίθηση, δεν ανήκει στο **P**). Μετά από μερικές αποτυχημένες προσπάθειες για αλγόριθμο πολυωνυμικού χρόνου, συνήθως στρέφουμε το ενδιαφέρον μας στη διατύπωση απόδειξης **NP**-πληρότητας. Αρκετά συχνά, η αιτία που δεν καταλήγουμε σε αλγόριθμο πολυωνυμικού χρόνου οδηγεί σε απόδειξη **NP**-πληρότητας. Άλλες φορές, η αιτία που δεν καταφέρνουμε να διατυπώσουμε μια απόδειξη **NP**-πληρότητας οδηγεί σε αλγόριθμο πολυωνυμικού χρόνου.

9.1 NP-Πλήρη Προβλήματα

Ικανοποιησιμότητα Λογικών Προτάσεων (SAT). Το πρώτο πρόβλημα που αποδείχτηκε πλήρες για το **NP** είναι αυτό της Ικανοποιησιμότητας λογικών προτάσεων σε Συζευκτική Κανονική Μορφή. Στο πρόβλημα της Ικανοποιησιμότητας, δίνεται μια λογική πρόταση φ σε CNF, και ζητείται να αποφανθούμε αν η φ είναι ικανοποιήσιμη (satisfiable), δηλαδή αν υπάρχει μια αποτίμηση των λογικών μεταβλητών που κάνει την φ αληθή. Το πρόβλημα της Ικανοποιησιμότητας αποδείχτηκε **NP**-πλήρες στις αρχές της δεκαετίας του 1970 από τον S. Cook [4].

Θεώρημα 8 (Cook). *Το πρόβλημα της Ικανοποιησιμότητας (SAT) είναι* **NP***-πλήρες.*

Αποδεικνύεται ευκολα ότι η Ικανοποιησιμότητα ανήκει στο **NP** (βλ Άσκηση 4). Ο Cook έδειξε πως μπορεί να κωδικοποιηθεί ο υπολογισμός μιας NTM N με είσοδο x σε μία λογική πρόταση $\varphi_{N,x}$ σε CNF, η οποία έχει πολυωνυμικό μήκος και μπορεί να υπολογισθεί από την περιγραφή της N και του x σε πολυωνυμικό χρόνο. Η κατασκευή εγγυάται ότι η $\varphi_{N,x}$ είναι ικανοποιήσιμη αν και μόνο αν $N(x) = \text{ΝΑΙ}$. Επομένως κάθε γλώσσα $L \in \mathbf{NP}$ ανάγεται πολυωνυμικά στην Ικανοποιησιμότητα.

Ειδική περίπτωση του προβλήματος της Ικανοποιησιμότητας αποτελεί το πρόβλημα της k-Ικανοποιησιμότητας (k-SAT), όπου η λογική πρόταση φ έχει το πολύ k άτομα (literals) σε κάθε φράση (clause). Η 2-Ικανοποιησιμότητα ανάγεται πολυωνυμικά στο πρόβλημα της εύρεσης μονοπατιού μεταξύ ζεύγους κορυφών σε κατευθυνόμενο γράφημα, και ανήκει στο **P**. Θα αποδείξουμε ότι:

Θεώρημα 9. *Το πρόβλημα της 3-Ικανοποιησιμότητας (3-SAT) είναι* **NP***-πλήρες.*

Απόδειξη. Το πρόβλημα της 3-Ικανοποιησιμότητας ανήκει στο **NP** ως ειδική περίπτωση της Ικανοποιησιμότητας. Θα αποδείξουμε ότι η Ικανοποιησιμότητα ανάγεται πολυωνυμικά στην 3-Ικανοποιησιμότητα (δηλαδή ότι ο περιορισμός του μέγιστου αριθμού ατόμων ανά φράση σε 3 δεν κάνει το πρόβλημα πιο εύκολο, όσον αφορά στον πολυωνυμικό υπολογισμό). Συγκεκριμένα, θα δείξουμε ότι μια λογική πρόταση φ σε CNF μπορεί να μετατραπεί σε πολυωνυμικό χρόνο σε μια ισοδύναμη ως προς την ικανοποιησιμότητα λογική πρόταση φ' σε CNF, με την φ' να έχει 3 το πολύ άτομα σε κάθε φράση. Η μετατροπή συνίσταται στην αντικατάσταση κάθε φράσης c_j της φ με $k \geq 4$ άτομα από μια ομάδα φράσεων c'_j με 3 άτομα ανά φράση.

Συγκεκριμένα, έστω φράση $c_j = \ell_{j_1} \vee \ldots \vee \ell_{j_k}$ της φ με $k \geq 4$ άτομα. Ορίζουμε $k-3$ νέες μεταβλητές $z_{j_1}, \ldots, z_{j_{k-3}}$, και αντικαθιστούμε τη φράση c_j με τις φράσεις:

$$\begin{aligned} c'_j = (\ell_{j_1} \vee \ell_{j_2} \vee z_{j_1}) \wedge (\neg z_{j_1} \vee \ell_{j_3} \vee z_{j_2}) \wedge (\neg z_{j_2} \vee \ell_{j_4} \vee z_{j_3}) \wedge \ldots \\ \wedge (\neg z_{j_{k-4}} \vee \ell_{j_{k-2}} \vee z_{j_{k-3}}) \wedge (\neg z_{j_{k-3}} \vee \ell_{j_{k-1}} \vee \ell_{j_k}) \end{aligned}$$

Η φ' προκύπτει από τη φ αντικαθιστώντας κάθε φράση c_j της φ με $k \geq 4$ άτομα με την αντίστοιχη ομάδα φράσεων c'_j. Η φ' μπορεί να υπολογιστεί σε πολυωνυμικό χρόνο (ως προς το μήκος της αναπαράστασης της φ).

Θα αποδείξουμε ότι η φ είναι ικανοποιήσιμη αν και μόνο αν η φ' είναι ικανοποιήσιμη (δηλ. ότι οι φ και φ' είναι ισοδύναμες ως προς την ικανοποιησιμότητα). Αρκεί να δείξουμε ότι η ισοδυναμία ισχύει για κάθε ζευγάρι φράσεων c_j και c'_j. Έστω μια αποτίμηση που ικανοποιεί τη φράση c_j, και έστω ℓ_p το πρώτο άτομο που γίνεται αληθές από αυτή την αποτίμηση. Η ομάδα φράσεων c'_j ικανοποιείται αν συμπληρώσουμε την αποτίμηση με τις ακόλουθες τιμές για τις μεταβλητές z_{j_i}, $i = 1, \ldots, k-3$:

$$z_{j_i} = \begin{cases} 1 & \text{αν } i < p-1 \\ 0 & \text{αν } i \geq p-1 \end{cases}$$

Για το αντίστροφο, παρατηρούμε ότι η ομάδα φράσεων c'_j είναι ικανοποιήσιμη μόνο αν τουλάχιστον ένα από τα άτομα $\ell_1, \ldots, \ell_k$ είναι αληθές. Επομένως, κάθε αποτίμηση που ικανοποιεί την ομάδα φράσεων c'_j ικανοποιεί και τη φράση c_j. □

Η μεθοδολογία της απόδειξης του Θεωρήματος 9 χρησιμοποιείται συχνά στις αποδείξεις **NP**-πληρότητας. Ειδικότερα, για να αναγάγουμε ένα πρόβλημα Π_1 σε ένα άλλο Π_2 επιχειρούμε να μετασχηματίσουμε τμήματα των στιγμιοτύπων του Π_1 σε τμήματα των στιγμιοτύπων του Π_2 (π.χ.

