Εθνικο Μετσοβιο Πολυτεχνειο
Τμημα Ηλεκτρολογων Μηχανικων και Μηχανικων Υπολογιστων

ΤΟΜΕΑΣ ΤΕΧΝΟΛΟΓΙΑΣ ΠΛΗΡΟΦΟΡΙΚΗΣ ΚΑΙ ΥΠΟΛΟΓΙΣΤΩΝ
ΕΡΓΑΣΤΗΡΙΟ ΛΟΓΙΚΗΣ ΚΑΙ ΕΠΙΣΤΗΜΗΣ ΥΠΟΛΟΓΙΣΜΩΝ (CoReLab)

ΑΛΓΟΡΙΘΜΙΚΕΣ ΤΕΧΝΙΚΕΣ ΣΤΗ ΘΕΩΡΙΑ ΠΟΛΥΠΛΟΚΟΤΗΤΑΣ

ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ

του

Ευάγγελου Γ. Μπαμπά

Επιβλέπων: Στάθης Ζάχος
Καθηγητής ΕΜΠ

Αθήνα, Οκτώβριος 2004

Εθνικο Μετσοβιο Πολυτεχνειο
ΤΜΗΜΑ ΗΛΕΚΤΡΟΛΟΓΩΝ ΜΗΧΑΝΙΚΩΝ ΚΑΙ ΜΗΧΑΝΙΚΩΝ ΥΠΟΛΟΓΙΣΤΩΝ
ΤΟΜΕΑΣ ΤΕΧΝΟΛΟΓΙΑΣ ΠΛΗΡΟΦΟΡΙΚΗΣ ΚΑΙ ΥΠΟΛΟΓΙΣΤΩΝ
ΕΡΓΑΣΤΗΡΙΟ ΛΟΓΙΚΗΣ ΚΑΙ ΕΠΙΣΤΗΜΗΣ ΥΠΟΛΟΓΙΣΜΩΝ (CoReLab)

ΑΛΓΟΡΙΘΜΙΚΕΣ ΤΕΧΝΙΚΕΣ ΣΤΗ ΘΕΩΡΙΑ ΠΟΛΥΠΛΟΚΟΤΗΤΑΣ

ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ

του

Ευάγγελου Γ. Μπαμπά

Επιβλέπων: Στάθης Ζάχος
Καθηγητής ΕΜΠ

Εγκρίθηκε από την τριμελή εξεταστική επιτροπή την 4-10-2004.

....	....	....
Στάθης Ζάχος	Φώτω Αφράτη	Νικόλαος Παπασπύρου
Καθηγητής ΕΜΠ	Καθηγήτρια ΕΜΠ	Λέκτορας ΕΜΠ

Αθήνα, Οκτώβριος 2004.

....
Ευάγγελος Γ. Μπαμπάς
Διπλωματούχος Ηλεκτρολόγος Μηχανικός και Μηχανικός Υπολογιστών ΕΜΠ

// Περίληψη

Σκοπός της παρούσας εργασίας είναι η μελέτη και παρουσίαση τεχνικών που έχουν χρησιμοποιηθεί για την απόδειξη αποτελεσμάτων στο αντικείμενο της θεωρίας πολυπλοκότητας. Οι τεχνικές αυτές είναι κατά βάση αλγοριθμικές, αφού το υποκείμενο υπολογιστικό μοντέλο είναι πάντα η μηχανή Turing.

Αυτο-αναγωγή Διερευνάται η δυσκολία των αραιών και μονοσυμβολικών γλωσσών ως προς τις κλάσεις **NP** και **coNP**. Το βασικό αποτέλεσμα είναι ότι αν μία αραιή γλώσσα είναι **NP**-$\leq^{\mathrm{p}}_{\mathrm{btt}}$-δύσκολη, τότε $\mathbf{P} = \mathbf{NP}$.

Συναρτήσεις μονής κατεύθυνσης Εξετάζεται η τεχνική των συναρτήσεων μονής κατεύθυνσης. Από την εφαρμογή της μεθόδου προκύπτει ότι αν υπάρχουν συναρτήσεις μονής κατεύθυνσης, τότε υπάρχουν συναρτήσεις μονής κατεύθυνσης που έχουν αυξημένη ασφάλεια και ορισμένες επιθυμητές αλγεβρικές ιδιότητες.

Τουρνουά και ημιαποδοτικοί υπολογισμοί Χρησιμοποιώντας δύο αποτελέσματα από τη θεωρία γραφημάτων, αποδεικνύεται ότι τα ημιαποδοτικώς υπολογιστά σύνολα μπορούν να αποφασιστούν με μικρό μήκος συμβουλής. Το κυρίως αποτέλεσμα του κεφαλαίου είναι ότι αν μπορούμε να απομονώσουμε μία τιμή μίας μη ντετερμινιστικής συνάρτησης, τότε η πολυωνυμική ιεραρχία καταρρέει στο δεύτερο επίπεδο.

Ελάττωση μαρτύρων Παρουσιάζεται μία θεωρία για τις ιδιότητες κλειστότητας της κλάσης $\#\mathbf{P}$, η οποία έχει ομοιότητες με τη θεωρία της **NP**-πληρότητας. Προκύπτει τελικά ότι είναι εξαιρετικά απίθανο η $\#\mathbf{P}$ να είναι κλειστή ως προς αφαίρεση ή διαίρεση.

Λέξεις κλειδιά: αυτο-αναγωγή, αραιά σύνολα, ψαλίδισμα δέντρου αυτο-αναγωγής, αριστερό σύνολο, συναρτήσεις μονής κατεύθυνσης, ισχυρή μη αντιστρεψιμότητα, ημιαποδοτικώς υπολογιστά σύνολα, μη ντετερμινιστικές συναρτήσεις, ελάττωση μαρτύρων, πλήρεις ιδιότητες κλειστότητας.

Abstract

The aim of this thesis is to study and present certain techniques that have been used in the proofs of complexity-theoretic results. Those techniques are inherently algorithmic, since the underlying computational model is always the Turing machine.

Self-reducibility The hardness of sparse and tally languages for the classes **NP** and **coNP** is investigated. The main result is that if a sparse language is **NP**-$\leq^{\mathrm{p}}_{\mathrm{btt}}$-hard, then $\mathbf{P} = \mathbf{NP}$.

One-way functions We survey the one-way function technique. The application of this method yields the result that if there exist one-way functions, then there exist one-way functions which possess increased security and certain desired algebraic properties.

Tournaments and semi-feasible computation Two graph-theoretic results allow us to prove that semi-feasible sets can be decided with small length of advice. The central result of this chapter is that if we can isolate a single value of a non-deterministic function then the polynomial hierarchy collapses at the second level.

Witness reduction A theory for the closure properties of $\#\mathbf{P}$ is put forth, which bears some resemblance to the theory of **NP**-completeness. It is proved highly improbable that $\#\mathbf{P}$ is closed under proper subtraction or integer division.

Keywords: self-reducibility, sparse sets, self-reducibility tree pruning, left set, one-way functions, strong non-invertibility, semi-feasible sets, non-deterministic functions, witness reduction, complete closure properties.

στο Νίκο Αλεξάκη

Ευχαριστίες

Θα ήθελα να ευχαριστήσω την οικογένειά μου για την παιδεία και το γόνιμο περιβάλλον που κατόρθωσε να μου προσφέρει. Τους καθηγητές μου Στάθη Ζάχο για τον ενθουσιασμό που μου μετέδωσε για τη θεωρητική πληροφορική και Νίκο Παπασπύρου για την άψογη συνεργασία και την ανεκτικότητά του, και την καθηγήτριά μου Φώτω Αφράτη για την υποστήριξή της σε όλα τα χρόνια των σπουδών μου. Όλα τα μέλη του Εργαστηρίου Λογικής και Επιστήμης Υπολογισμών για το φιλικότατο κλίμα που έχουν δημιουργήσει στην ομάδα. Όλα τα μέλη του Εργαστηρίου Τεχνολογίας Λογισμικού γιατί με κρατούσαν πάντοτε σε εγρήγορση. Τέλος, τον Ισίδωρο Σιδερή που είχε την υπομονή να εντοπίσει πολλά τπογραφικά λάθη, αλλά και το Χρυσάφη Σταμούδη που παρενέβη σε μία κρίσιμη καμπή της συγγραφής της εργασίας.

Περιεχόμενα

Κατάλογος Αλγορίθμων

Κεφάλαιο 1

Εισαγωγή στη Θεωρία Πολυπλοκότητας

1.1 Η θέση του Church

Στην αρχή του εικοστού αιώνα, ο μαθηματικός David Hilbert έθεσε στη μαθηματική κοινότητα 23 προβλήματα ως πρόκληση για το νέο αιώνα. Το 10ο από αυτά ζητούσε την εύρεση μιας υπολογιστικής διαδικασίας μέσω της οποίας μπορεί να αποφασιστεί με πεπερασμένο αριθμό βημάτων αν ένα δοσμένο πολυώνυμο πολλών μεταβλητών έχει ακέραια ρίζα. Ο τρόπος με τον οποίο έθεσε το πρόβλημα ο Hilbert υπονοούσε ότι ήταν βέβαιο πως μια τέτοια διαδικασία υπάρχει, απλά έπρεπε κάποιος να την ανακαλύψει.

Όπως είναι σήμερα γνωστό από τη δουλειά του Yuri Matijasevič (ο οποίος στηρίχθηκε σε αποτελέσματα των Davis, Putnam, Robinson), δεν υπάρχει αλγόριθμος που να λύνει το συγκεκριμένο πρόβλημα. Οι μαθηματικοί, μέχρι τη δεκαετία του 1930, είχαν μία εντελώς διαισθητική αντίληψη της έννοιας του αλγορίθμου, ως μιας διαδικασίας που περιγράφεται με απλές και σαφώς ορισμένες οδηγίες για την επίλυση ενός προβλήματος. Μία τέτοια αντίληψη ίσως είναι αρκετή για να επιτρέπει σε κάποιον να διατυπώνει και να αναγνωρίζει αλγορίθμους, όμως δεν είναι αρκετά βαθιά ώστε να μπορέσει να στηριχτεί κανείς σε αυτή για να αποδείξει ότι δεν υπάρχει αλγόριθμος για ένα συγκεκριμένο πρόβλημα. Ο Matijasevič το 1970 είχε στα χέρια του έναν τυπικό ορισμό του αλγορίθμου που έδωσαν το 1936 ο Alonzo Church (με βάση τον λ-λογισμό) και ο Alan Turing (με βάση τις "μηχανές" του). Οι δύο ορισμοί αποδείχθηκαν ισοδύναμοι.

Ο λ-λογισμός και οι μηχανές Turing είναι δύο μόνο από τα πολλά υπολογιστικά μοντέλα που μπορούν να χρησιμοποιηθούν για να περιγράψουν έναν αλγόριθμο. Το γεγονός ότι αποδείχθηκαν ισοδύναμα δεν είναι τυχαίο. Άλλα υπολογιστικά μοντέλα που μπορούν να αποδειχθούν ισοδύναμα με τα παραπάνω (άρα και μεταξύ τους) περιλαμβάνουν τις γλώσσες προγραμματισμού PASCAL, C, assembly, τη συνδυαστική λογική, semi-Thue processes, μερικές αναδρομικές συναρτήσεις, σχήματα McCarthy και πολλά άλλα. Το αίτημα ότι όλα τα "λογικά" υπολογιστικά μοντέλα που υπάρχουν ή πρόκειται να υπάρξουν είναι μεταξύ τους μηχανιστικά ισοδύναμα είναι γνωστό ως *θέση του Church*. Τα μοντέλα αυτά είναι ισοδύναμα με την έννοια ότι οι αλγόριθμοι που μπορούν να περιγραφούν με το καθένα από αυτά επιλύουν την ίδια κλάση προβλημάτων. Μάλιστα είναι μηχανιστικά ισοδύναμα, δηλαδή υπάρχει πρόγραμμα (compiler) που δέχεται ως είσοδο την περιγραφή αλγορίθμου σε ένα από αυτά τα μοντέλα και παράγει ως έξοδο τον ισοδύναμο αλγόριθμο σε κάποιο άλλο μοντέλο. Τα "λογικά" μοντέλα όπως αυτά που αναφέραμε έχουν ορισμένα κοινά σημεία, όπως είναι η ντετερμινιστική υπολογιστική ακολουθία σε διακριτά βήματα, πεπερασμένο σύνολο εντολών και πρόσβαση σε απεριόριστη μνήμη.

1.2 Η μηχανή Turing και παραλλαγές

Για να φτάσει στον ορισμό της μηχανής του, ο Turing κινήθηκε αφαιρετικά με βάση τον τρόπο που θα συμπεριφερόταν ένας άνθρωπος αν εκτελούσε με μολύβι και χαρτί τα βήματα ενός ντετερμινιστικού αλγορίθμου. Χωρίς βλάβη της γενικότητας, θεώρησε ότι το χαρτί που χρησιμοποιεί ο άνθρωπος δεν είναι επίπεδο αλλά γραμμικό, όπως η χαρτοταινία σε μια αριθμομηχανή. Οι λειτουργίες που εκτελεί ο άνθρωπος όταν υπολογίζει είναι ουσιαστικά αυτές της παρατήρησης ορισμένων συμβόλων του χαρτιού και της εγγραφής νέων συμβόλων πάνω σε αυτό. Επίσης χωρίς βλάβη της γενικότητας, μπορούμε να θεωρήσουμε ότι ο άνθρωπος παρατηρεί μόνο ένα σύμβολο κάθε φορά, εφόσον οποιαδήποτε πεπερασμένη ομάδα συμβόλων μπορεί να θεωρηθεί ως ένα μοναδικό υπερ-σύμβολο. Το τελευταίο στάδιο της αφαίρεσης είναι να θεωρήσουμε ότι η προσοχή του ανθρώπου μετατοπίζεται από το τρέχον σύμβολο σε κάποιο αμέσως διπλανό σύμβολο πάνω στο γραμμικό χαρτί. Για παράδειγμα, το να κοιτάξει κανείς τρία σύμβολα πιο δεξιά είναι ουσιαστικά ισοδύναμο με το να κοιτάξει ένα σύμβολο δεξιά τρεις φορές διαδοχικά.