μετασχηματισμός μιας φράσης με πολλά άτομα σε ομάδα φράσεων με 3 άτομα σε κάθε φράση). Αν ένας τέτοιος μετασχηματισμός διατηρεί τις επιθυμητές ιδιότητες (π.χ. ικανοποιησιμότητα), μπορεί να εφαρμοστεί τμηματικά και να αποτελέσει τη βάση για μια πολυωνυμική αναγωγή από το Π_1 στο Π_2. Αν το Π_1 είναι ένα γνωστό **NP**-πλήρες πρόβλημα και το Π_2 ανήκει στο **NP**, μια πολυωνυμική αναγωγή από το Π_1 στο Π_2 αποδεικνύει ότι και το Π_2 είναι **NP**-πλήρες (με την ίδια μεθοδολογία, ο αναγνώστης μπορεί να αναγάγει το πρόβλημα της 3-Ικανοποιησιμότητας στο πρόβλημα της Μέγιστης 2-Ικανοποιησιμότητας (MAX 2-SAT), αποδεικνύοντας έτσι ότι το τελευταίο είναι **NP**-πλήρες, βλ. Άσκηση 10).

Επιλέγοντας κατάλληλα το αρχικό **NP**-πλήρες πρόβλημα και εξειδικεύοντας κατάλληλα την παραπάνω μεθοδολογία, έχει αποδειχθεί η **NP**-πληρότητα μιας πληθώρας σημαντικών προβλημάτων. Από την εξαιρετικά μακρά λίστα των σημαντικών προβλημάτων που είναι πλήρη (ή δύσκολα) για την κλάση **NP** (βλ. π.χ. [7]), εδώ αναφέρουμε ενδεικτικά τα προβλήματα του Πλανόδιου Πωλητή (TSP), της *Διαμέρισης* (Partition) ενός συνόλου βαρών σε δύο ισοβαρή υποσύνολα, του *Διακριτού Σακιδίου* (Knapsack), του *Συνόλου Ανεξαρτησίας* (Independent Set, βλ. Άσκηση 11), του *Καλύμματος Κορυφών* (Vertex Cover, βλ. Άσκηση 12), της *Μέγιστης Τομής* (Maximum Cut), του *κύκλου Hamilton*, και του *Χρωματικού Αριθμού* (Chromatic Number) σε ένα γράφημα.

10 Άλλες Κλάσεις Πολυπλοκότητας

Με βάση τα κριτήρια που παρουσιάστηκαν στην Ενότητα 6, και κάποια άλλα κριτήρια των οποίων η παρουσίαση υπερβαίνει τους σκοπούς αυτού του κεφαλαίου, ορίζονται πολλές και σημαντικές κλάσεις υπολογιστικής πολυπλοκότητας. Στη συνέχεια, επιχειρούμε μια σύντομη ανασκόπηση των σημαντικότερων από αυτές και των σχέσεων μεταξύ τους.

10.1 Χωρική Πολυπλοκότητα

Εκτός του χρόνου εκτέλεσης, ένα βασικό κριτήριο για την αποδοτικότητα ενός αλγόριθμου είναι ο *χώρος αποθήκευσης* (μνήμη) που χρησιμοποιείται για την καταχώρηση των ενδιάμεσων αποτελεσμάτων.

Αν θεωρήσουμε μια TM M που σταματάει για κάθε είσοδο, η *χωρική πολυπλοκότητα* (space complexity) της M είναι μία συνάρτηση πολυπλοκότητας s τέτοια ώστε για κάθε $n \in \mathbb{N}$, $s(n)$ είναι ο μέγιστος αριθμός κυττάρων που χρησιμοποιεί η M για την αποθήκευση των ενδιάμεσων αποτελεσμάτων του υπολογισμού της όταν η συμβολοσειρά εισόδου έχει μήκος n. Τότε λέμε ότι η M χρησιμοποιεί χώρο (ή έχει χωρική πολυπλοκότητα) $s(n)$ ή ότι είναι μία TM $s(n)$-χώρου. Να σημειώσουμε ότι για τον υπολογισμό της χωρικής πολυπλοκότητας, δεν λαμβάνουμε υπόψη τα κύτταρα όπου αποθηκεύεται η είσοδος και η έξοδος της M, και θεωρούμε ότι η M χρησιμοποιεί τα κύτταρα όπου είναι τοποθετημένη η είσοδος μόνο για ανάγνωση, και τα κύτταρα όπου καταγράφεται η έξοδος μόνο για εγγραφή.

Σε αντιστοιχία με την χρονική πολυπλοκότητα, η χωρική πολυπλοκότητα μιας NTM N με είσοδο x καθορίζεται από τη χωρική πολυπλοκότητα του πιο απαιτητικού (σε χώρο αποθήκευσης ενδιάμεσων αποτελεσμάτων) κλάδου υπολογισμού της $N(x)$. Συγκεκριμένα, η χωρική πολυπλοκότητα μιας NTM N είναι $s(n)$ αν για κάθε συμβολοσειρά x μήκους n, ο αριθμός κυττάρων που χρησιμοποιεί κάθε κλάδος υπολογισμού της $N(x)$ είναι μικρότερος ή ίσος του $s(n)$.

Σε αντιστοιχία με τις κλάσεις χρονικής πολυπλοκότητας **DTIME** και **NTIME**, ορίζουμε τις κλάσεις ντετερμινιστικής και μη ντετερμινιστικής χωρικής πολυπλοκότητας **DSPACE** και **NSPACE**. Στην κλάση $\mathbf{DSPACE}[s(n)]$ εντάσσεται κάθε πρόβλημα υπολογιστό από μια (ντετερμινιστική) TM $s(n)$-χώρου. Στην $\mathbf{NSPACE}[s(n)]$ εντάσσεται κάθε πρόβλημα υπολογιστό από

μια NTM $s(n)$-χώρου. Ορίζουμε ακόμη τις κλάσεις ντετερμινιστικού και μη ντετερμινιστικού πολυωνυμικού και λογαριθμικού χώρου:

- $\mathbf{PSPACE} = \bigcup_{k\geq 0} \mathbf{DSPACE}[n^k]$ και $\mathbf{NPSPACE} = \bigcup_{k\geq 0} \mathbf{NSPACE}[n^k]$
- $\mathbf{L} = \mathbf{DSPACE}[\log n]$ και $\mathbf{NL} = \mathbf{NSPACE}[\log n]$

Για κάθε συνάρτηση πολυπλοκότητας $s(n)$, $\mathbf{DSPACE}[s(n)] \subseteq \mathbf{NSPACE}[s(n)]$, και άρα $\mathbf{PSPACE} \subseteq \mathbf{NPSPACE}$ και $\mathbf{L} \subseteq \mathbf{NL}$. Επίσης, αφού σε χρόνο $s(n)$ δεν μπορούν να εξετασθούν περισσότερα από $s(n)$ κύτταρα στην ταινία μιας TM, ισχύει ότι $\mathbf{NTIME}[s(n)] \subseteq \mathbf{DSPACE}[s(n)]$. Συνεπώς $\mathbf{NP} \subseteq \mathbf{PSPACE}$. Επιπλέον, αποδεικνύεται ότι $\mathbf{NSPACE}[s(n)] \subseteq \mathbf{DTIME}[c^{\log n + s(n)}]$, για κάποια σταθερά $c > 1$.

Όπως και οι κλάσεις χρονικής πολυπλοκότητας, οι κλάσεις $\mathbf{DSPACE}[s(n)]$ και $\mathbf{NSPACE}[s(n)]$ διευρύνονται καθώς αυξάνει η τάξη μεγέθους του $s(n)$.

Θεώρημα 10 (Ιεραρχία Χωρικής Πολυπλοκότητας, [11]). *Για όλες τις συναρτήσεις πολυπλοκότητας $s_1(n), s_2(n) \geq \log n$, αν $s_1(n) = o(s_2(n))$, τότε*

$$\mathbf{DSPACE}[s_1(n)] \subset \mathbf{DSPACE}[s_2(n)]$$

Θεώρημα 11 (Ιεραρχία Μη Ντετερμινιστικής Χωρικής Πολυπλοκότητας, [20]). *Για όλες τις συναρτήσεις πολυπλοκότητας $s_1(n), s_2(n) \geq \log n$, αν $s_1(n) = o(s_2(n))$, τότε $\mathbf{NSPACE}[s_1(n)] \subset \mathbf{NSPACE}[s_2(n)]$.*

Ως συνέπεια, οι κλάσεις $\mathbf{L}$ και $\mathbf{NL}$ αποτελούν *γνήσια υποσύνολα* της $\mathbf{PSPACE}$.

Είναι εξαιρετικά ενδιαφέρον ότι οι κλάσεις ντετερμινιστικού και μη ντετερμινιστικού πολυωνυμικού χώρου ταυτίζονται (δηλ. $\mathbf{PSPACE} = \mathbf{NPSPACE}$), το οποίο προκύπτει ως άμεση συνέπεια από το ακόλουθο αποτέλεσμα του W.J. Savitch [19]:

Θεώρημα 12 (Savitch). *Για κάθε συνάρτηση πολυπλοκότητας $s(n) \geq \log n$,*

$$\mathbf{NSPACE}[s(n)] \subseteq \mathbf{DSPACE}[s^2(n)]$$

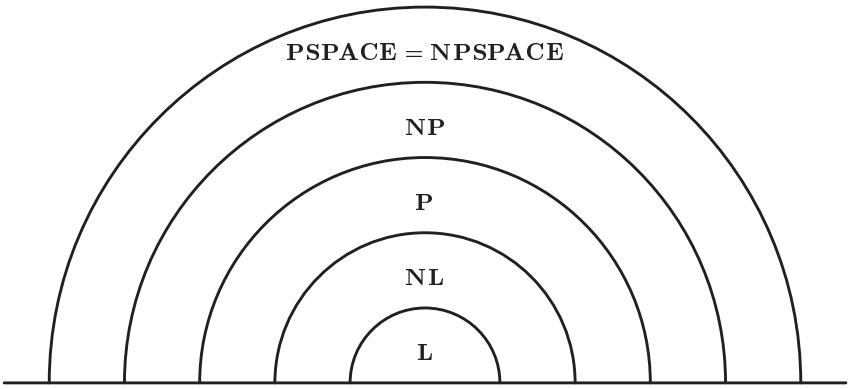
Από τα παραπάνω, προκύπτει η εξής ιεραρχία για τις κλάσεις υπολογιστικής πολυπλοκότητας που έχουμε δει μέχρι τώρα:

$$\mathbf{L} \subseteq \mathbf{NL} \subseteq \mathbf{P} \subseteq \mathbf{NP} \subseteq \mathbf{PSPACE} = \mathbf{NPSPACE} \tag{1}$$

Γνωρίζουμε ότι $\mathbf{L} \subset \mathbf{PSPACE}$ και $\mathbf{NL} \subset \mathbf{PSPACE}$, άρα τουλάχιστον ένας από τους παραπάνω εγκλεισμούς είναι γνήσιος. Πιστεύουμε δε ότι όλοι οι παραπάνω εγκλεισμοί είναι γνήσιοι. Οι αποδείξεις της γνησιότητας των παραπάνω εγκλεισμών αποτελούν σημαντικά ανοικτά προβλήματα της Θεωρίας Υπολογιστικής Πολυπλοκότητας.

10.2 Πολυπλοκότητα Συναρτήσεων

Οι κλάσεις χρονικής και χωρικής πολυπλοκότητας που ορίσαμε στις προηγούμενες ενότητες αφορούν σε προβλήματα απόφασης. Αντίστοιχα, μπορούμε να ορίσουμε κλάσεις πολυπλοκότητας που περιλαμβάνουν *συναρτήσεις*. Σημαντικές είναι η κλάση $\mathbf{FP}$, που περιλαμβάνει τις συναρτήσεις που υπολογίζονται από (ντετερμινιστικές) TM πολυωνυμικού χρόνου, και η κλάση $\mathbf{FL}$, που περιλαμβάνει συναρτήσεις που υπολογίζονται από ντετερμινιστικές TM λογαριθμικού χώρου. Έτσι κάθε πολυωνυμική αναγωγή ανήκει στην κλάση $\mathbf{FP}$, και κάθε αναγωγή λογαριθμικού χώρου ανήκει στην κλάση $\mathbf{FL}$.



Σχήμα 1. Σχέσεις Κλάσεων Χρονικής και Χωρικής Πολυπλοκότητας

10.3 Συμπληρωματικές Κλάσεις Πολυπλοκότητας

Για μια κλάση πολυπλοκότητας $\mathbf{C}$, ορίζουμε την κλάση $\mathbf{coC} = \{\overline{L} : L \in \mathbf{C}\}$ που περιλαμβάνει το συμπλήρωμα κάθε γλώσσας L που ανήκει στην $\mathbf{C}$. Για παράδειγμα, η κλάση $\mathbf{coNP}$ αποτελείται από τα συμπληρώματα των γλωσσών στο $\mathbf{NP}$ (όσον αφορά στην κωδικοποίηση των κλάσεων με ποσοδείκτες, βλ. 7.2, έχουμε ότι $\mathbf{coNP} = (\forall, \exists)$). Έτσι αφού το πρόβλημα της Ικανοποιησιμότητας ανήκει στο $\mathbf{NP}$, το πρόβλημα της *Μη Ικανοποιησιμότητας*, δηλαδή να αποφανθούμε αν μια λογική πρόταση σε CNF είναι αντίφαση, ανήκει στο $\mathbf{coNP}$.

Έχει ενδιαφέρον να δούμε ποιες κλάσεις πολυπλοκότητας είναι κλειστές ως προς συμπλήρωμα, δηλαδή για ποιες κλάσεις $\mathbf{C}$ ισχύει ότι $\mathbf{C} = \mathbf{coC}$. Γενικά, οι ντετερμινιστικές κλάσεις πολυπλοκότητας (είτε χρονικές, είτε χωρικές) είναι κλειστές ως προς συμπλήρωμα, αφού για να υπολογίσουμε την γλώσσα $\overline{L}$ αρκεί να αντιστρέψουμε την απάντηση μιας TM που υπολογίζει την L, παραμένοντας στην ίδια πολυπλοκότητα. Έτσι οι κλάσεις $\mathbf{DTIME}[t(n)]$, $\mathbf{P}$, $\mathbf{DSPACE}[s(n)]$, και $\mathbf{PSPACE}$ είναι κλειστές ως προς συμπλήρωμα. Το πρόβλημα είναι ανοιχτό για τις κλάσεις μη ντετερμινιστικής χρονικής πολυπλοκότητας. Για παράδειγμα δεν γνωρίζουμε αν $\mathbf{NP} \neq \mathbf{coNP}$. Γνωρίζουμε βέβαια ότι $\mathbf{P} \subseteq \mathbf{NP} \cap \mathbf{coNP}$, αφού το $\mathbf{P}$ αποτελεί υποσύνολο του $\mathbf{NP}$ και είναι κλειστό ως προς το συμπλήρωμα. Έτσι το ερώτημα αν $\mathbf{NP} \neq \mathbf{coNP}$ συνδέεται με το ερώτημα αν $\mathbf{P} \neq \mathbf{NP}$, αφού αν $\mathbf{NP} \neq \mathbf{coNP}$, τότε $\mathbf{P} \neq \mathbf{NP}$.