Η μηχανή Turing λοιπόν είναι μια μηχανική συσκευή που περιλαμβάνει τα εξής: μια χαρτοταινία που είναι χωρισμένη σε κελιά και μπορεί να προεκτείνεται απεριόριστα και προς τις δύο κατευθύνσεις, και μία κεφαλή που μπορεί να διαβάζει ένα σύμβολο από το τρέχον κελί, να γράφει ένα σύμβολο σε αυτό ή να μετακινείται από ένα κελί σε κάποιο διπλανό (αριστερά ή δεξιά). Η μηχανή βρίσκεται σε κάθε στιγμή σε κάποια εσωτερική κατάσταση, και μπορεί να αλλάζει κατάσταση ανάλογα με το σύμβολο που διαβάζει από το τρέχον κελί. Το σύνολο των δυνατών καταστάσεων είναι πεπερασμένο. Ο συνδυασμός του συμβόλου που διαβάζεται από το τρέχον κελί και της τρέχουσας εσωτερικής κατάστασης της μηχανής καθορίζει την επόμενη ενέργειά της, που μπορεί να είναι μία από τις ακόλουθες (με βάση την αφαίρεση του Turing):

- εγγραφή νέου συμβόλου στο τρέχον κελί
- μετακίνηση κατά ένα κελί αριστερά
- μετακίνηση κατά ένα κελί δεξιά

Μετά από κάθε ενέργεια, η μηχανή μπορεί να αλλάξει ή να διατηρήσει την εσωτερική της κατάσταση.

Συμβολίζουμε τις καταστάσεις της μηχανής Turing με τα στοιχεία ενός πεπερασμένου συνόλου $\mathcal{Q} = \{q_1, q_2, \ldots, q_m\}$ και τα σύμβολα που μπορεί να βρίσκονται στα κελιά της ταινίας με τα στοιχεία ενός επίσης πεπερασμένου συνόλου $\mathcal{S} = \{s_0, s_1, s_2, \ldots, s_r\}$. Το σύμβολο s_0 αναπαριστά το κενό. Το πρόγραμμα μίας μηχανής Turing είναι ένα σύνολο τετράδων. Κάθε τετράδα έχει μία από τις ακόλουθες μορφές:

1. $q_i \, s_j \, s_k \, q_l$
2. $q_i \, s_j \, R \, q_l$
3. $q_i \, s_j \, L \, q_l$

Μία τετράδα της μορφής 1 σημαίνει ότι αν η μηχανή βρίσκεται στην κατάσταση q_i και διαβάσει από το τρέχον κελί το σύμβολο s_j, τότε γράφει στο τρέχον κελί το σύμβολο s_k και μεταβαίνει στην κατάσταση q_l. Μία τετράδα της μορφής 2 σημαίνει ότι αν η μηχανή βρίσκεται στην κατάσταση q_i και διαβάσει από το τρέχον κελί το σύμβολο s_j, τότε μετακινείται κατά ένα κελί δεξιά (το οποίο γίνεται τρέχον κελί) και μεταβαίνει στην κατάσταση q_l. Αντίστοιχα, μία τετράδα της μορφής 3 σημαίνει ότι αν η μηχανή βρίσκεται στην κατάσταση q_i και διαβάσει από το τρέχον κελί το σύμβολο s_j, τότε μετακινείται κατά ένα κελί αριστερά και μεταβαίνει στην κατάσταση q_l.

Θεωρούμε ότι αρχική κατάσταση της μηχανής Turing είναι η κατάσταση q_1. Ακόμη, αν η μηχανή βρίσκεται στην κατάσταση q_i με τρέχον σύμβολο s_j αλλά δεν υπάρχει τετράδα στο πρόγραμμα της μηχανής που να αρχίζει με $q_i \, s_j$, ορίζουμε ότι η μηχανή Turing σταματάει την εκτέλεσή της. Επιβάλλουμε ακόμη στο πρόγραμμα της μηχανής Turing τον περιορισμό ότι δεν μπορεί να περιλαμβάνει δύο τετράδες που να ξεκινούν με το ίδιο ζεύγος $q_i \, s_j$, εξασφαλίζοντας ότι η μηχανή δεν έχει ποτέ να επιλέξει ανάμεσα σε δύο ή περισσότερες έγκυρες ενέργειες. Ουσιαστικά δηλαδή το πρόγραμμα καθορίζει μια συνάρτηση μετάβασης $\delta : \mathcal{Q} \times \mathcal{S} \rightarrow (\mathcal{S} \cup \{L, R\}) \times \mathcal{Q}$, με τη σημασιολογία που εξηγήσαμε παραπάνω. Μία μηχανή Turing της οποίας το πρόγραμμα τηρεί αυτόν τον περιορισμό λέγεται και ντετερμινιστική μηχανή Turing (DTM).

Η είσοδος μιας μηχανής Turing είναι μία πεπερασμένη ακολουθία συμβόλων από το $\mathcal{S} \setminus \{s_0\}$. Θεωρούμε ότι η ακολουθία αυτή είναι ήδη γραμμένη στην ταινία σε συνεχόμενες θέσεις προτού αρχίσει η εκτέλεση του προγράμματος και όλες οι υπόλοιπες θέσεις περιέχουν το σύμβολο του κενού s_0. Η κεφαλή αρχικά βρίσκεται στο κελί αμέσως αριστερά από το πρώτο σύμβολο της εισόδου και η μηχανή, όπως έχουμε πει, βρίσκεται στην κατάσταση q_1. Συνεπώς η πρώτη μετάβαση της μηχανής θα καθοριστεί από την τετράδα που ξεκινάει με $q_1\, s_0$. Η έξοδος της μηχανής Turing είναι η ακολουθία συμβόλων που προκύπτει αν, όταν τερματιστεί η εκτέλεσή της, αγνοήσουμε όλα τα κενά και συμπτήξουμε τα υπόλοιπα σύμβολα σε μία ακολουθία με τη σειρά που είναι γραμμένα στην ταινία[1].

Ορίζουμε ως στιγμιότυπο της μηχανής Turing το περιεχόμενο της ταινίας, τη θέση στην οποία βρίσκεται η κεφαλή και την εσωτερική κατάσταση της μηχανής σε δεδομένη στιγμή της εκτέλεσης του προγράμματος. Αναπαριστούμε ένα στιγμιότυπο γράφοντας εκείνο το μέρος της ταινίας που περιέχει πεπερασμένο αριθμό στοιχείων του $\mathcal{S} \setminus \{s_0\}$, βάζοντας σε αγκύλες το σύμβολο του τρέχοντος κελιού και δίνοντας σε παρένθεση την τρέχουσα κατάσταση, π.χ.:

$$\ldots\ s_0\, s_0\, s_3\, s_1\, s_0\, [s_1]\, s_0\, s_2\, s_0\, s_0\ \ldots (q_4)$$

Παράδειγμα

Εφόσον η μηχανή Turing είναι ισοδύναμη με τα υπόλοιπα υπολογιστικά μοντέλα, περιμένει κανείς ότι θα είναι σε θέση να υπολογίσει και απλές αριθμητικές συναρτήσεις. Απομένει να δούμε κατά πόσο είναι εύκολη η υλοποίηση μίας τέτοιας συνάρτησης με πρόγραμμα μηχανής Turing. Χρησιμοποιούμε για ευκολία την unary αναπαράσταση αριθμών, όπου κάθε ακέραιος $n \geq 0$ αναπαρίσταται με μία ακολουθία $n+1$ ίδιων συμβόλων. Θα δώσουμε στη συνέχεια ένα πρόγραμμα που υπολογίζει τη συνάρτηση $f(x) = \lfloor \frac{x}{2} \rfloor$, για ακέραιο $x \geq 0$.

Έστω το σύνολο καταστάσεων $\mathcal{Q} = \{q_1, q_2, q_3, q_4\}$ και το σύνολο συμβόλων $\mathcal{S} = \{s_0, s_1\}$. Το πρόγραμμα μηχανής Turing για την παραπάνω συνάρτηση δίνεται από τις εξής τετράδες:

$$q_1\, s_0\, R\, q_2$$

$$q_2\, s_0\, R\, q_5$$

$$q_2\, s_1\, R\, q_3$$

$$q_3\, s_0\, R\, q_5$$

[1] Θα μπορούσαμε να ορίσουμε πιο αυστηρά, ότι η μηχανή πρέπει να τερματίζει με την κεφαλή να βρίσκεται στο κενό αμέσως αριστερά από τον πρώτο χαρακτήρα της εξόδου και να μην υπάρχουν κενά μεταξύ των συμβόλων της ακολουθίας εξόδου.

$$q_3\ s_1\ s_0\ q_4$$

$$q_4\ s_0\ R\ q_2$$

Την πληροφορία αυτή μπορούμε να την αναπαραστήσουμε και με τη μορφή του πίνακα μετάβασης.

	Κατάσταση				
Σύμβολο	q_1	q_2	q_3	q_4	q_5
s_0	$R\,q_2$	$R\,q_5$	$R\,q_5$	$R\,q_2$	
s_1		$R\,q_3$	$s_0\,q_4$		

Η ακολουθία διαδοχικών στιγμιοτύπων για τον υπολογισμό του $f(5) = 2$ δίνεται παρακάτω (η είσοδος αναπαρίσταται από έξι συνεχόμενα σύμβολα s_1, με βάση την unary αναπαράσταση).

$$\ldots\ s_0\ [s_0]\ s_1\ s_1\ s_1\ s_1\ s_1\ s_1\ s_0\ s_0\ \ldots (q_1)$$

$$\ldots\ s_0\ s_0\ [s_1]\ s_1\ s_1\ s_1\ s_1\ s_1\ s_0\ s_0\ \ldots (q_2)$$

$$\ldots\ s_0\ s_0\ s_1\ [s_1]\ s_1\ s_1\ s_1\ s_1\ s_0\ s_0\ \ldots (q_3)$$

$$\ldots\ s_0\ s_0\ s_1\ [s_0]\ s_1\ s_1\ s_1\ s_1\ s_0\ s_0\ \ldots (q_4)$$

$$\ldots\ s_0\ s_0\ s_1\ s_0\ [s_1]\ s_1\ s_1\ s_1\ s_0\ s_0\ \ldots (q_2)$$

$$\ldots\ s_0\ s_0\ s_1\ s_0\ s_1\ [s_1]\ s_1\ s_1\ s_0\ s_0\ \ldots (q_3)$$

$$\ldots\ s_0\ s_0\ s_1\ s_0\ s_1\ [s_0]\ s_1\ s_1\ s_0\ s_0\ \ldots (q_4)$$

$$\ldots\ s_0\ s_0\ s_1\ s_0\ s_1\ s_0\ [s_1]\ s_1\ s_0\ s_0\ \ldots (q_2)$$

$$\ldots\ s_0\ s_0\ s_1\ s_0\ s_1\ s_0\ s_1\ [s_1]\ s_0\ s_0\ \ldots (q_3)$$

$$\ldots\ s_0\ s_0\ s_1\ s_0\ s_1\ s_0\ s_1\ [s_0]\ s_0\ s_0\ \ldots (q_4)$$

$$\ldots\ s_0\ s_0\ s_1\ s_0\ s_1\ s_0\ s_1\ s_0\ [s_0]\ s_0\ \ldots (q_2)$$

$$\ldots\ s_0\ s_0\ s_1\ s_0\ s_1\ s_0\ s_1\ s_0\ s_0\ [s_0]\ \ldots (q_5)$$

Το αποτέλεσμα του υπολογισμού είναι $s_1\ s_1\ s_1$, που αναπαριστά τον αριθμό 2.

Εφόσον η μηχανή Turing παίρνει μία ακολουθία συμβόλων εισόδου και παράγει μία ακολουθία συμβόλων εξόδου, λέμε ότι υπολογίζει μια συνάρτηση $f : A^\star \to A^\star$, όπου $A = \mathcal{S} \setminus \{s_0\}$ και $A^\star$ συμβολίζει το σύνολο των ακολουθιών οποιουδήποτε μήκους που αποτελούνται από σύμβολα του A[2]. Παρατηρούμε ότι πουθενά στους περιορισμούς για το πρόγραμμα της μηχανής Turing δεν αναφέρεται ότι για κάθε είσοδο πρέπει να υπάρχει ένα τελικό στιγμιότυπο της εκτέλεσής του. Αυτό σημαίνει ότι η εκτέλεση για ορισμένες εισόδους ενδέχεται να μη σταματάει ποτέ (πρβλ. με ατέρμονα βρόγχο στις γνωστές γλώσσες προγραμματισμού PASCAL, C). Λέμε ότι για αυτές τις εισόδους η συνάρτηση f που υπολογίζει η μηχανή Turing δεν ορίζεται, και τότε η f λέγεται μερική συνάρτηση[3]. Αν επιθυμούμε να υπολογίσουμε συνάρτηση με περισσότερα από ένα ορίσματα μπορούμε να κωδικοποιήσουμε τα ορίσματα σε ένα όρισμα και να περάσουμε αυτό ως είσοδο στη μηχανή Turing. Βολεύει όμως να θεωρούμε ότι τα ορίσματα (έστω $x_1, x_2, \ldots, x_n$) δίνονται ξεχωριστά στην αρχή της εκτέλεσης, χωριζόμενα με ένα κενό. Δηλαδή το αρχικό στιγμιότυπο έχει ως εξής:

$$\ldots\ s_0\, [s_0]\, x_1\, s_0\, x_2\, s_0\ \ldots\ x_n\, s_0\, s_0\ \ldots (q_1)$$

Αν το μοντέλο της μηχανής Turing φαίνεται περιοριστικό, μπορεί κανείς να σκεφτεί πολλές βελτιώσεις και να δημιουργήσει νέα μοντέλα που φαινομενικά προσφέρουν περισσότερες δυνατότητες από τη μηχανή Turing όπως την ορίσαμε παραπάνω, όμως αποδεικνύονται τελικά ισοδύναμα με αυτήν (θέση του Church). Δίνουμε παρακάτω ορισμένες από αυτές τις παραλλαγές, στις οποίες συμπεριλαμβάνονται και κάποιες που φαίνονται να περιορίζουν τις δυνατότητες της μηχανής Turing. Τελικά όμως προκύπτει ότι οι περιορισμοί που θέτουν δεν είναι τόσο ισχυροί ώστε να διαταράξουν την ισοδυναμία της με τα άλλα μοντέλα.