Ενώ η κατάσταση φαινόταν να είναι παρόμοια και για τις κλάσεις μη ντετερμινιστικής χωρικής πολυπλοκότητας, στα μέσα της δεκαετίας του 1980, οι N. Immerman και R. Szelepcsenyi απέδειξαν (σε ανεξάρτητες εργασίες, βλ. [13,23]) ότι για κάθε συνάρτηση πολυπλοκότητας $s(n) \geq \log n$, η κλάση $\mathbf{NSPACE}[s(n)]$είναι κλειστή ως προς συμπλήρωμα. Επομένως, $\mathbf{NPSPACE} = \mathbf{coNPSPACE}$ (το οποίο προκύπτει άλλωστε από την ισότητα $\mathbf{NPSPACE}$ και $\mathbf{PSPACE}$) και $\mathbf{NL} = \mathbf{coNL}$.

10.4 Κλάσεις Πολυπλοκότητας για Πιθανοτικούς Αλγόριθμους

Συνεχίζουμε με τον ορισμό κάποιων σημαντικών κλάσεων χρονικής πολυπλοκότητας για *πιθανοτικούς αλγόριθμους* (randomized algorithms), που λειτουργούν με βάση τυχαίες επιλογές. Οι κλάσεις αυτές ορίζονται με αναφορά στο δέντρο υπολογισμού μιας (μη ντετερμινιστικής) NTM N, για την οποία θεωρούμε, χωρίς βλάβη της γενικότητας, ότι για κάθε είσοδο x, το δέντρο υπολογισμού της $N(x)$ είναι ένα πλήρες και γεμάτο δυαδικό δέντρο (δηλ. κάθε κόμβος του είτε είναι φύλλο είτε έχει 2 παιδιά, και όλα τα φύλλα βρίσκονται στο ίδιο επίπεδο). Ας σημειώσουμε ότι,

αφού θεωρούμε πιθανοτικούς αλγόριθμους, οι δυνατές απαντήσεις για έναν κλάδο υπολογισμού μιας NTM είναι πλέον τρεις. Ένας κλάδος υπολογισμού μπορεί να καταλήξει σε κατάσταση YES ή NO, αλλά μπορεί να καταλήξει και σε κατάσταση UNK, που σημαίνει δεν μπορεί να αποφασίσει αν πρέπει να αποδεχθεί την είσοδο ή όχι.

Στην κλάση **BPP** (από τα αρχικά του Bounded Probabilistic Polynomial) εντάσσονται οι γλώσσες L για τις οποίες υπάρχει μια NTM N πολυωνυμικού χρόνου τέτοια ώστε για κάθε είσοδο x:

- Αν $x \in L$, τότε $\mathbb{P}\mathrm{r}[N(x) = \text{YES}] \geq 1/2 + \varepsilon$, και
- αν $x \notin L$, τότε $\mathbb{P}\mathrm{r}[N(x) = \text{NO}] \geq 1/2 + \varepsilon$,

όπου $\varepsilon > 0$ μία σταθερά, και $\mathbb{P}\mathrm{r}[N(x) = \text{YES}]$ (αντίστοιχα, $\mathbb{P}\mathrm{r}[N(x) = \text{NO}]$) είναι ο λόγος των φύλλων στο δέντρο υπολογισμού της $N(x)$ σε κατάσταση YES (αντίστοιχα, NO) προς το σύνολο των φύλλων. Δηλαδή, για κάθε είσοδο x, οι κλάδοι υπολογισμού της $N(x)$ που καταλήγουν στη σωστή απάντηση είναι *σημαντικά περισσότεροι* αυτών που καταλήγουν στη λάθος απάντηση[7].

Ο ορισμός της κλάσης **PP** μοιάζει με αυτόν της **BPP**. Η μόνη (αλλά πολύ σημαντική) διαφορά είναι ότι στον ορισμό της **PP**, οι κλάδοι υπολογισμού της $N(x)$ που καταλήγουν στη σωστή απάντηση αρκεί να ξεπερνούν (έστω και κατά 1) αυτούς που καταλήγουν στη λάθος απάντηση. Έτσι στην κλάση **PP** εντάσσονται οι γλώσσες L για τις οποίες υπάρχει μια NTM N πολυωνυμικού χρόνου τέτοια ώστε για κάθε είσοδο x, $x \in L$ αν και μόνο αν $\mathbb{P}\mathrm{r}[N(x) = \text{YES}] > 1/2$. Εξ' ορισμού, $\mathbf{BPP} \subseteq \mathbf{PP}$, ενώ αποδεικνύεται ότι το **PP** περιλαμβάνει την κλάση **NP** (βλ. Άσκηση 13). Μάλιστα, επειδή στον ορισμό της **PP** η απόφαση για αποδοχή λαμβάνεται με *απλή* πλειοψηφία, δεν μπορούμε με πολυωνυμικό αριθμό επαναλήψεων του ίδιου αλγόριθμου να αυξήσουμε την πιθανότητα επιτυχίας σε τιμές πολυωνυμικά κοντά στο 1 (όπως συμβαίνει με την περίπτωση των **BPP** αλγορίθμων).

Ο ορισμός της κλάσης **RP** (από τα αρχικά του Randomized Polynomial) προκύπτει από τον ορισμό του **BPP** αν επιτρέψουμε πιθανότητα λάθους μόνο όταν $x \in L$. Ειδικότερα, στην κλάση **RP** εντάσσονται οι γλώσσες L για τις οποίες υπάρχει μια NTM N πολυωνυμικού χρόνου τέτοια ώστε για κάθε είσοδο x:

- Αν $x \in L$, τότε $\mathbb{P}\mathrm{r}[N(x) = \text{YES}] \geq 1/2 + \varepsilon$, όπου $\varepsilon > 0$ μία σταθερά, και
- αν $x \notin L$, τότε $\mathbb{P}\mathrm{r}[N(x) = \text{NO}] = 1$.

Με άλλα λόγια, αν $N(x) = \text{YES}$, είμαστε σίγουροι ότι $x \in L$. Αντίθετα, η απάντηση $N(x) = \text{NO}$ είναι επισφαλής, αφού προκύπτει με δύο τρόπους: (με βεβαιότητα) όταν $x \notin L$, και (με σχετικά μικρή πιθανότητα) όταν $x \in L$. Η συμπληρωματική κλάση της **RP**, η **coRP**, ορίζεται κατά τον ίδιο τρόπο, μόνο που πιθανότητα σφάλματος υπάρχει μόνο όταν $x \notin L$. Για την κλάση **coRP**, όταν η απάντηση είναι NO, είμαστε σίγουροι ότι είναι σωστή.

Από τον ορισμό της κλάσης **RP**, έχουμε ότι $\mathbf{RP} \subseteq \mathbf{NP}$, $\mathbf{RP} \subseteq \mathbf{BPP}$ και $\mathbf{coRP} \subseteq \mathbf{BPP}$, αλλά δεν γνωρίζουμε αν $\mathbf{RP} = \mathbf{coRP}$. Επίσης δεν είναι γνωστή η σχέση των **BPP** και **NP**, αλλά γνωρίζουμε ότι αν $\mathbf{NP} \subseteq \mathbf{BPP}$, τότε $\mathbf{NP} = \mathbf{RP}$ (με το τελευταίο να θεωρείται απίθανο).

Τέλος ορίζουμε μια κλάση που αφορά σε πιθανοτικούς αλγόριθμους που, όταν απαντούν, δεν κάνουν λάθος. Στην κλάση **ZPP** (από τα αρχικά του Zero-error Probabilistic Polynomial) εντάσσονται οι γλώσσες L για τις οποίες υπάρχει μια NTM N πολυωνυμικού χρόνου τέτοια ώστε για κάθε είσοδο x:

[7] Στην πραγματικότητα δεν ενδιαφέρει ο ακριβής ορισμός της ενισχυμένης πλειοψηφίας που απαιτείται για την απάντηση (δηλαδή το ακριβές μέγεθος του $\varepsilon > 0$), αρκεί αυτή να είναι σημαντική. Έτσι αντί για $1/2 + \varepsilon$, μπορούμε να ορίσουμε την πιθανότητα σωστής απάντησης ως $1/2 + 1/q(|x|)$, για κάποιο πολυώνυμο $q(|x|) \geq 2$. Ο αναγνώστης μπορεί να διαπιστώσει ότι σε κάθε περίπτωση, με πολυωνυμικό αριθμό επαναλήψεων του ίδιου αλγόριθμου, μπορούμε να ενισχύσουμε την πιθανότητα σωστής απάντησης σε $1 - 2^{-q(|x|)}$, για οποιοδήποτε πολυώνυμο $q(|x|) \geq 2$.

– Αν $x \in L$, τότε $\mathbb{P}\mathrm{r}[N(x) = \text{NO}] = 0$,
– αν $x \notin L$, τότε $\mathbb{P}\mathrm{r}[N(x) = \text{YES}] = 0$, και
– υπάρχει σταθερά $\varepsilon > 0$, τέτοια ώστε $\mathbb{P}\mathrm{r}[M(x) = \text{UNK}] < \varepsilon$.

Ισοδύναμα, μπορούμε να ορίσουμε την **ZPP** ως την τομή των κλάσεων **RP** και **coRP**, δηλ. $\mathbf{ZPP} = \mathbf{RP} \cap \mathbf{coRP}$. Αποδεικνύεται ότι ένα πρόβλημα ανήκει στην κλάση **ZPP** αν υπάρχει πιθανοτικός αλγόριθμος με πολυωνυμικό *αναμενόμενο* χρόνο εκτέλεσης που δίνει πάντοτε σωστή απάντηση.

Παρατήρηση 1. Δεδομένου ότι υπάρχουν πιθανοτικοί αλγόριθμοι ευρείας χρήσης για πολλά πρακτικά προβλήματα, πολλοί διευρύνουν την κλάση των ευεπίλυτων προβλημάτων, ταυτίζοντας την με τις πιθανοτικές κλάσεις πολυπλοκότητας **BPP** και **ZPP**. Να σημειώσουμε επίσης ότι σε αντίθεση με τις κλάσεις **P**, **NP**, και **PSPACE**, που όλες έχουν ενδιαφέροντα πλήρη προβλήματα, δεν γνωρίζουμε αν οι πιθανοτικές κλάσεις **BPP**, **ZPP**, και **RP** έχουν πλήρη προβλήματα.

10.5 Πλειοψηφικοί ποσοδείκτες

Σε αντιστοιχία με τις κλάσεις **P**, **NP**, και **coNP**, θέλουμε να κωδικοποιήσουμε τις πιθανοτικές κλάσεις της προηγούμενης ενότητας με χρήση ποσοδεικτών. Βέβαια ο υπαρξιακός και ο καθολικός ποσοδείκτης δεν αρκούν για αυτόν τον σκοπό, έτσι ορίζουμε τον *ποσοδείκτη ενισχυμένης πλειοψηφίας* $\exists^+$, και τον *ποσοδείκτη (απλής) πλειοψηφίας* $\exists^{\frac{1}{2}}$ (για άλλες ενδιαφέρουσες εφαρμογές των πλειοψηφικών ποσοδεικτών, βλ. [24]).

Ορισμός 5. *Έστω $|\Sigma| = k$ το μέγεθος του αλφαβήτου, $R \subseteq \Sigma^* \times \Sigma^*$ μία πολυωνυμικά αποκρίσιμη και πολυωνυμικά ισορροπημένη διμελής σχέση, $x \in \Sigma^*$ μια συμβολοσειρά, και p το πολυώνυμο που σχετίζεται με την R.*

– *Συμβολίζουμε με $\exists^+ yR(x, y)$ το γεγονός ότι υπάρχει μια σταθερά $\varepsilon \in (0, 1/2)$, τέτοια ώστε για τουλάχιστον $(1/2 + \varepsilon)k^{p(|x|)}$ από τις συμβολοσειρές $y \in \Sigma^*$ με μήκος $p(|x|)$, ισχύει ότι $(x, y) \in R$.*
– *Συμβολίζουμε με $\exists^{\frac{1}{2}} yR(x, y)$ το γεγονός ότι για περισσότερες από $k^{p(|x|)}/2$ συμβολοσειρές $y \in \Sigma^*$ μήκους $p(|x|)$, ισχύει ότι $(x, y) \in R$.*

Αυτό που κάναμε ουσιαστικά ήταν να εκφράσουμε τις πιθανότητες με τη βοήθεια των πλειοψηφικών ποσοδεικτών. Με χρήση τώρα των ποσοδεικτών απλής και ενισχυμένης πλειοψηφίας, μπορούμε να εκφράσουμε τις πιθανότητες επιτυχίας στους ορισμούς των κλάσεων **BPP**, **PP**, **RP**, και **coRP**, και να τις κωδικοποιήσουμε έτσι με βάση αυτούς τους ποσοδείκτες. Συγκεκριμένα, η κλάση **BPP** ορίζεται ως:

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

όπου το $R \in \mathbf{P}$ δηλώνει ότι η $R \subseteq \Sigma^* \times \Sigma^*$ είναι μια πολυωνυμικά αποκρίσιμη και πολυωνυμικά ισορροπημένη διμελής σχέση. Πιο σύντομα μπορούμε να γράψουμε $\mathbf{BPP} = (\exists^+, \exists^+)$.

Για την κλάση **PP** έχουμε:

$$L \in \mathbf{PP} \iff \exists R \in \mathbf{P} : \begin{cases} x \in L \Rightarrow \exists^{\frac{1}{2}} yR(x, y) \\ x \notin L \Rightarrow \exists^{\frac{1}{2}} y \neg R(x, y) \end{cases}$$

ή πιο σύντομα $\mathbf{PP} = (\exists^{\frac{1}{2}}, \exists^{\frac{1}{2}})$.

Για την κλάση $\mathbf{RP}$ έχουμε:

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

ή πιο σύντομα $\mathbf{RP} = (\exists^{+}, \forall)$, ενώ για την συμπληρωματική κλάση $\mathbf{coRP}$ έχουμε:

$$L \in \mathbf{coRP} \iff \exists R \in \mathbf{P} : \begin{cases} x \in L \Rightarrow \forall y R(x,y) \\ x \notin L \Rightarrow \exists^{+} y \neg R(x,y) \end{cases}$$

ή πιο σύντομα $\mathbf{coRP} = (\forall, \exists^{+})$.

10.6 Διαλογικά Συστήματα Απόδειξης

Θα ολοκληρώσουμε με ένα παράδειγμα κλάσης πολυπλοκότητας που ορίζεται με τη βοήθεια δύο TM, που αλληλεπιδρούν ανταλλάσσοντας μηνύματα μεταξύ τους. Η πρώτη, που καλούμε *αποδείκτη* (prover), προσπαθεί να αποδείξει στη δεύτερη, που καλούμε *επαληθευτή* (verifier), ότι μια πρόταση της μορφής $x \in L$ είναι αληθής.