Ταυτόχρονη εγγραφή και πιθανή μετακίνηση

Η συνάρτηση μετάβασης δ ορίζεται ως $\delta : \mathcal{Q} \times \mathcal{S} \to \mathcal{S} \times \{L, R, -\} \times \mathcal{Q}$. Δηλαδή κάθε φορά που διαβάζει ένα σύμβολο, η κεφαλή μπορεί να

[2]Πρόκειται για το άστρο του Kleene. Για περισσότερα βλέπε ενότητα 1.3.

[3]Αν η $f : A^\star \to A^\star$ ορίζεται για κάθε $x \in A^\star$ καλείται ολική συνάρτηση.

γράψει ένα σύμβολο και στη συνέχεια να μετακινηθεί αριστερά ή δεξιά, ή να παραμείνει στην ίδια θέση (αν έχουμε L ή R ή $-$ αντίστοιχα). Το πρόγραμμα της μηχανής Turing αποτελείται τώρα από πεντάδες.

Ταινία άπειρη μόνο προς τη μία κατεύθυνση

Αυτή ήταν η παραλλαγή που όρισε αρχικά και χρησιμοποίησε ο Turing. Η ταινία έχει ένα κελί από το οποίο η κεφαλή δεν μπορεί να μετακινηθεί αριστερά. Ο ορισμός του τι συμβαίνει αν η κεφαλή βρίσκεται στο αριστερότερο κελί που περιέχει το σύμβολο s_j, η μηχανή είναι στην κατάσταση q_i και υπάρχει τετράδα της μορφής $q_i\, s_j\, L\, q_l$ στο πρόγραμμα της μηχανής Turing είναι αυθαίρετος και δεν επηρεάζει την ισοδυναμία του μοντέλου με τα υπόλοιπα. Μπορεί κανείς να ορίσει ότι σε μια τέτοια περίπτωση η εκτέλεση του προγράμματος σταματάει.

Πολλαπλές ταινίες

Η μηχανή διαθέτει $k \geq 1$ ταινίες με τις αντίστοιχες κεφαλές. Η συνάρτηση μετάβασης μετατρέπεται ώστε να ορίζει την συμπεριφορά όλων των κεφαλών ανάλογα με το σύμβολο που διαβάζει η κάθε μία και την εσωτερική κατάσταση της μηχανής, δηλαδή παίρνει τη μορφή $\delta : \mathcal{Q} \times \mathcal{S}^k \rightarrow (\mathcal{S} \cup \{L, R\})^k \times \mathcal{Q}$.

Ταινίες εισόδου-εξόδου

Εκτός από τη συνηθισμένη ταινία, η μηχανή διαθέτει μία επιπλέον ταινία εισόδου στην οποία επιτρέπεται μόνο ανάγνωση και μία επιπλέον ταινία εξόδου στην οποία επιτρέπεται μόνο εγγραφή. Η είσοδος λαμβάνεται από την ταινία εισόδου και η έξοδος γράφεται στην ταινία εξόδου. Η αρχική ταινία χρησιμοποιείται ως πρόχειρο για την εκτέλεση του υπολογισμού και αναφέρεται ως ταινία εργασίας (επιτρέπεται εγγραφή και ανάγνωση).

Ταινία μίας εγγραφής

Κάθε κελί της ταινίας μπορεί να αλλάξει περιεχόμενο το πολύ μία φορά κατά τη διάρκεια της εκτέλεσης του υπολογισμού.

Μηχανή με ολική επαναφορά προς τα αριστερά

Η δυνατότητα μετακίνησης της κεφαλής προς τα αριστερά αντικαθίσταται από τη δυνατότητα μετακίνησης της κεφαλής απ'ευθείας στο αριστερότερο κελί που περιέχει σύμβολο διάφορο του s_0 (reset). Η συνάρτηση μετάβασης είναι της μορφής $\delta : \mathcal{Q} \times \mathcal{S} \to (\mathcal{S} \cup \{R, RST\}) \times \mathcal{Q}$.

Μηχανή με χαρτί δύο διαστάσεων

Η γραμμική χαρτοταινία της μηχανής Turing αντικαθίσταται από ένα επίπεδο φύλλο χαρτί, άπειρο προς όλες τις κατευθύνσεις. Είναι και αυτό χωρισμένο σε κελιά που το καθένα περιέχει ένα σύμβολο. Η κεφαλή μπορεί να μετακινείται κατά ένα τετράγωνο προς οποιαδήποτε κατεύθυνση και η συνάρτηση μετάβασης έχει τη μορφή $\delta : \mathcal{Q} \times \mathcal{S} \to (\mathcal{S} \cup \{U, D, L, R\}) \times \mathcal{Q}$, όπου τα U, D αντιπροσωπεύουν κίνηση προς τα πάνω ή προς τα κάτω. Ο ορισμός γενικεύεται και για χώρους περισσότερων διαστάσεων.

Συνδυασμοί των παραπάνω

Για παράδειγμα, μπορούμε να φανταστούμε μία μηχανή Turing με k κεφαλές όπου κάθε κεφαλή κινείται σε ένα ογδοημόριο χώρου τριών διαστάσεων και κάθε κελί μπορεί να αλλάξει το σύμβολό του το πολύ μία φορά.

1.3 Κωδικοποίηση προβλημάτων απόφασης

Έστω ένα σύνολο συμβόλων Σ, που στη συνέχεια θα καλούμε αλφάβητο. Ορίζουμε:

$$\Sigma^k = \{a_1 a_2 \ldots a_k : a_1, a_2, \ldots, a_k \in \Sigma\} \ , \ k \geq 0$$

$$\Sigma^\star = \bigcup_{k \geq 0} \Sigma^k$$

Δηλαδή Σ^k είναι το σύνολο των συμβολοακολουθιών που αποτελούνται από σύμβολα του Σ και έχουν μήκος k, ενώ $\Sigma^\star$ είναι το σύνολο όλων των (πεπερασμένων) συμβολοακολουθιών που αποτελούνται από σύμβολα του Σ. Η συμβολοακολουθία μήκους 0 συμβολίζεται με ε και ονομάζεται κενή συμβολοακολουθία. Ένα σύνολο $L \subseteq \Sigma^\star$ ονομάζεται γλώσσα επί του αλφαβήτου

Σ. Για μία γλώσσα L, ένα βασικό πρόβλημα είναι η απάντηση στο ερώτημα αν μία δοσμένη συμβολοακολουθία ανήκει στη γλώσσα ή όχι.

Τροποποιούμε ελαφρώς τον ορισμό της μηχανής Turing, ώστε η εκτέλεση ενός προγράμματος να τερματίζει μόνο όταν φτάσει σε μία από τις δύο ειδικές καταστάσεις τερματισμού q_{acc} και q_{rej}. Λέμε ότι μία μηχανή Turing αποδέχεται την είσοδο x, αν και μόνο αν ολοκληρώνει την εκτέλεσή της με είσοδο x στην κατάσταση q_{acc}, και ότι απορρίπτει την είσοδο x αν και μόνο αν ολοκληρώνει την εκτέλεσή της με είσοδο x στην κατάσταση q_{rej}. Επίσης, μία μηχανή Turing M *αποδέχεται* τη γλώσσα L αν και μόνο αν ισχύει $x \in L \iff M$ αποδέχεται x. Ο ορισμός αυτός δεν καθορίζει τι συμβαίνει με τις εισόδους που δεν ανήκουν στη γλώσσα L που αποδέχεται μία μηχανή Turing. Για αυτές τις εισόδους, μπορεί είτε η εκτέλεση να ολοκληρώνεται στην κατάσταση q_{rej}, είτε να μην ολοκληρώνεται ποτέ. Λέμε ότι μία μηχανή Turing M *αποφασίζει* τη γλώσσα L αν και μόνο αν ισχύει (M αποδέχεται L) $\wedge$ ($x \notin L \iff M$ απορρίπτει x).

Μία μη ντετερμινιστική μηχανή Turing (NTM) είναι μία μηχανή Turing με τη διαφορά ότι στο πρόγραμμά της μπορούν να υπάρχουν περισσότερες από μία τετράδες που αρχίζουν με τον ίδιο συνδυασμό $q_i\, s_j$. Έτσι, σε κάθε βήμα η μη ντετερμινιστική μηχανή Turing έχει μία επιλογή για το ποια μετάβαση θα εκτελέσει. Τυπικά, η συνάρτηση μετάβασης ορίζεται ως $\delta : \mathcal{Q} \times \mathcal{S} \to 2^{(\mathcal{S} \cup \{L,R\}) \times \mathcal{Q}}$. Η διαδικασία υπολογισμού της εξόδου δεν είναι μία γραμμική ακολουθία στιγμιοτύπων αλλά ένα δέντρο. Κάθε μονοπάτι του δέντρου από τη ρίζα (αρχικό στιγμιότυπο) προς τα φύλλα (τελικό στιγμιότυπο) αναπαριστά μία δυνατή επιλογή μεταβάσεων[4]. Θεωρούμε ότι η μηχανή ακολουθεί όλα αυτά τα δυνατά μονοπάτια παράλληλα. Ισοδύναμα, μπορεί κανείς να σκέφτεται ότι σε κάθε σημείο επιλογής η μηχανή δημιουργεί αρκετά αντίγραφα του εαυτού της ώστε το καθένα να ακολουθήσει μία από τις δυνατές μεταβάσεις.

Μία μη ντετερμινιστική μηχανή Turing *αποδέχεται* τη γλώσσα L αν και μόνο αν για κάθε είσοδο $x \in L$ υπάρχει μονοπάτι εκτέλεσης που καταλήγει

[4]Ορισμένα μονοπάτια μπορεί να μην καταλήγουν σε τελικό στιγμιότυπο αλλά να συνεχίζουν να εκτελούνται επ'άπειρον.

στην κατάσταση q_{acc}, ενώ αν $x \notin L$ όλα τα μονοπάτια είτε είναι απείρου μήκους είτε καταλήγουν στην κατάσταση q_{rej}. Μία μη ντετερμινιστική μηχανή Turing *αποφασίζει* τη γλώσσα L αν και μόνο αν για κάθε είσοδο x όλα τα μονοπάτια καταλήγουν στην q_{acc} αν $x \in L$ και όλα τα μονοπάτια καταλήγουν στην q_{rej} αν $x \notin L$.

Πρόβλημα απόφασης είναι, εντελώς γενικά, ένα πρόβλημα που επιδέχεται απάντηση του τύπου ναι/όχι. Ένα πρόβλημα απόφασης ορίζεται από δύο μέρη: το πρώτο μέρος καθορίζει τη γενική μορφή που έχουν τα στιγμιότυπα του προβλήματος και το δεύτερο μέρος καθορίζει την ερώτηση που τίθεται επί ενός γενικού στιγμιοτύπου (η οποία έχει απάντηση "ναι" ή "όχι"). Για παράδειγμα, ένα γνωστό πρόβλημα στη θεωρητική πληροφορική είναι το **HAMILTON CIRCUIT**. Ένα στιγμιότυπο του προβλήματος αποτελείται από έναν γράφο $G(V, E)$, και η ερώτηση που τίθεται για το δοσμένο γράφο είναι αν έχει κύκλο Hamilton[5].

Ένας δεύτερος τύπος προβλημάτων που έχει συγγένεια με τα προβλήματα απόφασης είναι τα προβλήματα βελτιστοποίησης. Γενικά, σε ένα πρόβλημα βελτιστοποίησης ορίζεται μία συνάρτηση κόστους $f : S \to \mathbb{N}$, όπου το S είναι ένα σύνολο δυνατών λύσεων και ζητείται το κόστος της λύσης που ελαχιστοποιεί ή μεγιστοποιεί την f. Τα προβλήματα βελτιστοποίησης μπορούν να αντιστοιχιστούν σε προβλήματα απόφασης. Έστω ένα πρόβλημα ελαχιστοποίησης. Δημιουργούμε το αντίστοιχο πρόβλημα απόφασης ως εξής: "Έχει το πρόβλημα ελαχιστοποίησης λύση με κόστος $\leq B$;", όπου το B θεωρείται μέρος του στιγμιοτύπου του προβλήματος απόφασης (αντίστοιχα και για πρόβλημα μεγιστοποίησης). Για παράδειγμα, έστω το ακόλουθο πρόβλημα μεγιστοποίησης (**DISCRETE KNAPSACK**): "Δίνονται θετικοί ακέραιοι $p_1, p_2, \ldots, p_n, w_1, w_2, \ldots, w_n, C, B$. Υπάρχει ανάθεση τιμών στα x_i, με τους περιορισμούς $\sum_{i=1}^{n} w_i x_i \leq C$ και $x_i \in \{0, 1\}$, ώστε $\sum_{i=1}^{n} p_i x_i \geq B$;". Εφόσον η συνάρτηση κόστους είναι σχετικά εύκολο να υπολογιστεί, ένα πρόβλημα απόφασης δεν είναι πιο δύσκολο από το αντίστοιχο πρόβλημα βελτιστοποίη-

[5]Κύκλος Hamilton σε ένα γράφο είναι ένας κύκλος που διέρχεται ακριβώς μία φορά από κάθε κορυφή του.

σης. Πραγματικά, αν μπορούμε να λύσουμε το πρόβλημα βελτιστοποίησης εύκολα[6], τότε και το αντίστοιχο πρόβλημα απόφασης λύνεται εύκολα, συγκρίνοντας απλά το κόστος της βέλτιστης λύσης με το B. Αντιστρόφως, αν το πρόβλημα απόφασης λύνεται "δύσκολα", τότε και το αντίστοιχο πρόβλημα βελτιστοποίησης θα είναι τουλάχιστον εξίσου δύσκολο. Συνεπώς μπορούμε να περιοριζόμαστε σε προβλήματα απόφασης και τα αποτελέσματα της θεωρίας επεκτείνονται και σε προβλήματα βελτιστοποίησης.

Ο λόγος για τον οποίο περιοριζόμαστε σε προβλήματα απόφασης είναι ότι τα προβλήματα αυτά απεικονίζονται με πολύ φυσικό τρόπο σε προβλήματα ελέγχου για το αν μία συμβολοακολουθία w ανήκει σε κάποια γλώσσα L. Πραγματικά μία μηχανή Turing M που αποφασίζει για κάποια γλώσσα $L(M)$ ουσιαστικά λύνει το πρόβλημα απόφασης "$x \in L(M)$;". Έτσι αν μπορούμε να κωδικοποιήσουμε το γενικό στιγμιότυπο ενός προβλήματος απόφασης ως μία γλώσσα επί κάποιου αλφαβήτου, ουσιαστικά μπορούμε να χρησιμοποιήσουμε τη μηχανή Turing για να λύσει οποιοδήποτε (υπολογιστό) πρόβλημα απόφασης.

Η απεικόνιση από το σύνολο των στιγμιοτύπων ενός προβλήματος στο σύνολο των συμβολοακολουθιών μιας γλώσσας γίνεται μέσω ενός σχήματος κωδικοποίησης. Ένα σχήμα κωδικοποίησης e για ένα πρόβλημα απόφασης Π παρέχει έναν τρόπο με τον οποίο κάθε στιγμιότυπο του προβλήματος μπορεί να παρασταθεί με μία κατάλληλη συμβολοακολουθία επί ενός αλφαβήτου Σ. Έτσι, ένα πρόβλημα Π μαζί με το σχήμα κωδικοποίησης e για αυτό το πρόβλημα χωρίζουν το σύνολο $\Sigma^\star$ σε τρία ξένα μεταξύ τους υποσύνολα: αυτό που περιέχει συμβολοακολουθίες που δεν αναπαριστούν κάποιο στιγμιότυπο του Π, αυτό που περιέχει συμβολοακολουθίες που αναπαριστούν στιγμιότυπα του προβλήματος στα οποία η απάντηση είναι "όχι" και τέλος αυτό που περιέχει συμβολοακολουθίες που αναπαριστούν στιγμιότυπα του προβλήματος στα οποία η απάντηση είναι "ναι". Το τελευταίο σύνολο είναι αυτό που αντιστοιχίζουμε με ένα πρόβλημα Π και ένα σχήμα κωδικοποίησης e, και το συμβολίζουμε με $L[\Pi, e]$. Δεν είναι απαραίτητο να υπάρχει μόνο μία συμβολοακολουθία που κωδικοποιεί ένα συγκεκριμένο στιγμιότυπο.

[6] Δηλαδή σε πολυωνυμικό χρόνο με ντετερμινιστικό αλγόριθμο.

Στην πραγματικότητα, θα θέλαμε να καταλήγουμε σε συμπεράσματα για ένα πρόβλημα, ανεξαρτήτως του τρόπου με τον οποίο τα στιγμιότυπά του κωδικοποιούνται ως συμβολοακολουθίες. Αυτό επιτυγχάνεται αν περιοριστούμε στη χρήση μιας συγκεκριμένης κλάσης "λογικών" σχημάτων κωδικοποίησης E, τέτοιας ώστε αν $e, e' \in E$, δεδομένου ενός προβλήματος απόφασης Π μία ιδιότητα θα ισχύει για τη γλώσσα $L[\Pi, e]$ αν και μόνο αν ισχύει για τη γλώσσα $L[\Pi, e']$. Έτσι μπορούμε να διατυπώνουμε συμπεράσματα μιλώντας αφηρημένα για το πρόβλημα Π και υπονοώντας ότι μπορούμε να δώσουμε ένα σχήμα κωδικοποίησης e ώστε τα συμπεράσματα αυτά να ισχύουν για τη γλώσσα $L[\Pi, e]$. Προφανώς, εφόσον δεν λαμβάνουμε υπόψη κάποιο συγκεκριμένο σχήμα κωδικοποίησης, δεν έχουμε και συγκεκριμένο μήκος για τις συμβολοακολουθίες που αναπαριστούν τα στιγμιότυπα του προβλήματος. Αντί αυτού, θεωρούμε ότι για κάθε πρόβλημα απόφασης Π υπάρχει ανεξαρτήτως σχήματος κωδικοποίησης μία συνάρτηση μήκους $l : D_\Pi \to \mathbb{Z}^+$, όπου D_Π είναι το σύνολο των στιγμιοτύπων του προβλήματος, τέτοια ώστε για κάθε $e \in E$ η αντίστοιχη συνάρτηση μήκους εισόδου l_e που προκύπτει για το συγκεκριμένο σχήμα κωδικοποίησης είναι πολυωνυμικά συσχετισμένη με την l[7].

Η παραπάνω συζήτηση στηρίζεται στην κοινή αποδοχή του τι σημαίνει "λογικό" σχήμα κωδικοποίησης. Αν και δεν υπάρχει ικανοποιητικός τυπικός ορισμός της έννοιας αυτής, ένα λογικό σχήμα κωδικοποίησης θα πρέπει να είναι συνοπτικό και εύκολα αποκωδικοποιήσιμο. Πιο συγκεκριμένα, η κωδικοποίηση ενός στιγμιοτύπου δεν θα πρέπει να περιλαμβάνει επιπλέον σύμβολα ή πληροφορίες από ότι είναι απαραίτητο, και όλοι οι ακέραιοι αριθμοί που εμφανίζονται ως μέρος του στιγμιοτύπου θα πρέπει να γράφονται στο δυαδικό σύστημα αρίθμησης (ή γενικότερα σε οποιοδήποτε σύστημα αρίθμησης με βάση μεγαλύτερη του 1). Επίσης, στα πλαίσια της εύκολης αποκωδικοποίησης, θα πρέπει να υπάρχει γρήγορος αλγόριθμος ο οποίος να μπορεί να εξάγει τα χαρακτηριστικά ενός στιγμιοτύπου, δεδομένης της συμβολοακολουθίας που το

[7]Οι συναρτήσεις l και l_e λέγονται πολυωνυμικά συσχετισμένες αν και μόνο αν υπάρχουν δύο πολυώνυμα p_1, p_2 ώστε για κάθε στιγμιότυπο I να ισχύουν: $l(I) \leq p_1(l_e(I))$ και $l_e(I) \leq p_2(l(I))$.

αναπαριστά.

1.4 Οι κλάσεις P, NP. Πολυωνυμικές αναγωγές και NP-πληρότητα

Έστω μία ντετερμινιστική μηχανή Turing M που σταματάει για όλες τις εισόδους, δηλαδή δεν εμπλέκεται ποτέ σε άπειρο loop. Ορίζουμε την πολυπλοκότητα χρόνου της M ως μία συνάρτηση $f : \mathbb{N} \rightarrow \mathbb{N}$, όπου $f(n)$ είναι ο μέγιστος αριθμός μεταβάσεων που εκτελεί η μηχανή Turing αν η είσοδος έχει μήκος n. Λέμε τότε ότι το πρόγραμμα της μηχανής Turing M εκτελείται σε χρόνο $f(n)$. Ακόμα, αν $t : \mathbb{N} \rightarrow \mathbb{N}$ είναι μία συνάρτηση, ορίζουμε την κλάση πολυπλοκότητας χρόνου $\mathbf{TIME}(t(n))$ ως την κλάση των γλωσσών[8] για κάθε μία από τις οποίες υπάρχει μηχανή Turing που λειτουργεί σε χρόνο $O(t(n))$ και την αποφασίζει.

Αποδεικνύεται ότι κάθε μηχανή Turing με πολλές ταινίες που λειτουργεί σε χρόνο $t(n) \geq n$ [9] έχει μία ισοδύναμη μηχανή Turing με μία ταινία που λειτουργεί σε χρόνο $O(t^2(n))$. Επίσης, κάθε μη ντετερμινιστική μηχανή Turing με μία ταινία που λειτουργεί σε χρόνο $t(n) \geq n$ έχει μία ισοδύναμη ντετερμινιστική μηχανή Turing με μία ταινία που λειτουργεί σε χρόνο $O(2^{t(n)})$. Τα θεωρήματα αυτά επιδεικνύουν ένα σημαντικό διαχωρισμό: από τη μία πλευρά, η πολυπλοκότητα χρόνου μιας ντετερμινιστικής μηχανής Turing με μία ταινία δεν είναι παρά πολυωνυμικά μεγαλύτερη από την πολυπλοκότητα χρόνου μιας ισοδύναμης μηχανής Turing με πολλές ταινίες. Αντίθετα, αν από μη ντετερμινιστική μηχανή Turing μεταβαίνουμε σε ισοδύναμη ντετερμινιστική μηχανή Turing, η αύξηση στην πολυπλοκότητα είναι εκθετική. Για τους σκοπούς της θεωρίας πολυπλοκότητας, η πολυωνυμική διαφορά στο χρόνο εκτέλεσης θεωρείται μικρή, ενώ η εκθετική θεωρείται μεγάλη. Αρκεί να αναλογιστεί κανείς τη

[8] Στα επόμενα θα χρησιμοποιούμε την έννοια του προβλήματος απόφασης και της αντίστοιχης γλώσσας χωρίς διάκριση, με βάση τη συζήτηση της προηγούμενης ενότητας.

[9] Πρόκειται για μία λογική παραδοχή, αφού με λιγότερες από n μεταβάσεις μία μηχανή Turing δεν μπορεί να διαβάσει ολόκληρη την είσοδό της.

διαφορά μεταξύ των χρόνων εκτέλεσης μιας μηχανής Turing που λειτουργεί σε χρόνο n^3 και μιας άλλης που λειτουργεί σε χρόνο 2^n, για ένα τυπικό μέγεθος εισόδου $n = 1000$. Ενώ η πρώτη χρειάζεται ένα μεγάλο αλλά ανεκτό αριθμό μεταβάσεων, η δεύτερη χρειάζεται χρόνο κατά πολλές τάξεις μεγέθους μεγαλύτερο από την εκτιμόμενη ηλικία του σύμπαντος ακόμα και αν θεωρήσουμε εξωπραγματικό αριθμό μεταβάσεων ανά δευτερόλεπτο.

Όλα τα λογικά ντετερμινιστικά μοντέλα υπολογισμού είναι μεταξύ τους πολυωνυμικά ισοδύναμα. Αυτό σημαίνει ότι το καθένα μπορεί να εξομοιώσει οποιοδήποτε άλλο με το πολύ πολυωνυμική αύξηση στην πολυπλοκότητα χρόνου. Επομένως, αν αγνοήσουμε τις πολυωνυμικές διαφορές στο χρόνο εκτέλεσης, μπορούμε να μιλάμε γενικά για αλγορίθμους που λύνουν ένα πρόβλημα γρήγορα-αποδοτικά (εννοώντας σε πολυωνυμικό χρόνο) χωρίς να αναφερόμαστε σε συγκεκριμένο μοντέλο. Ουσιαστικά, θεωρούμε ότι ο αλγόριθμος δεν αναφέρεται σε κάποια αναπαράσταση του στιγμιοτύπου του προβλήματος για συγκεκριμένο υπολογιστικό μοντέλο, αλλά αναφέρεται απ'ευθείας στα συστατικά μέρη του στιγμιοτύπου (π.χ. στους κόμβους ενός γράφου ή στη συνάρτηση κόστους και όχι σε κάποια αναπαράσταση συνόλων ή συναρτήσεων με συμβολοακολουθίες). Η δικαιολογία αυτού του σκεπτικού είναι ότι, αν κάποιος έχει την υπομονή, μπορεί να χρησιμοποιήσει ένα λογικό σχήμα κωδικοποίησης για κάποιο υπολογιστικό μοντέλο, και στη συνέχεια να γράψει ένα πρόγραμμα για το μοντέλο αυτό που να λύνει το συγκεκριμένο πρόβλημα, καταλήγοντας τελικά το πολύ σε πολυωνυμική αύξηση της πολυπλοκότητας χρόνου, την οποία έχουμε θεωρήσει αμελητέα.

Ορίζουμε την κλάση των προβλημάτων απόφασης που λύνονται σε πολυωνυμικό χρόνο με ντετερμινιστικό αλγόριθμο, ως:

$$\mathbf{P} = \bigcup_{k \geq 0} \mathbf{TIME}(n^k)$$

Η κλάση αυτή παίζει κεντρικό ρόλο στη θεωρία πολυπλοκότητας και είναι σημαντική γιατί παραμένει αμετάβλητη ανεξάρτητα από το μοντέλο υπολογισμού που θα χρησιμοποιήσουμε, με την προϋπόθεση ότι είναι πολυωνυμικά ισοδύναμο με την ντετερμινιστική μηχανή Turing με μία ταινία. Επίσης, τα προβλήματα που ανήκουν σε αυτήν την κλάση είναι ουσιαστικά αυτά που μπορούν να λυθούν αποδοτικά με έναν ηλεκτρονικό υπολογιστή. Βέβαια, είναι αμφίβολο το κατά πόσο ένα πρόβλημα που λύνεται από αλγόριθμο χρονικής πολυπλοκότητας π.χ. $O(n^{100})$ μπορεί να θεωρηθεί αποδοτικά λυμένο. Όμως, η θεώρηση ότι ένας πολυωνυμικός αλγόριθμος είναι αποδοτικός έχει αποδειχθεί χρήσιμη στην πράξη. Συνήθως, όταν ανακαλύπτεται πολυωνυμικός αλγόριθμος για ένα πρόβλημα ακολουθούν επιπλέον βελτιώσεις στην πολυπλοκότητά του, ώστε ακόμα και αν αρχικά ο αλγόριθμος ήταν της τάξης του n^{100} να καταστεί τελικά και πρακτικά χρήσιμος.

Αλγόριθμοι που εκτελούνται σε εκθετικό χρόνο συνήθως εμφανίζονται όταν λύνουμε ένα πρόβλημα με εξαντλητική αναζήτηση στο χώρο των λύσεων. Αν και για ορισμένα προβλήματα είναι δυνατό να αποφύγουμε την εξαντλητική αναζήτηση και να καταλήξουμε με κάποιο πολυωνυμικό αλγόριθμο, για κάποια άλλα δεν έχει επιτευχθεί κάτι τέτοιο ούτε είναι γνωστό αν υπάρχει πολυωνυμικός αλγόριθμος που να τα λύνει. Το ενδιαφέρον είναι ότι πολλά από αυτά τα προβλήματα φαίνεται να είναι κατά κάποιο τρόπο συνδεδεμένα. Αν βρεθεί πολυωνυμικός αλγόριθμος που να λύνει κάποιο από αυτά, τότε όλα μπορούν να λυθούν με πολυωνυμικό αλγόριθμο. Αν αποδειχθεί ότι δεν υπάρχει πολυωνυμικός αλγόριθμος για κάποιο από αυτά, τότε δεν θα υπάρχει για κανένα.