Ο αποδείκτης είναι παντοδύναμος, με την έννοια ότι λειτουργεί χωρίς κανένα περιορισμό στους υπολογιστικούς του πόρους. Αντίθετα, ο επαληθευτής είναι ένας πιθανοτικός αλγόριθμος πολυωνυμικού χρόνου. Ο επαληθευτής και ο αποδείκτης συμμετέχουν σε ένα πρωτόκολλο επικοινωνίας ανταλλάσσοντας μηνύματα. Ανάλογα με τα μηνύματα που λαμβάνει από τον αποδείκτη, ο επαληθευτής μπορεί να δεχθεί ή να απορρίψει την απόδειξη. Ο αποδείκτης μπορεί να μην είναι έντιμος, και να θέλει να πείσει τον επαληθευτή ότι $x \in L$, ακόμη και όταν $x \notin L$. Ο επαληθευτής, απέναντι στον παντοδύναμο αποδείκτη, μπορεί να χρησιμοποιήσει, εκτός του πολυωνυμικού υπολογιστικού χρόνου, την τυχαιότητα που διαθέτει. Η κλάση $\mathbf{IP}$ (από τα αρχικά του Interactive Proof) ορίστηκε από τους S. Goldwasser, S. Micali, και C. Rackoff [9] το 1985. Η κλάση $\mathbf{IP}$ περιλαμβάνει τις γλώσσες L για τις οποίες:

- Όταν $x \in L$, υπάρχει αποδείκτης, ώστε ο επαληθευτής να αποδέχεται με πιθανότητα τουλάχιστον $1/2 + \varepsilon$, όπου $\varepsilon \in (0, 1/2]$ μία σταθερά, και
- όταν $x \notin L$, για κάθε αποδείκτη, ο επαληθευτής αποδέχεται με πιθανότητα μικρότερη ή ίση του ε', όπου $\varepsilon' \in [0, 1/2)$ μία σταθερά.

Στον παραπάνω ορισμό, οι πιθανότητες εξαρτώνται μόνο από τις τυχαίες επιλογές που χρησιμοποιεί ο επαληθευτής, και τις οποίες κρατά μυστικές από τον αποδείκτη.

Παράδειγμα 2. Ας θεωρήσουμε το πρόβλημα του *Μη Ισομορφισμού Γραφημάτων* (Graph Non-Isomorphism), όπου δίνονται δύο γραφήματα G_1 και G_2, και πρέπει να αποφασίσουμε αν *δεν* είναι ισομορφικοί. Αυτό το πρόβλημα ανήκει στο $\mathbf{coNP}$, αφού το συμπληρωματικό πρόβλημα, αυτό του *Ισομορφισμού Γραφημάτων* (Graph Isomorphism), ανήκει στο $\mathbf{NP}$ [8]. Στη συνέχεια περιγράφουμε ένα διαλογικό σύστημα απόδειξης, και αποδεικνύουμε ότι ο Μη Ισομορφισμός Γραφημάτων ανήκει στο $\mathbf{IP}$.

Στο διαλογικό σύστημα απόδειξης, ο επαληθευτής επιλέγει τυχαία ένα από τα δύο γραφήματα, έστω το γράφημα G_i, $i \in \{1, 2\}$, και υπολογίζει ένα τυχαίο γράφημα H ισομορφικό του G_i (αυτό

[8] Δεν είναι γνωστό αν το πρόβλημα του Ισομορφισμού Γραφημάτων ανήκει στο $\mathbf{P}$ ή είναι $\mathbf{NP}$-πλήρες. Μάλιστα υπάρχουν ισχυρές ενδείξεις ότι δεν είναι $\mathbf{NP}$-πλήρες, βλ. [3]. Έτσι αποτελεί υποψήφιο $\mathbf{NP}$-ενδιάμεσο πρόβλημα.

γίνεται με την επιλογή μιας τυχαίας μετάθεσης των κορυφών του G_i). Ο επαληθευτής στέλνει το γράφημα H στον αποδείκτη, και του ζητά ένα $j \in \{1,2\}$ ώστε το γράφημα G_j να είναι ισομορφικό του H. Ο επαληθευτής αποδέχεται (δηλ. καταλήγει στο συμπέρασμα ότι τα G_1 και G_2 δεν είναι ισομορφικά) αν $i = j$ (δηλ. αν η απάντηση του αποδείκτη ταυτίζεται με την επιλογή του επαληθευτή), και απορρίπτει (δηλ. καταλήγει ότι τα G_1 και G_2 είναι ισομορφικά) διαφορετικά.

Αν τα G_1 και G_2 δεν είναι ισομορφικά, τότε ο αποδείκτης (ως παντοδύναμος), βρίσκει το ισομορφικό γράφημα του H, και απαντά την τιμή $j = i$, οδηγώντας τον επαληθευτή στην αποδοχή (με βεβαιότητα). Αν τα G_1 και G_2 είναι ισομορφικά, ο αποδείκτης δεν μπορεί να συμπεράνει από ποιο γράφημα προήλθε το H. Έτσι απαντά μια οποιαδήποτε τιμή $j \in \{1,2\}$, η οποία οδηγεί τον επαληθευτή στην απόρριψη με πιθανότητα $1/2$. □

Παρατηρούμε ότι η κλάση **IP** περιλαμβάνει τις κλάσεις **NP** και **BPP**, αλλά και προβλήματα που πιστεύουμε ότι δεν ανήκουν σε αυτές (π.χ. το πρόβλημα του Μη Ισομορφισμού Γραφημάτων). Ειδικότερα, το **NP** περιλαμβάνει τις γλώσσες του **IP** που ο επαληθευτής αναγνωρίζει χωρίς να καταφύγει σε τυχαίες επιλογές, και το **BPP** περιλαμβάνει τις γλώσσες του **IP** που ο επαληθευτής αναγνωρίζει μόνο με τις δικές του δυνάμεις (δηλαδή χωρίς να λαμβάνει υπόψη του τις απαντήσεις του αποδείκτη). Μάλιστα, σε ένα αξιοσημείωτο αποτέλεσμα, ο A. Shamir απέδειξε ότι στην κλάση **IP** εντάσσονται όλες οι γλώσσες με πολυωνυμική χωρική πολυπλοκότητα και μόνον αυτές [21]. Δηλαδή,

Θεώρημα 13 (Shamir). $\mathbf{IP} = \mathbf{PSPACE}$.