Ορίζουμε και τυπικά την πολυπλοκότητα χρόνου μιας μη ντετερμινιστικής μηχανής Turing. Αν M είναι μία μη ντετερμινιστική μηχανή Turing που αποφασίζει κάποια γλώσσα L, η πολυπλοκότητα χρόνου της M είναι η συνάρτηση $f : \mathbb{N} \to \mathbb{N}$ όπου $f(n)$ είναι ο μέγιστος αριθμός μεταβάσεων που εκτελεί η M σε οποιοδήποτε μονοπάτι υπολογισμού και για οποιαδήποτε είσοδο μήκους n. Επίσης ορίζουμε την κλάση πολυπλοκότητας χρόνου $\mathbf{NTIME}(t(n))$ ως την κλάση των γλωσσών για κάθε μία από τις οποίες υπάρχει μη ντετερμινιστική μηχανή Turing που λειτουργεί σε χρόνο $O(t(n))$ και την αποφασίζει, καθώς και την κλάση των προβλημάτων απόφασης που λύνονται σε πολυωνυμικό χρόνο με μη ντετερμινιστικό αλγόριθμο ως:

$$\mathbf{NP} = \bigcup_{k \geq 0} \mathbf{NTIME}(n^k)$$

Η κλάση $\mathbf{NP}$, όπως και η $\mathbf{P}$, παραμένει αμετάβλητη ως προς την επιλογή μη ντετερμινιστικού υπολογιστικού μοντέλου επειδή όλα τα λογικά[10] μη ντετερμινιστικά υπολογιστικά μοντέλα είναι μεταξύ τους πολυωνυμικά ισοδύναμα. Είναι προφανές ότι $\mathbf{P} \subseteq \mathbf{NP}$, αφού μία ντετερμινιστική μηχανή Turing μπορεί να θεωρηθεί ως μία μη ντετερμινιστική μηχανή Turing όπου σε κάθε στιγμιότυπο υπάρχει μόνο μία δυνατή επόμενη μετάβαση. Όμως δεν είναι γνωστό αν $\mathbf{P} \subset \mathbf{NP}$ ή $\mathbf{P} = \mathbf{NP}$. Οι περισσότεροι ερευνητές πιστεύουν ότι $\mathbf{P} \subset \mathbf{NP}$, όμως κανείς μέχρι σήμερα δεν έχει αποδείξει την ύπαρξη μιας γλώσσας που να ανήκει στην κλάση $\mathbf{NP}$ αλλά όχι στην $\mathbf{P}$.

Στις αρχές της δεκαετίας του 1970 έγινε σημαντική πρόοδος στην κατεύθυνση της απάντησης αυτού του ερωτήματος, με την ανακάλυψη από τους Stephen Cook και Leonid Levin ορισμένων προβλημάτων της κλάσης $\mathbf{NP}$, των οποίων η πολυπλοκότητα χρόνου συνδέεται με την πολυπλοκότητα χρόνου όλων των προβλημάτων της κλάσης αυτής με τον εξής τρόπο: αν υπάρχει πολυωνυμικός αλγόριθμος για κάποιο από αυτά τα προβλήματα, τότε υπάρχει πολυωνυμικός αλγόριτθμος για κάθε πρόβλημα της κλάσης $\mathbf{NP}$. Αυτά τα προβλήματα λέγονται $\mathbf{NP}$-πλήρη. Βασικό εργαλείο για τον τυπικό ορισμό της κλάσης των $\mathbf{NP}$-πλήρων προβλημάτων είναι η έννοια της αναγωγής.

Έστω A, B γλώσσες ορισμένες επί ενός αλφαβήτου Σ. Λέμε ότι η γλώσσα A ανάγεται πολυωνυμικά[11] στη γλώσσα B και γράφουμε $A \leq_{\mathrm{m}}^{\mathrm{p}} B$ αν και μόνο αν υπάρχει συνάρτηση $f : \Sigma^\star \to \Sigma^\star$, $f \in \mathbf{FP}$ [12], τέτοια ώστε για κάθε $w \in \Sigma^\star$, $w \in A \iff f(w) \in B$. Η συνάρτηση f καλείται πολυωνυμική αναγωγή της A στην B και παρέχει έναν τρόπο μετατροπής του προβλήματος ελέγχου για το αν μία συμβολοακολουθία ανήκει στην A, στο αντίστοιχο πρόβλημα για την B. Η μετατροπή γίνεται σε πολυωνυμικό χρόνο, γεγονός που σημαίνει ότι αν

[10] Σε ένα λογικό μη ντετερμινιστικό υπολογιστικό μοντέλο, κάθε βήμα ενός μη ντετερμινιστικού αλγορίθμου πρέπει να έχει προφανή υλοποίηση σε μη ντετερμινιστικό πολυωνυμικό χρόνο.

[11] Πρόκειται για την αναγωγή κατά Karp.

[12] Η κλάση $\mathbf{FP}$ ορίζεται ως η κλάση των συναρτήσεων από το $\Sigma^\star$ στο $\Sigma^\star$ για κάθε μία από τις οποίες υπάρχει ντετερμινιστική μηχανή Turing πολυωνυμικού χρόνου που για κάθε είσοδο $w \in \Sigma^\star$ ολοκληρώνει την εκτέλεσή της με έξοδο $f(w)$.

$A \leq_{\mathrm{m}}^{\mathrm{p}} B$ και $B \in \mathbf{P}$, τότε θα ισχύει και $A \in \mathbf{P}$. Πραγματικά, για να βρούμε σε πολυωνυμικό χρόνο τη λύση για το πρόβλημα A, αρκεί να εφαρμόσουμε την f στο στιγμιότυπο του A, να εφαρμόσουμε τον πολυωνυμικό αλγόριθμο για το παραγόμενο στιγμιότυπο του B, και να δώσουμε ως έξοδο την απάντηση που θα δώσει ο αλγόριθμος για το B. Επειδή το $f(w)$ υπολογίζεται σε πολυωνυμικό χρόνο, θα έχει μήκος το πολύ πολυωνυμικά μεγαλύτερο από το μήκος του w. Επειδή η σύνθεση πολυωνύμων είναι η ίδια πολυώνυμο, η διαδικασία που μόλις περιγράψαμε για τη λύση του A αποτελεί πολυωνυμικό αλγόριθμο. Η σχέση $\leq_{\mathrm{m}}^{\mathrm{p}}$ μεταξύ γλωσσών (ή προβλημάτων απόφασης) ικανοποιεί την ανακλαστική και τη μεταβατική ιδιότητα.

Η κλάση των **NP**-πλήρων προβλημάτων (συμβολίζεται και με **NPC**) ορίζεται τυπικά ως εξής:

$$\mathbf{NPC} = \{L : L \in \mathbf{NP} \wedge \forall L' \in \mathbf{NP}(L' \leq_{\mathrm{m}}^{\mathrm{p}} L)\}$$

Δηλαδή πρόκειται για εκείνα τα προβλήματα της κλάσης **NP**, στα οποία μπορούν να αναχθούν πολυωνυμικά όλα τα υπόλοιπα προβλήματα αυτής της κλάσης. Αν όλα τα προβλήματα της κλάσης **NP** ανάγονται σε ένα πρόβλημα L, χωρίς να είναι γνωστό αν $L \in \mathbf{NP}$, τότε το L καλείται **NP**-δύσκολο. Κατά μία έννοια τα **NP**-πλήρη προβλήματα είναι τα δυσκολότερα προβλήματα της κλάσης **NP**. Αν οποιοδήποτε από αυτά τα προβλήματα αποδειχθεί ότι έχει ντετερμινιστικό πολυωνυμικό αλγόριθμο που το επιλύει, τότε $\mathbf{P} = \mathbf{NP}$. Για να αποδείξουμε ότι ένα πρόβλημα $Q \in \mathbf{NP}$ είναι **NP**-πλήρες, αρκεί να αποδείξουμε ότι για κάποιο $R \in \mathbf{NPC}$ ισχύει $R \leq_{\mathrm{m}}^{\mathrm{p}} Q$. Επομένως, από τη στιγμή που γνωρίζουμε έστω και ένα **NP**-πλήρες πρόβλημα μπορούμε να χρησιμοποιούμε τη διαδικασία της αναγωγής για να ανακαλύψουμε και νέα **NP**-πλήρη προβλήματα. Με αυτόν τον τρόπο χιλιάδες προβλήματα έχουν αποδειχθεί ότι είναι **NP**-πλήρη. Πρωτού όμως μπορέσει να αποδώσει καρπούς μία τέτοια προσέγγιση, χρειάζεται ένα πρώτο **NP**-πλήρες πρόβλημα. Ο Cook απέδειξε το 1971 ότι το πρόβλημα **SAT** είναι **NP**-πλήρες [Coo71]. Το **SAT** διατυπώνεται ως εξής: "Δίνεται μία λογική έκφραση σε κανονική συζευκτική μορφή (CNF). Υπάρχει απονομή αλήθειας στις μεταβλητές της, τέτοια ώστε η λογική έκφραση να αποτιμάται ως αληθής (ικανοποιήσιμη έκφραση);".

1.5 Περιορισμοί χώρου. Οι κλάσεις PSPACE, NPSPACE

Στα προηγούμενα μας ενδιέφεραν μηχανές Turing που μπορούν να αποφασίσουν γλώσσες σε περιορισμένο χρόνο, συγκεκριμένα πολυωνυμικό ως προς το μέγεθος της εισόδου. Έχει νόημα να θεωρήσουμε μηχανές Turing που λειτουργούν χωρίς περιορισμό χρόνου, αλλά με περιορισμό χώρου. Για την μέτρηση του χώρου που χρειάζεται μία μηχανή Turing για να εκτελέσει τον υπολογισμό της, θα χρησιμοποιήσουμε μία (ακόμη) παραλλαγή του βασικού μοντέλου της μηχανής Turing:

Μηχανή με ταινίες εισόδου και εξόδου

Προκείται για μία μηχανή Turing με $k > 2$ ταινίες. Η πρώτη από αυτές προορίζεται μόνο για ανάγνωση και καλείται *ταινία εισόδου*, ενώ η τελευταία προορίζεται μόνο για εγγραφή και καλείται *ταινία εξόδου*. Οι υπόλοιπες ταινίες συμπεριφέρονται ακριβώς όπως οι ταινίες μίας μηχανής Turing με πολλαπλές ταινίες και καλούνται *ταινίες εργασίας*.

Ο *χώρος* που απαιτείται για μία εκτέλεση ορίζεται ως το άθροισμα των μηκών των συμβολοσειρών που βρίσκονται στις ταινίες εργασίας στο τελικό στιγμιότυπο του υπολογισμού[13]. Για μη ντετερμινιστικούς υπολογισμούς, ο χώρος που απαιτείται για μία εκτέλεση είναι ο χώρος που απαιτείται από εκείνο το υπολογιστικό μονοπάτι που απαιτεί τον περισσότερο χώρο.

Κατ'αναλογία με τις κλάσεις **TIME**$(f(n))$, **NTIME**$(f(n))$, ορίζουμε τις κλάσεις **SPACE**$(f(n))$ και **NSPACE**$(f(n))$. Λέμε ότι μία γλώσσα L ανήκει στην κλάση **SPACE**$(f(n))$, αν υπάρχει (ντετερμινιστική) μηχανή Turing με είσοδο και έξοδο που λειτουργεί σε χώρο $O(f(n))$ και αποφασίζει την L. Αντίστοιχα, μία γλώσσα L ανήκει στην κλάση **NSPACE**$(f(n))$ αν υπάρχει μη ντετερμινιστική μηχανή Turing με είσοδο και έξοδο που λειτουργεί σε χώρο

[13]Θεωρούμε ότι το μήκος των συμβολοσειρών αυτών είναι αύξουσα συνάρτηση του πλήθους των μεταβάσεων της μηχανής. Αν η μηχανή γράφει κενούς χαρακτήρες στο τέλος της συμβολοσειράς, ή αντικαθιστά χαρακτήρες από το τέλος της συμβολοσειράς με κενούς, θεωρούμε ότι οι κενοί χαρακτήρες αποτελούν μέρος της συμβολοσειράς.

$O(f(n))$ και αποφασίζει την L. Για λόγους παρόμοιους με την περίπτωση του περιορισμού χρόνου, μας ενδιαφέρουν κυρίως πολυωνυμικά (ως προς το μήκος της εισόδου) άνω φράγματα στις απαιτήσεις χώρου. Σε αντιστοιχία με τις κλάσεις **P**, **NP**, ορίζουμε λοιπόν τις κλάσεις **PSPACE** και **NPSPACE**:

$$\mathbf{PSPACE} = \bigcup_{k \geq 0} \mathbf{SPACE}(n^k)$$

$$\mathbf{NPSPACE} = \bigcup_{k \geq 0} \mathbf{NSPACE}(n^k)$$

Από την εργασία του W. J. Savitch προκύπτει το εξής πολύ σημαντικό αποτέλεσμα:

Θεώρημα 1.5.1 *[Sav70]* **PSPACE = NPSPACE**

Δηλαδή η άρση της απαίτησης της ντετερμινιστικής υπολογιστικής ακολουθίας δεν προσφέρει επιπλέον δυνατότητες στο υπολογιστικό μας μοντέλο όταν περιοριζόμαστε σε χώρο πολυωνυμικό ως προς το μέγεθος της εισόδου.

Μία γλώσσα L λέγεται **PSPACE**-πλήρης αν $L \in \mathbf{PSPACE}$ και για κάθε $L' \in \mathbf{PSPACE}$ ισχύει ότι $L' \leq_{\mathrm{m}}^{\mathrm{p}} L$. Τα πρώτα **PSPACE**-πλήρη προβλήματα εμφανίζονται στην εργασία του R. M. Karp [Kar72]. Οι Stockmeyer και Meyer αποδεικνύουν [SM73] ότι το πρόβλημα **QBF** είναι **PSPACE**-πλήρες.

1.5.1 Το πρόβλημα QBF

Για τη διατύπωση του προβλήματος **QBF** απαιτούνται κάποιοι ορισμοί:

Ορισμός 1.5.2 Boolean μεταβλητή *είναι ένα οποιοδήποτε σύμβολο με το οποίο μπορούμε να αντιστοιχήσουμε μία από τις δύο* αληθοτιμές 0 *ή* 1 *(μερικές φορές τις συμβολίζουμε με false ή true, αντίστοιχα).*

Ορισμός 1.5.3 *Αν το X είναι σύνολο boolean μεταβλητών, ορίζουμε το σύνολο $\mathcal{F}_X$ των* boolean τύπων επί του X *ως το μικρότερο σύνολο για το οποίο ισχύουν:*

- $0 \in \mathcal{F}_X$, $1 \in \mathcal{F}_X$.

- $x \in X \Rightarrow x \in \mathcal{F}_X$
- $F_1 \in \mathcal{F}_X \wedge F_2 \in \mathcal{F}_X \Rightarrow (F_1 \wedge F_2) \in \mathcal{F}_X \wedge (F_1 \vee F_2) \in \mathcal{F}_X \wedge \neg(F_1) \in \mathcal{F}_X$

Ορισμός 1.5.4 *Αν το X είναι σύνολο boolean μεταβλητών, ορίζουμε το σύνολο $\mathcal{QF}_X$ των* boolean τύπων με ποσοδείκτες επί του *X ως το μικρότερο σύνολο για το οποίο ισχύουν:*

- $F \in \mathcal{F}_X \Rightarrow F \in \mathcal{QF}_X$
- $x \in X \wedge F \in \mathcal{QF}_X \Rightarrow (\exists x F) \in \mathcal{QF}_X \wedge (\forall x F) \in \mathcal{QF}_X$

Ορισμός 1.5.5 *Αν $F \in \mathcal{QF}_X$, οι* ελεύθερες μεταβλητές *του F είναι εκείνες οι μεταβλητές $x \in X$ που δεν δεσμεύονται από ποσοδείκτη $\forall x$ ή $\exists x$. Οι θέσεις στις οποίες εμφανίζεται το σύμβολο μίας ελεύθερης μεταβλητής ονομάζονται* ελεύθερες εμφανίσεις *της μεταβλητής.*

Ορισμός 1.5.6 *Αν $F \in \mathcal{QF}_X$, η έκφραση $F|_{x:=a}$, όπου $a \in X \cup \{0,1\}$, συμβολίζει τον τύπο που προκύπτει από τον F αν κάθε ελεύθερη εμφάνιση της μεταβλητής x στον F αντικατασταθεί με a.*

Ορισμός 1.5.7 Ανάθεση τιμών *στις μεταβλητές ενός συνόλου X boolean μεταβλητών είναι μία απεικόνιση από το σύνολο X στο σύνολο $\{0,1\}$. Αν $F \in \mathcal{QF}_X$ και V είναι μία ανάθεση τιμών στις μεταβλητές του X, ορίζεται η* αληθοτιμή *$V(F)$ ως εξής:*

- $V(F) = 0$, *αν* $F = 0$
- $V(F) = 1$, *αν* $F = 1$
- $V(F) = V(x)$, *αν* $F = x \in X$
- $V(F) = V(F_1) \vee V(F_2)$, *αν* $F = (F_1 \vee F_2)$
- $V(F) = V(F_1) \wedge V(F_2)$, *αν* $F = (F_1 \wedge F_2)$
- $V(F) = \neg V(F_1)$, *αν* $F = \neg(F_1)$

- $V(F) = V\left(F_1|_{x:=0}\right) \wedge V\left(F_1|_{x:=1}\right)$, *αν* $F = (\exists x F_1)$
- $V(F) = V\left(F_1|_{x:=0}\right) \vee V\left(F_1|_{x:=1}\right)$, *αν* $F = (\forall x F_1)$

Ορισμός 1.5.8 **QBF** *είναι το σύνολο των κωδικοποιήσεων όλων των boolean τύπων με ποσοδείκτες χωρίς ελεύθερες μεταβλητές, οι οποίοι έχουν αληθοτιμή 1.*

Θεώρημα 1.5.9 *[SM73] Το* **QBF** *είναι* **PSPACE***-πλήρες.*

Κεφάλαιο 2

Αυτο-αναγωγή

Οι αναδρομικοί αλγόριθμοι εμφανίζονται πολύ φυσικά όταν, κατά την προσπάθεια σχεδίασης ενός αλγορίθμου για κάποιο πρόβλημα, ασχολούμαστε με αντικείμενα που είναι ορισμένα με επαγωγικό τρόπο. Κλασικό παράδειγμα αντικειμένου που ορίζεται με αυτόν τον τρόπο είναι το δέντρο, και οι αλγόριθμοι που εκτελούν λειτουργίες πάνω σε δέντρα είναι εκ φύσεως αναδρομικοί.

Η έννοια της αυτο-αναγωγής συνδέεται στενά με την αναδρομή. Αν μία γλώσσα είναι αυτο-αναγωγική, ουσιαστικά έχουμε αμέσως διαθέσιμο έναν αναδρομικό αλγόριθμο για να αποφασίσουμε τη γλώσσα. Στην ενότητα 2.2 θα εκμεταλλευτούμε την ιδιότητα αυτο-αναγωγής του συνόλου **SAT** ώστε να καταλήξουμε σε συμπεράσματα σχετικά με την ύπαρξη δύσκολων γλωσσών με "λίγα" στοιχεία στις κλάσεις **NP**, **coNP**. Υποθέτοντας την ύπαρξη τέτοιων γλωσσών, θα κατασκευάσουμε αποδοτικούς ντετερμινιστικούς αλγορίθμους που αποφασίζουν το **SAT** ψαλιδίζοντας το δέντρο αυτο-αναγωγής του, καταλήγοντας έτσι στο συμπέρασμα $\mathbf{P} = \mathbf{NP}$ που θεωρείται ευρέως αναληθές χωρίς ωστόσο να έχει αποδειχθεί. Επεκτείνοντας αυτήν την τεχνική του ψαλιδίσματος, στην ενότητα 2.3 θα δούμε ότι η κλάση **NP** δεν έχει πλήρη αραιά σύνολα, εκτός αν $\mathbf{P} = \mathbf{NP}$. Για την απόδειξη αυτού του αποτελέσματος θα χρησιμοποιήσουμε την έννοια του αριστερού συνόλου.

2.1 Εισαγωγικά

2.1.1 Υπολογισμοί με μαντείο

Ορισμός 2.1.1 *Η μηχανή Turing με μαντείο (OTM) είναι μία μηχανή Turing με πολλαπλές ταινίες, μία από τις οποίες είναι η ειδική* ταινία ερωτήσεων *προς το μαντείο, και τρεις ξεχωριστές καταστάσεις query, yes, no.*

Το μαντείο είναι ένα προκαθορισμένο, για κάθε μηχανή, σύνολο στο οποίο η μηχανή μπορεί ανά πάσα στιγμή να γνωρίζει αν μία δεδομένη συμβολοακολουθία ανήκει ή όχι. Συγκεκριμένα, έστω ότι μία μηχανή έχει μαντείο A, και ότι κάποια στιγμή κατά τη διάρκεια του υπολογισμού μεταβαίνει στην κατάσταση query, ενώ η ταινία ερωτήσεων περιέχει τη συμβολοακολουθία w. Τότε η επόμενη μετάβαση θα είναι προς την κατάσταση yes, αν $w \in A$, ή προς την κατάσταση no, αν $w \notin A$. Θεωρούμε ότι τα περιεχόμενα της ταινίας ερωτήσεων σβήνονται ακαριαία κατά τη μετάβαση στην κατάσταση yes ή no, καθώς και ότι οι υπόλοιπες ταινίες και οι θέσεις των κεφαλών παραμένουν αμετάβλητες. Αν M είναι μία OTM με μαντείο A, συμβολίζουμε με $L(M, A)$ τη γλώσσα που αποφασίζει η μηχανή αυτή.

Μία ντετερμινιστική μηχανή Turing με μαντείο μπορεί να εξομοιώσει τη λειτουργία μίας μη ντετερμινιστικής Turing χωρίς μαντείο. Αν M είναι μία NTM, θεωρούμε την OTM M_0 με μαντείο A, το οποίο περιέχει ως στοιχεία όλες τις νόμιμες ακολουθίες στιγμιοτύπων της M που οδηγούν σε κατάσταση αποδοχής, καθώς και όλα τα αρχικά τμήματά τους. Η M_0 εξομοιώνει τη λειτουργία της M σε είσοδο x με τον αλγόριθμο 1. Η M_0 χρησιμοποιεί το μαντείο A ώστε να επιλέγει, αναμέσα σε όλες τις δυνατές μεταβάσεις της M σε κάθε βήμα υπολογισμού, εκείνη που είναι εγγυημένο ότι θα οδηγήσει σε κατάσταση αποδοχής. Για την ορθότητα του αλγορίθμου 1, παρατηρούμε τα εξής:

- Αν $x \in L(M)$, τότε εξ ορισμού υπάρχει τουλάχιστον μία έγκυρη ακολουθία στιγμιοτύπων της M που καταλήγει σε κατάσταση αποδοχής. Άρα η συνθήκη της γραμμής 1.9 δεν θα ικανοποιηθεί ποτέ, ενώ όταν η c

Αλγόριθμος 1 Εξομοίωση της NTM M από την OTM M_0

Είσοδος: x

1: $s \leftarrow$ αρχικό στιγμιότυπο της M με είσοδο x ▶ s: τρέχον στιγμιότυπο
2: $c \leftarrow \varepsilon$ ▶ c: μέχρι στιγμής ακολουθία στιγμιοτύπων
3: **loop**
4:     **if** η κατάσταση του s είναι η κατάσταση αποδοχής **then**
5:         τερμάτισε την εκτέλεση σε κατάσταση αποδοχής
6:     **end if**
7:     $c \leftarrow cs$
8:     $R \leftarrow \{t : t$ είναι έγκυρο επόμενο στιγμιότυπο του $s\}$
9:     **if** $\forall t \in R(ct \notin A)$ **then** ▶ ερωτήσεις στο μαντείο
10:         τερμάτισε την εκτέλεση σε κατάσταση απόρριψης
11:     **else if** $\exists t \in R(ct \in A)$ **then**
12:         $s \leftarrow t$
13:     **end if**
14: **end loop**

περιέχει ολόκληρη την ακολουθία αυτή, τότε θα ικανοποιηθεί η συνθήκη της γραμμής 1.4 και η M_0 θα αποδεχθεί το x.

- Αντίθετα, αν $x \notin L(M)$ τότε δεν υπάρχει ακολουθία στιγμιοτύπων της M που να καταλήγει σε κατάσταση αποδοχής, δηλαδή το μαντείο A της M_0 είναι το κενό σύνολο. Άρα η συνθήκη της γραμμής 1.9 θα ικανοποιηθεί αμέσως και η M_0 θα απορρίψει το x.

Προκύπτει λοιπόν ότι οι M, M_0 αποδέχονται ακριβώς την ίδια γλώσσα.

Η προσθήκη του μαντείου όμως ενδυναμώνει το μοντέλο της μηχανής Turing κατά πολύ περισσότερο από την άρση του περιορισμού του ντετερμινισμού. Αρκεί να σκεφτεί κανείς ότι θα μπορούσαμε να έχουμε θεωρήσει μία OTM M_1 με μαντείο το ίδιο το σύνολο $L(M)$. Τότε η εξομοίωση της M από την M_1 θα εκφυλιζόταν στην απλή αντιγραφή της εισόδου στην ταινία ερωτήσεων προς το μαντείο, και στην αποδοχή αν $x \in L(M)$ ή στην απόρριψη αν $x \notin L(M)$. Γενικά, μία OTM μπορεί να αποφασίσει οποιοδήποτε σύνολο R, αν ως μαντείο θεωρήσουμε το ίδιο το σύνολο R. Επομένως η εξομοίωση μίας OTM από μία NTM δεν είναι εφικτή στη γενική περίπτωση.

Για τους υπολογισμούς με μαντείο εισάγουμε τους ακόλουθους συμβολισμούς:

Ορισμός 2.1.2 *Αν $\mathcal{C}$ είναι μία κλάση πολυπλοκότητας που ορίζεται με περιορισμό χρόνου και A είναι ένα σύνολο, ορίζεται η κλάση πολυπλοκότητας $\mathcal{C}^A$ ως η κλάση που περιέχει τις γλώσσες που αποφασίζονται με μηχανές Turing που έχουν τους ίδιους περιορισμούς χρόνου όπως η $\mathcal{C}$, αλλά μπορούν να χρησιμοποιούν το σύνολο A ως μαντείο.*

Ορισμός 2.1.3 *Αν $\mathcal{C}$ είναι μία κλάση πολυπλοκότητας που ορίζεται με περιορισμό χρόνου και $\mathcal{D}$ είναι μία οποιαδήποτε κλάση γλωσσών, τότε:*

$$\mathcal{C}^{\mathcal{D}} = \bigcup_{A \in \mathcal{D}} \mathcal{C}^A$$

2.1.2 Ορισμός της αυτο-αναγωγής

Ορισμός 2.1.4 *Ένα σύνολο A έχει την ιδιότητα της* αυτο-αναγωγής *αν και μόνο αν υπάρχει μία ντετερμινιστική OTM πολυωνυμικού χρόνου M, ώστε να ισχύουν:*

1. *$A = L(M, A)$.*

2. *Για κάθε είσοδο μήκους n, η M θέτει στο μαντείο ερωτήσεις για συμβολοσειρές μήκους το πολύ $n - 1$.*

Η πρώτη συνθήκη που απαιτούμε υποδηλώνει ότι το σύνολο A είναι σταθερό σημείο της συνάρτησης $L(M, \cdot)$. Μάλιστα, χρησιμοποιώντας τη δεύτερη συνθήκη μπορούμε να αποδείξουμε ότι αυτό το σταθερό σημείο είναι και μοναδικό.

Θεώρημα 2.1.5 *Αν το σύνολο A έχει την ιδιότητα της αυτο-αναγωγής, M είναι η OTM της οποίας η ύπαρξη προκύπτει εξ ορισμού της ιδιότητας αυτής και B είναι ένα σύνολο για το οποίο ισχύει $B = L(M, B)$, τότε $A = B$.*

Απόδειξη

Χωρίς βλάβη της γενικότητας υποθέτουμε ότι οι A, B είναι γλώσσες επί ενός κοινού αλφαβήτου Σ. Θα αποδείξουμε ότι

$$\forall n \in \mathbb{N}\, \forall x \in \Sigma^n \, (x \in A \iff x \in B)$$

Η απόδειξη θα γίνει με επαγωγή στο n.