Βιβλιογραφία

1. M. Agrawal, N. Kayal, and N. Saxena PRIMES is in **P**. *Annals of Mathematics* vol. **160** (2), pp. 781−793, 2004.
2. S. Arora and B. Barak. *Computational Complexity: A Modern Approach*. Cambridge University Press, 2009.
3. R.B. Boppana, J. Hastad, and S. Zachos: Does **coNP** Have Short Interactive Proofs? *Information Processing Letters*, vol. **25** (2), pp. 127−132, 1987.
4. S. Cook. The Complexity of Theorem Proving Procedures. In *Proc. of the 3th ACM Symp. on Theory of Computing (STOC '71)*, pp. 151−158, 1971.
5. M. Davis, R. Sigal, and E.J. Weyuker. *Computability, Complexity, and Languages (2nd ed.)*. Morgan Kauffman, 1994.
6. M. Furer. The Tight Deterministic Time Hierarchy. In *Proc. of the 14th ACM Symp. on Theory of Computing (STOC '82)*, pp. 8−16, 1982.
7. M.R. Garey and D.S. Johnson. *Computers and Intractability: A Guide to the Theory of* **NP***-completeness*. W.H. Freeman and Company, 1979.
8. O. Goldreich. *Computational Complexity: A Conceptual Perspective*. Cambridge University Press, 2009.
9. S. Goldwasser, S. Micali, and C. Rackoff. The Knowledge Complexity of Interactive Proof Systems. *SIAM Journal on Computing*, vol. **18**, pp. 186−208, 1989.
10. J. Grollman and A.L. Selman. Complexity Measures for Public-Key Cryptography. *SIAM Journal on Computing*, vol. **17**, pp. 309−335, 1988.
11. J. Hartmanis, P.M. Lewis II, R.E. Stearns. Hierarchies of Memory Limited Computations. In *Proc. of 6th IEEE Symp. on Switching Circuit Theory and Logic Design*, pp. 179–190, 1965.
12. J.E. Hopcroft, R. Motwani, and J.D. Ullman. *Introduction to Automata Theory, Languages, and Computation (3rd ed.)*. Addison-Wesley, 2007.
13. N. Immerman. Nondeterministic Space is Closed under Complementation. *SIAM Journal on Computing*, vol. **17**, pp. 935−938, 1988.
14. L.G. Khachian. A Polynomial Time Algorithm for Linear Programming. *Doklady Akad. Nauk USSR*, vol. **244** (5), pp. 1093−1096, 1979.
15. R.E. Ladner. On the Structure of Polynomial Time Reducibility. *Journal of the ACM*, vol. **22** (1), pp. 155−171, 1975.
16. H.R. Lewis and C.H. Papadimitriou. *Elements of the Theory of Computation (2nd ed.)*. Prentice-Hall, 1998.
17. C.H. Papadimitriou. *Computational Complexity*. Addison-Wesley, 1994.

18. W.J. Paul, N. Pippenger, E. Szemeredi, and W.T. Trotter. On Determinism versus Non-Determinism and Related Problems. In *Proc. of the 24th IEEE Symp. on Foundations of Computer Science (FOCS '83)*, pp. 429−438, 1983.
19. W.J. Savitch. Relationship between Nondeterministic and Deterministic Tape Classes. *Journal of Computer and System Sciences*, vol. **4**, pp. 177−192, 1983.
20. J.I. Seiferas, M.J. Fischer, and A.R. Meyer. Refinements of Nondeterministic Time and Space Hierarchies. In *Proc. of the 14th IEEE Symp. on Foundations of Computer Science (FOCS '73)*, pp. 130−137, 1973.
21. A. Shamir. **IP** = **PSPACE**. In *Proc. of the 31th IEEE Symp. on Foundations of Computer Science (FOCS '90)*, pp. 11−15, 1990.
22. M. Sipser. *Introduction to the Theory of Computation (2nd ed.)*. Course Technology, 2005.
23. R. Szelepcsenyi. The Method of Forcing for Nondeterministic Automata. *Bulletin of the EATCS*, vol. **33**, pp. 96−100, 1987.
24. S. Zachos. Probabilistic Quantifiers and Games. *Journal of Computer and System Sciences*, vol. **36**, pp. 433−451, 1983.
25. Ε. Ζάχος. *Υπολογισιμότητα και Πολυπλοκότητα*. Εθνικό Μετσόβιο Πολυτεχνείο, 2008.

11 Ανοικτά Ερευνητικά Προβλήματα

Η Θεωρία Υπολογιστικής Πολυπλοκότητας είναι μια ιδαίτερα ενεργή και εξαιρετικά ενδιαφέρουσα ερευνητική περιοχή της Θεωρητικής Επιστήμης των Υπολογιστών, με πολλά και σημαντικά ανοικτά ερωτήματα. Σε αυτή την ενότητα, θα περιοριστούμε στα ανοικτά ερωτήματα που αφορούν έννοιες και κλάσεις πολυπλοκότητας που παρουσιάστηκαν σε αυτό το κεφάλαιο.

Το σημαντικότερο ανοικτό ερώτημα στη Θεωρητική Επιστήμη των Υπολογιστών αφορά στον γνήσιο εγκλεισμό της κλάσης **P** στην κλάση **NP**. Η σχέση των κλάσεων **P** και **NP** έχει πολλές και σημαντικές (θεωρητικές και πρακτικές) συνέπειες. Το ερώτημα για τη σχέση των κλάσεων **P** και **NP** έχει επιλεγεί από το Clay Mathematics Institute ως ένα από τα επτά σημαντικότερα ανοικτά προβλήματα στα Μαθηματικά. Για την επίλυση καθενός από αυτά τα προβλήματα, το ινστιτούτο έχει θεσπίσει βραβείο ενός εκατομμυρίου δολαρίων.

Ένα γενικότερο ανοικτό ερώτημα είναι η γνησιότητα των εγκλεισμών για τις κλάσεις ντετερμινιστικής και μη ντετερμινιστικής χρονικής και χωρικής πολυπλοκότητας στην σχέση (1). Γνωρίζουμε ότι **L** ⊂ **PSPACE** και ότι **NL** ⊂ **PSPACE**, ως άμεση συνέπεια του Θεωρήματος 10 και του Θεωρήματος 11 αντίστοιχα. Άρα τουλάχιστον ένας από τους εγκλεισμούς στην (1) είναι γνήσιος. Αποτελεί κοινή πεποίθηση ότι όλοι οι εγκλεισμοί στην (1) είναι γνήσιοι, αλλά μέχρι στιγμής κάτι τέτοιο δεν έχει αποδειχθεί.

Ένα σημαντικό ανοικτό πρόβλημα που σχετίζεται ευθέως με την κρυπτογραφία αφορά στην σχέση των κλάσεων **P** και **UP**, οι οποίες ταυτίζονται αν και μόνο αν υπάρχουν συναρτήσεις μονής κατεύθυνσης (βλ. Θεώρημα 6).

Άλλα σημαντικά ανοικτά ερωτήματα αφορούν στην κλειστότητα των μη ντετερμινιστικών κλάσεων χρονικής πολυπλοκότητας ως προς το συμπλήρωμα, με ιδιαίτερο ενδιαφέρον για τη σχέση των κλάσεων **NP** και **coNP**, και στη σχέση των πιθανοτικών κλάσεων πολυπλοκότητας **BPP** και **ZPP** με την κλάση **NP**.

12 Ασκήσεις

Άσκηση 1. Να αποδείξετε ότι η κλάση **P** είναι κλειστή ως προς την ένωση, την τομή, το συμπλήρωμα, και την παράθεση (concatenation)[9] Δηλαδή να δείξετε ότι αν δύο γλώσσες L_1 και L_2 ανήκουν στο **P**, τότε και οι γλώσσες $L_1 \cap L_2$, $L_1 \cup L_2$, $\overline{L}_1$, και $L_1 L_2$ ανήκουν στο **P**.

[9] Υπενθυμίζεται ότι η παράθεση δύο γλωσσών L_1 και L_2, που ορίζονται σε ένα αλφάβητο Σ, είναι μια γλώσσα η οποία συμβολίζεται με $L_1 L_2$, και αποτελείται από κάθε συμβολοσειρά που προκύπτει από παράθεση μιας συμβολοσειράς

Άσκηση 2. Να αποδείξετε το Θεώρημα 4, δηλαδή να δείξετε ότι κάθε πρόβλημα που υπολογίζεται από μια NTM $t(n)$-χρόνου, μπορεί να υπολογιστεί από μια TM $O(c^{t(n)})$-χρόνου, για κάποια σταθερά $c > 1$.

Άσκηση 3. Να αποδείξετε ότι το πρόβλημα του Πλανόδιου Πωλητή (στη διατύπωσή του ως πρόβλημα απόφασης) ανήκει στην κλάση **NP**.