Για $n = 0$, αρκεί να δείξουμε ότι $\varepsilon \in A \iff \varepsilon \in B$. Επειδή $|\varepsilon| = 0$, η M δεν θέτει καμία ερώτηση στο μαντείο. Επομένως αποδέχεται ή απορρίπτει το ε ανεξάρτητα από το μαντείο: $\varepsilon \in L(M, A) \iff \varepsilon \in L(M, B)$, και επειδή $A = L(M, A)$, $B = L(M, B)$, παίρνουμε το ζητούμενο: $\varepsilon \in A \iff \varepsilon \in B$.

Έστω ότι για κάθε $n < k$ ισχύει ότι:

$$\forall x \in \Sigma^n \, (x \in A \iff x \in B) \tag{2.1}$$

Θα δείξουμε ότι $\forall x \in \Sigma^k \, (x \in A \iff x \in B)$. Κατά τον υπολογισμό της με είσοδο $x \in \Sigma^k$, η M θέτει στο μαντείο ερωτήσεις με μήκος το πολύ $k - 1$. Ας είναι $a_1, a_2, \ldots$ η ακολουθία των ερωτήσεων της M κατά τον υπολογισμό με είσοδο x και μαντείο A, $b_1, b_2, \ldots$ η ακολουθία ερωτήσεων της M κατά τον υπολογισμό με είσοδο x και μαντείο B, και j ο μικρότερος δείκτης για τον οποίο ισχύει $a_j \neq b_j$. Εφόσον οι ερωτήσεις με δείκτη $< j$ προς τα δύο μαντεία είναι ίδιες και έχουν μήκος $< k$, ισχύει για αυτές η επαγωγική υπόθεση 2.1 και κατά συνέπεια οι απαντήσεις που θα λάβει η M είναι ίδιες. Επιπρόσθετα, οι κανόνες μεταβάσεων της M είναι ίδιες και για τις δύο εκτελέσεις, άρα η μηχανή θα βρίσκεται ακριβώς στο ίδιο στιγμιότυπο και στις δύο εκτελέσεις, προτού θέσει την j-οστή ερώτηση. Άρα δεν μπορεί να ισχύει $a_j \neq b_j$. Προκύπτει λοιπόν ότι η M θέτει την ίδια ακολουθία ερωτήσεων προς τα δύο μαντεία, λαμβάνει τις ίδιες απαντήσεις λόγω της επαγωγικής υπόθεσης, και κατά συνέπεια αποδέχεται ή απορρίπτει το x και στις δύο εκτελέσεις.

Αποδείξαμε ότι $\forall n \in \mathbb{N} \forall x \in \Sigma^n \, (x \in A \iff x \in B)$, από το οποίο έπεται άμεσα ότι $A = B$.

■

Για τα σύνολα που έχουν την ιδιότητα της αυτο-αναγωγής, ένα ερώτημα της μορφής "$x \in A$;" με $|x| = n$ ανάγεται σε ένα πλήθος ερωτημάτων της μορφής "$w \in A$;" με $|w| < n$. Αντί να αναθέσουμε σε ένα μαντείο την απάντηση σε αυτά τα "μικρότερα" προβλήματα, μπορούμε να θεωρήσουμε μία μηχανή Turing η οποία χρησιμοποιεί τους δικούς της πόρους για να δώσει απάντηση σε αυτά και κατ'επέκταση στο αρχικό ερώτημα. Συγκεκριμένα, ας υποθέσουμε ότι το σύνολο A έχει την ιδιότητα της αυτο-αναγωγής. Εξ ορισμού της ιδιότητας αυτής, υπάρχει μία OTM πολυωνυμικού χρόνου M που θέτει στο μαντείο ερωτήσεις με μήκος αυστηρά μικρότερο του μήκους της εισόδου της και αποδέχεται το A χρησιμοποιώντας το A ως μαντείο. Θεωρούμε την DTM M' που εξομοιώνει τη λειτουργία της M, με μία διαφορά: κάθε φορά που η M θέτει μία ερώτηση w στο μαντείο, η M' αποθηκεύει το στιγμιότυπο στο οποίο βρίσκεται σε μία στοίβα και ξεκινάει από την αρχή με είσοδο w για να υπολογίσει αν $w \in A$ ή όχι. Όταν ολοκληρωθεί ο υπολογισμός, η μηχανη εισέρχεται στην κατάσταση yes αν $w \in A$ ή στην κατάσταση no αν $w \notin A$ και ανασύρει από τη στοίβα το στιγμιότυπο από το οποίο θα συνεχίσει την εκτέλεσή της. Πρόκειται για μία αναδρομική διαδικασία στην οποία απαιτείται χώρος τόσο για τη στοίβα όσο και για τον καθεαυτό υπολογισμό της μηχανής. Επειδή η M είναι πολυωνυμικού χρόνου, για κάθε επιμέρους υπολογισμό χρειάζεται το πολύ πολυωνυμικός χώρος. Επίσης, το μέγεθος της στοίβας δεν πρόκειται να ξεπεράσει τα $|x|$ στιγμιότυπα, όπου x η αρχική είσοδος. Άρα για τον υπολογισμό της M' αρκεί συνολικά πολυωνυμικός χώρος. Καταλήγουμε λοιπόν στο:

Θεώρημα 2.1.6 *Αν το σύνολο A έχει την ιδιότητα της αυτο-αναγωγής, τότε $A \in$ **PSPACE**.*

2.1.3 Αραιά σύνολα

Στην παράγραφο 1.3 ορίσαμε τις γλώσσες ως υποσύνολα του σύνολου των πεπερασμένων ακολουθιών που προκύπτουν από τα σύμβολα ενός αλφαβήτου Σ. Σε κάθε γλώσσα L αντιστοιχεί μία *συνάρτηση πυκνότητας* $\mathrm{dens}_L : \mathbb{N} \to \mathbb{N}$ (census/density function). Για κάθε φυσικό αριθμό n, η τιμή $\mathrm{dens}_L(n)$ είναι το πλήθος των συμβολοσειρών που ανήκουν στην L και έχουν μήκος το πολύ n:

$$\mathrm{dens}_L(n) = |\{x \in L : |x| \leq n\}|$$

Η πυκνότητα μίας γλώσσας δεν μπορεί να αυξάνει με ρυθμό μεγαλύτερο απο εκθετικό, αφού η γλώσσα περιέχει το πολύ k^n συμβολοσειρές μήκους n, όπου k το πλήθος των συμβόλων του αλφαβήτου. Αν η πυκνότητα μίας γλώσσας φράσσεται άνω από κάποιο πολυώνυμο, η γλώσσα ονομάζεται *αραιή*:

Ορισμός 2.1.7 *Μία γλώσσα L ονομάζεται* αραιή *αν και μόνο αν υπάρχει ένα πολυώνυμο $p(n)$ ώστε $\mathrm{dens}_L(n) \leq p(n)$, για κάθε φυσικό n.*

Σε ορισμένες περιπτώσεις είναι χρήσιμο να θεωρούμε τις αραιές γλώσσες ως τις γλώσσες που περιέχουν πολυωνυμικό πλήθος συμβολοσειρών για κάθε δοσμένο μήκος. Αποδεικνύεται εύκολα ότι οι δύο ορισμοί είναι ισοδύναμοι.

Θεώρημα 2.1.8 *Μία γλώσσα S είναι αραιή αν και μόνο αν υπάρχει ένα πολυώνυμο $p(n)$ ώστε $|S^{=n}| \leq p(n)$, για κάθε φυσικό n* [1].

Απόδειξη

Έστω ότι η γλώσσα S είναι αραιή. Ισχύει $S^{=n} \subseteq S^{\leq n} \Rightarrow |S^{=n}| \leq \left|S^{\leq n}\right| \Rightarrow |S^{=n}| \leq \mathrm{dens}_S(n)$. Επειδή η S είναι αραιή, η dens_S είναι πολυωνυμική συνάρτηση.

Αντίστροφα, έστω ότι υπάρχει ένα πολυώνυμο $p(n) = \sum_{i=0}^{k} a_i n^i$, ώστε για κάθε φυσικό n να ισχύει $|S^{=n}| \leq p(n)$. Έχουμε ότι $S^{\leq n} = \bigcup_{j=0}^{n} S^{=j} \Rightarrow \left|S^{\leq n}\right| \leq \sum_{j=0}^{n} |S^{=j}| \leq \sum_{j=0}^{n} p(j) = \sum_{j=0}^{n} \sum_{i=0}^{k} a_i j^i = \sum_{i=0}^{k} a_i \sum_{j=0}^{n} j^i \leq \sum_{i=0}^{k} |a_i| \sum_{j=0}^{n} n^i = (n+1) \sum_{i=0}^{k} |a_i| n^i = (n+1)p'(n)$, όπου το $p'(n)$ έχει ως συντελεστές τις απόλυτες τιμές των συντελεστών του $p(n)$.

■

Υποκατηγορία των αραιών γλωσσών είναι οι *μονοσυμβολικές* (tally) γλώσσες, οι οποίες περιέχουν συμβολοσειρές που αποτελούνται από επαναλήψεις ενός μοναδικού συμβόλου:

[1] Το σύνολο $S^{=n}$ είναι το σύνολο των στοιχείων του S που έχουν μήκος n. Συμβολίζουμε επίσης με $S^{\leq n}$ το σύνολο των στοιχείων του S που έχουν μήκος το πολύ n. Σύμφωνα με αυτόν το συμβολισμό, $\mathrm{dens}_S(n) = \left|S^{\leq n}\right|$.

Ορισμός 2.1.9 *Μία γλώσσα ονομάζεται* μονοσυμβολική *αν και μόνο αν είναι υποσύνολο του* $\{\alpha\}^\star$*, όπου* α *είναι κάποιο σύμβολο.*

Είναι εύκολο να δει κανείς ότι αν η A είναι μονοσυμβολική γλώσσα, τότε $\mathrm{dens}_A(n) \leq n$. Κατά συνέπεια οι μονοσυμβολικές γλώσσες είναι αραιές. Οι πιο πυκνές γλώσσες είναι αυτές που η συνάρτηση πυκνότητάς τους αυξάνει εκθετικά. Για παράδειγμα, αν έχουμε ένα αλφάβητο Σ με $|\Sigma| = k$, η συνάρτηση πυκνότητας της γλώσσας που περιλαμβάνει όλες τις πεπερασμένες συμβολοακολουθίες επί αυτού του αλφαβήτου είναι:

$$\mathrm{dens}_{\Sigma^\star}(n) = \sum_{i=0}^{n} k^i = \frac{k^{n+1}-1}{k-1}$$

Από μία άποψη, οι πιο αραιές γλώσσες είναι οι πεπερασμένες γλώσσες, αφού η συνάρτηση πυκνότητας οποιασδήποτε πεπερασμένης γλώσσας φράσσεται άνω από κάποια σταθερά.

Οι αραιές γλώσσες είναι αρκετά "απλές" ώστε να είναι εξαιρετικά απίθανο κάποια από αυτές να είναι **NP**-πλήρης. Στις επόμενες παραγράφους θα δούμε ορισμένα αποτελέσματα που συσχετίζουν τη δυσκολία των αραιών συνόλων με το ερώτημα $\mathbf{P} \stackrel{?}{=} \mathbf{NP}$. Οι αποδείξεις αυτών των αποτελεσμάτων επιδεικνύουν ενδιαφέρουσες αλγοριθμικές τεχνικές που εκμεταλλεύονται την ιδιότητα της αυτο-αναγωγής που διαθέτει το πρόβλημα **SAT**.

2.2 Η τεχνική του ψαλιδίσματος

Στο πρόβλημα **SAT**, δίνεται μία λογική έκφραση ϕ με n ελεύθερες μεταβλητές και τίθεται το ερώτημα αν υπάρχει ανάθεση τιμών στις μεταβλητές αυτές ώστε η έκφραση να είναι αληθής. Αν x είναι μία από τις ελεύθερες μεταβλητές της ϕ, τότε οι εκφράσεις $\phi\,[x := 1]$ και $\phi\,[x := 0]$ έχουν ακριβώς $n-1$ ελεύθερες μεταβλητές. Επίσης, είναι προφανές ότι αν η ανάθεση u ικανοποιεί την $\phi\,[x := \mathrm{v}]$, όπου $\mathrm{v} \in \{0,1\}$, τότε και μόνο τότε η ανάθεση $u \cup \{(x,\mathrm{v})\}$ θα ικανοποιεί την ϕ. Η OTM με μαντείο **SAT** που περιγράφεται από τον αλγόριθμο 2 ικανοποιεί τις απαιτήσεις του ορισμού 2.1.4, εφόσον είναι εύκολο να επινοήσει κανείς ένα σχήμα κωδικοποίησης λογικών εκφράσεων ώστε οι εκφράσεις

Αλγόριθμος 2 Αλγόριθμος πολυωνυμικού χρόνου που αποφασίζει το **SAT**, με μαντείο **SAT** και ερωτήσεις με μικρότερο μήκος από την είσοδο

Είσοδος: ϕ

```
if η φ δεν έχει μεταβλητές then
    αποτίμησε την φ και αποδέξου αν η τιμή είναι 1, αλλιώς απόρριψε
end if
έστω x η πρώτη μεταβλητή της φ
if φ[x := 0] ∈ SAT ή φ[x := 1] ∈ SAT then αποδέξου
else απόρριψε
end if
```

$\phi\,[x := 1]$ και $\phi\,[x := 0]$ να έχουν μικρότερο μήκος από την ϕ. Κατά συνέπεια το **SAT** έχει την ιδιότητα της αυτο-αναγωγής, γεγονός το οποίο αναφέρει για πρώτη φορά ο Schnorr στο [Sch76].

2.2.1 NP-δύσκολες μονοσυμβολικές γλώσσες

Θεώρημα 2.2.1 *[Ber78] Αν υπάρχει μονοσυμβολική γλώσσα που είναι* **NP**-*δύσκολη, τότε* $\mathbf{P} = \mathbf{NP}$.

Απόδειξη

Έστω T μία μονοσυμβολική **NP**-δύσκολη γλώσσα. Κάθε πρόβλημα της κλάσης **NP** ανάγεται στην T, συνεπώς και $\mathbf{SAT} \leq^{\mathrm{p}}_{\mathrm{m}} T$. Υπάρχει δηλαδή μία συνάρτηση g, υπολογιστή σε ντετερμινιστικό πολυωνυμικό χρόνο, τέτοια ώστε:

$$(\forall x)\, x \in \mathbf{SAT} \iff g(x) \in T$$

Με τις προϋποθέσεις του θεωρήματος, δηλαδή ότι υπάρχει η γλώσσα T και η συνάρτηση g, ο αλγόριθμος 3 αποφασίζει το **SAT**. Θα δείξουμε στη συνέχεια αφενός την ορθότητα του αλγορίθμου και αφετέρου ότι είναι αλγόριθμος πολυωνυμικού χρόνου, από το οποίο έπεται ότι $\mathbf{P} = \mathbf{NP}$.