Άσκηση 4. Να αποδείξετε ότι το πρόβλημα της Ικανοποιησιμότητας μιας λογικής πρότασης σε Συζευκτική Κανονική Μορφή ανήκει στην κλάση **NP**.

Άσκηση 5. Να δείξετε ότι η κλάση **NP** είναι κλειστή ως προς την ένωση, την τομή, και την παράθεση. Δηλαδή να δείξετε ότι αν δύο γλώσσες $L_1, L_2 \in \mathbf{NP}$, τότε και οι γλώσσες $L_1 \cup L_2, L_1 \cap L_2$, και $L_1 L_2$ ανήκουν στο **NP**.

Άσκηση 6. Να αποδείξετε ότι ότι η πολυωνυμική αναγωγή είναι μεταβατική. Δηλαδή να δείξετε ότι αν το Π_1 ανάγεται πολυωνυμικά στο Π_2, και το Π_2 ανάγεται πολυωνυμικά στο Π_3, τότε το Π_1 ανάγεται πολυωνυμικά στο Π_3.

Άσκηση 7. Να αποδείξετε ότι η κλάση **P** είναι κλειστή ως προς την πολυωνυμική αναγωγή. Δηλαδή να δείξετε ότι αν το Π_1 ανάγεται πολυωνυμικά στο Π_2 και $\Pi_2 \in \mathbf{P}$, τότε και $\Pi_1 \in \mathbf{P}$.

Άσκηση 8. Να δείξετε ότι η κλάση **NP** είναι κλειστή ως προς την πολυωνυμική αναγωγή. Δηλαδή να δείξετε ότι αν το πρόβλημα Π_1 ανάγεται πολυωνυμικά στο πρόβλημα Π_2 και $\Pi_2 \in \mathbf{NP}$, τότε και $\Pi_1 \in \mathbf{NP}$.

Άσκηση 9. Να αποδείξετε ότι αν δύο κλάσεις **C** και **C**′ είναι κλειστές ως προς την αναγωγή $\leq$, και υπάρχει ένα πρόβλημα Π που είναι πλήρες τόσο για τη **C** όσο και για τη **C**′ ως προς την αναγωγή $\leq$, τότε $\mathbf{C} = \mathbf{C}'$.

Άσκηση 10. Στο πρόβλημα της Μέγιστης 2-Ικανοποιησιμότητας (Maximum 2-Satisfiability) δίνεται ένα σύνολο όρων φ αποτελούμενων από διάζευξη δύο ατόμων, και ένας φυσικός αριθμός $B > 0$. Το ζητούμενο είναι αν υπάρχει αποτίμηση που ικανοποιεί τουλάχιστον B όρους του φ. Να αποδείξετε ότι το πρόβλημα της Μέγιστης 2-Ικανοποιησιμότητας είναι **NP**-πλήρες.

Άσκηση 11. Ένα *σύνολο ανεξαρτησίας* (independent set) ενός γραφήματος είναι ένα σύνολο κορυφών χωρίς ακμές μεταξύ τους. Στο πρόβλημα του *Συνόλου Ανεξαρτησίας* (Independent Set), δίνεται ένα γράφημα $G(V, E)$ και ένας φυσικός αριθμός $B > 0$. Το ζητούμενο είναι αν το G περιέχει σύνολο ανεξαρτησίας με τουλάχιστον B κορυφές. Να αποδείξετε ότι το Σύνολο Ανεξαρτησίας είναι **NP**-πλήρες.

Άσκηση 12. Ένα σύνολο κορυφών C ενός γραφήματος αποτελεί *κάλυμμα* (vertex cover) όταν κάθε ακμή έχει τουλάχιστον ένα άκρο στο C. Στο πρόβλημα του *Καλύμματος Κορυφών* (Vertex Cover), δίνεται ένα γράφημα $G(V, E)$ και ένας φυσικός αριθμός $B > 0$. Το ζητούμενο είναι αν το G περιέχει κάλυμμα κορυφών με το πολύ B κορυφές. Να αποδείξετε ότι το Κάλλυμα Κορυφών είναι **NP**-πλήρες.

Άσκηση 13. Να αποδείξετε ότι $\mathbf{NP} \subseteq \mathbf{PP}$. Ειδικότερα, δεδομένης μιας NTM N πολυωνυμικού χρόνου που αποφασίζει μια γλώσσα L, να κατασκευάσετε μια NTM N' πολυωνυμικού χρόνου τέτοια ώστε για κάθε είσοδο x, $N(x) = \text{NAI}$ αν και μόνο αν $\Pr[N'(x) = \text{NAI}] > 1/2$.

της L_1 και μιας συμβολοσειράς της L_2. Τυπικά,

$$L_1 L_2 = \{w \in \Sigma^* : w = w_1 w_2 \text{ για κάποιες συμβολοσειρές } w_1 \in L_1 \text{ και } w_2 \in L_2\}$$

13 Αντιστοίχηση Ελληνικών - Αγγλικών Όρων

C-δύσκολο πρόβλημα	**C**-hard problem
C-πλήρες πρόβλημα	**C**-complete problem
NP-ενδιάμεσο πρόβλημα	**NP**-intermediate problem
Αναγωγή	Reduction
Αναγωγή λογαριθμικού χώρου	Log-space reduction
Αποδείκτης	Prover
Αποκρίσιμη γλώσσα	Decidable language
Άτομο (λογικής πρότασης)	Literal
Βέλτιστη λύση	Optimal solution
Δέντρο υπολογισμού	Computation tree
Διαλογικό σύστημα απόδειξης	Interactive proof system
Δυσεπίλυτο πρόβλημα	Intractable problem
Επαληθευτής	Verifier
Ευεπίλυτο πρόβλημα	Tractable problem
Εφικτή λύση	Feasible solution
Ικανοποιησιμότητας (πρόβλημα)	Satisfiability Problem, SAT
Ισομορφισμός γραφημάτων	Graph Isomorphism
Καθολική Μηχανή Turing	Universal Turing Machine
Κλάση Πολυπλοκότητας	Complexity Class
Κωδικοποίηση	Encoding
Μη αποκρίσιμη γλώσσα	Undecidable language
Μη ντετερμινιστική Μηχανή Turing	Nondeterministic Turing Machine
Μηχανή Turing	Turing Machine
Πιθανοτικός αλγόριθμος	Randomized algorithm
Πληρότητα	Completeness
Πολυωνυμικά αποκρίσιμη σχέση	Polynomially decidable relation
Πολυωνυμικά ισορροπημένη σχέση	Polynomially balanced relation
Πολυωνυμική αναγωγή	Polynomial reduction
Πρόβλημα απόφασης	Decision problem
Πρόβλημα βελτιστοποίησης	Optimization problem
Πρόβλημα Πλανόδιου Πωλητή	Traveling Salesperson Problem
Πρόβλημα Συντομότερου Μονοπατιού	Shortest Path Problem
Πρόβλημα Τερματισμού	Halting Problem
Συζευκτική Κανονική Μορφή	Conjunctive Normal Form, CNF
Συμβολοσειρά	String
Συμπλήρωμα	Complement
Συναρτήσεις μονής κατεύθυνσης	One-way functions
Συνάρτηση μετάβασης	Transition function
Τυπική γλώσσα	Formal language
Υπολογιστή ή υπολογίσιμη γλώσσα	Computable language
Υπολογιστική Πολυπλοκότητα	Computational Complexity
Υπολογιστότητα ή υπολογισιμότητα	Computability
Φράση (λογικής πρότασης)	Clause
Χρονική πολυπλοκότητα	Time complexity
Χωρική πολυπλοκότητα	Space complexity