Αρχικά παρατηρούμε ότι αν για μία έκφραση ψ ισχύει ότι $g(\psi) \notin \{1\}^\star$, τότε $\psi \notin \mathbf{SAT}$, διότι σε αντίθετη περίπτωση θα είχαμε $\psi \in \mathbf{SAT} \Rightarrow g(\psi) \in T \Rightarrow$

Αλγόριθμος 3 Ντετερμινιστικός αλγόριθμος πολυωνυμικού χρόνου για το **SAT**, υπό τις προϋποθέσεις του θεωρήματος 2.2.1

Είσοδος: λογική έκφραση ϕ με μεταβλητές $v_1, v_2, \ldots, v_m$, $m \geq 0$

$C \leftarrow \{\phi\}$
for $i \leftarrow 1$ **to** m **do**
    $S \leftarrow \emptyset$
    for all $\psi \in C$ **do**
        $S \leftarrow S \cup \{\psi\,[v_i := 1]\,, \psi\,[v_i := 0]\}$
    end for
    $C \leftarrow \emptyset$
    for all $\psi \in S$ **do**
        if $g(\psi) \in \{1\}^\star$ και δεν υπάρχει άλλη έκφραση $\psi' \in C$ με $g(\psi') = g(\psi)$ **then**
            $C \leftarrow C \cup \{\psi\}$
        end if
    end for
end for
αν υπάρχει έκφραση $\psi \in C$ με τιμή 1 τότε αποδέξου, αλλιώς απόρριψε

$g(\psi) \in \{1\}^\star$. Επίσης, αν για δύο εκφράσεις ψ, ψ' ισχύει $g(\psi') = g(\psi)$, τότε είτε και οι δύο είναι ικανοποιήσιμες, είτε και οι δύο είναι μη ικανοποιήσιμες. Πραγματικά, $\psi \in \mathbf{SAT} \Leftrightarrow g(\psi) \in T \Leftrightarrow g(\psi') \in T \Leftrightarrow \psi' \in \mathbf{SAT}$.

Θεωρούμε μία εκτέλεση του αλγορίθμου με είσοδο ϕ, και τις ακολουθίες $\{C_i\}_{0\leq i\leq m}$, $\{S_i\}_{1\leq i\leq m}$, όπου m το πλήθος των μεταβλητών της ϕ, $C_0 = \{\phi\}$, C_i είναι το σύνολο C που προκύπτει στο τέλος της i-οστής επανάληψης του εξωτερικού βρόχου for της γραμμής 3.2, και S_i είναι το σύνολο S που αντιστοιχεί στην ίδια επανάληψη. Προφανώς $\phi \in \mathbf{SAT}$ αν και μόνο αν το σύνολο C_0 περιέχει ικανοποιήσιμη έκφραση. Θα δείξουμε ότι το C_i περιέχει ικανοποιήσιμη έκφραση αν και μόνο αν το C_{i+1} περιέχει ικανοποιήσιμη έκφραση.

Λόγω της ιδιότητας της αυτο-αναγωγής του $\mathbf{SAT}$, αν το C_{i+1} περιέχει μία ικανοποιήσιμη έκφραση $\psi\,[x := v]$, τότε το C_i περιέχει την ικανοποιήσιμη έκφραση ψ. Αντίστροφα, έστω ότι $\psi \in C_i$ και η ψ είναι ικανοποιήσιμη. Οι εκφράσεις $\psi_0 \equiv \psi\,[v_{i+1} := 0]$ και $\psi_1 \equiv \psi\,[v_{i+1} := 1]$, από τις οποίες τουλάχιστον η μία είναι ικανοποιήσιμη, θα περιέχονται στο S_{i+1}. Διακρίνουμε περιπτώσεις:

- Αν η ψ_0 είναι ικανοποιήσιμη, στην επανάληψη του βρόχου for 3.8 που αντιστοιχεί στην ψ_0 θα έχουμε $g(\psi_0) \in \{1\}^\star$ (αφού $\psi_0 \in \mathbf{SAT} \Rightarrow g(\psi_0) \in T$), και αν δεν υπάρχει άλλη έκφραση $\psi' \in C_{i+1}$ με $g(\psi') = g(\psi_0)$, η ψ_0 θα εισαχθεί στο C_{i+1}. Αν υπάρχει έκφραση $\psi' \in C_{i+1}$ με $g(\psi') = g(\psi_0)$, τότε η ψ_0 δεν θα περιληφθεί στο C_{i+1} όμως για την ψ' ισχύει $g(\psi') = g(\psi_0) \in T \Rightarrow \psi' \in \mathbf{SAT}$. Επομένως σε κάθε περίπτωση θα υπάρχει στο C_{i+1} μία ικανοποιήσιμη έκφραση.

- Αν η ψ_0 δεν είναι ικανοποιήσιμη, τότε η ψ_1 πρέπει να είναι ικανοποιήσιμη. Όπως και προηγουμένως, στην επανάληψη του βρόχου for 3.8 που αντιστοιχεί στην ψ_1, είτε η ψ_1 θα περιληφθεί στο C_{i+1}, είτε θα προκύψει ότι υπάρχει ήδη στο C_{i+1} κάποια ικανοποιήσιμη έκφραση.

Για κάθε i, το σύνολο C_i περιέχει εκφράσεις με ακριβώς $m - i$ μεταβλητές. Αν η είσοδος είναι ικανοποιήσιμη, τότε και μόνο τότε το C_m θα περιέχει τουλάχιστον μία ικανοποιήσιμη έκφραση χωρίς μεταβλητές, δηλαδή τουλάχιστον μία έκφραση που αποτιμάται στην αληθοτιμή 1. Αυτό ακριβώς ελέγχει ο αλγόριθμος στο βήμα 3.14. Άρα η ορθότητα αποδείχτηκε και απομένει να δειχτεί ότι πρόκειται για αλγόριθμο πολυωνυμικού χρόνου.

Η συνάρτηση g είναι υπολογιστή σε πολυωνυμικό χρόνο. Η μηχανή Turing που την υπολογίζει μπορεί να γράφει στην έξοδο το πολύ ένα σύμβολο ανά μετάβαση, συνεπώς υπάρχει ένα πολυώνυμο p ώστε για κάθε x να ισχύει $|g(x)| \leq p(|x|)$. Κάθε ένα από τα σύνολα $\{C_i\}_{1 \leq i \leq m}$ περιέχει εκφράσεις με μήκος το πολύ $|\phi|$ οι οποίες αντιστοιχίζονται μέσω της g σε διαφορετικά στοιχεία του $\{1\}^\star$, τα οποία κατά συνέπεια έχουν μήκος το πολύ $p(|\phi|)$. Επομένως το πλήθος τους είναι το πολύ $\mathrm{dens}_{\{1\}^\star}(p(|\phi|)) \leq p(|\phi|)$. Εφόσον το σύνολο C περιέχει σε κάθε επανάληψη το πολύ πολυωνυμικό (ως προς το $|\phi|$) πλήθος εκφράσεων, και επειδή η συνάρτηση g είναι υπολογιστή σε πολυωνυμικό χρόνο, προκύπτει ότι ολόκληρος ο αλγόριθμος εκτελείται σε πολυωνυμικό χρόνο.

■

Πόρισμα 2.2.2 *[Ber78] Αν υπάρχει μονοσυμβολική γλώσσα που είναι* **NP***-πλήρης, τότε* $\mathbf{P} = \mathbf{NP}$.

2.2.2 coNP-δύσκολες αραιές γλώσσες

Ορισμός 2.2.3 $\mathbf{coNP} = \left\{L : \overline{L} \in \mathbf{NP}\right\}$

Δηλαδή οι γλώσσες που ανήκουν στην κλάση πολυπλοκότητας **coNP** είναι ακριβώς αυτές που τα συμπληρώματά τους ανήκουν στην **NP**. Γενικότερα, αν $\mathcal{C}$ είναι μία κλάση πολυπλοκότητας, η κλάση $\mathbf{co}\mathcal{C}$ είναι η κλάση που περιέχει ακριβώς εκείνες τις γλώσσες που τα συμπληρώματά τους ανήκουν στην $\mathcal{C}$.

Θεώρημα 2.2.4 *[For79] Αν υπάρχει αραιή γλώσσα που είναι* **coNP***-δύσκολη, τότε* $\mathbf{P} = \mathbf{NP}$.

Απόδειξη

Έστω S μία αραιή **coNP**-δύσκολη γλώσσα, και p_d το πολυώνυμο που φράσσει την πυκνότητα της S: $(\forall n)\ \mathrm{dens}_S(n) \leq p_d(n)$. Γνωρίζουμε ότι

Αλγόριθμος 4 Ντετερμινιστικός αλγόριθμος πολυωνυμικού χρόνου για το **SAT**, υπό τις προϋποθέσεις του θεωρήματος 2.2.4

Είσοδος: λογική έκφραση ϕ με μεταβλητές $v_1, v_2, \ldots, v_m$, $m \geq 0$

$C \leftarrow \{\phi\}$
for $i \leftarrow 1$ **to** m **do**
    $R \leftarrow \emptyset$
    for all $\psi \in C$ **do**
        $R \leftarrow R \cup \{\psi\,[v_i := 1]\,, \psi\,[v_i := 0]\}$
    end for
    $C \leftarrow \emptyset$
    for all $\psi \in R$ **do**
        if δεν υπάρχει άλλη έκφραση $\psi' \in C$ με $g(\psi') = g(\psi)$ **then**
            $C \leftarrow C \cup \{\psi\}$
        end if
    end for
    if $|C| \geq p_d\,(p_k\,(|\phi|)) + 1$ **then**
        τερμάτισε την εκτέλεση σε κατάσταση αποδοχής
    end if
end for
αν υπάρχει έκφραση $\psi \in C$ με τιμή 1 τότε αποδέξου, αλλιώς απόρριψε

SAT $\in$ **NP**, άρα ισχύει ότι $\overline{\textbf{SAT}} \in$ **coNP** και $\overline{\textbf{SAT}} \leq^{\mathrm{p}}_{\mathrm{m}} S$. Έστω g η σχετική συνάρτηση αναγωγής, για την οποία ισχύει ότι είναι υπολογιστή σε ντετερμινιστικό πολυωνυμικό χρόνο και:

$$(\forall x)\, x \in \overline{\textbf{SAT}} \iff g(x) \in S$$

Επειδή η g υπολογίζεται σε πολυωνυμικό χρόνο, θα υπάρχει ένα πολυώνυμο $p_k(n)$ ώστε $(\forall x)\ |g(x)| \leq p_k(|x|)$, όπως και στην απόδειξη του θεωρήματος 2.2.1. Υπό αυτές τις προϋποθέσεις, ο αλγόριθμος 4 αποφασίζει το **SAT** σε ντετερμινιστικό πολυωνυμικό χρόνο, κατά συνέπεια $\mathbf{P} = \mathbf{NP}$.

Πρόκειται σαφώς για αλγόριθμο πολυωνυμικού χρόνου, αφού σε κάθε επανάληψη το μέγεθος του συνόλου C δεν ξεπερνάει την τιμή $p_d(p_k(|\phi|))$. Όσον αφορά την ορθότητα, αν η συνθήκη της γραμμής 4.13 δεν ικανοποιηθεί σε καμία επανάληψη τότε ο αλγόριθμος αποφασίζει το **SAT** για τους ίδιους λόγους με τον αλγόριθμο 3.

Αν σε κάποια επανάληψη ικανοποιηθεί η συνθήκη της γραμμής 4.13, τότε υπάρχουν τουλάχιστον $p_d(p_k(|\phi|))+1$ διαφορετικές μεταξύ τους συμβολοακολουθίες, στις οποίες απεικονίζονται μέσω της g οι εκφράσεις του συνόλου C εκείνης της επανάληψης. Όμως οι εκφράσεις αυτές έχουν μήκος το πολύ $|\phi|$, αρά οι συμβολοακολουθίες στις οποίες απεικονίζονται έχουν μήκος το πολύ $p_k(|\phi|)$. Επειδή το σύνολο S είναι αραιό, περιέχει το πολύ $p_d(p_k(|\phi|))$ συμβολοακολουθίες μήκους μέχρι $p_k(|\phi|)$. Συμπεραίνουμε λοιπόν ότι υπάρχει τουλάχιστον μία έκφραση $\psi \in C$ που απεικονίζεται σε συμβολοακολουθία που δεν ανήκει στο S, από το οποίο συνεπάγεται: $g(\psi) \notin S \Rightarrow \psi \notin \overline{\mathbf{SAT}} \Rightarrow \psi \in \mathbf{SAT}$. Επειδή η ψ προέκυψε από την ϕ αντικαθιστώντας κάποιες μεταβλητές με αληθοτιμές, συμπεραίνουμε ότι και $\phi \in \mathbf{SAT}$. Άρα ο αλγόριθμος ορθά αποδέχεται την είσοδο σε αυτήν την περίπτωση.

■

Πόρισμα 2.2.5 *[For79] Αν υπάρχει αραιή γλώσσα που είναι* **coNP***-πλήρης, τότε* $\mathbf{P} = \mathbf{NP}$.

2.2.3 Παρατηρήσεις και άλλες εφαρμογές της τεχνικής του ψαλιδισμού

Για κάθε στιγμιότυπο I ενός προβλήματος απόφασης Π που έχει την ιδιότητα της αυτο-αναγωγής, θεωρούμε το αντίστοιχο *δέντρο αυτο-αναγωγής*. Πρόκειται για ένα δέντρο του οποίου οι κόμβοι αντιστοιχούν σε στιγμιότυπα του Π. Ένας κόμβος v' είναι παιδί του κόμβου v αν και μόνο αν το στιγμιότυπο v' είναι μία απο τις ερωτήσεις που θέτει στο μαντείο η OTM που αποφασίζει το Π (βάσει του ορισμού της αυτο-αναγωγής), αν ξεκινήσει τον υπολογισμό της με είσοδο v. Τα *διαζευκτικά* δέντρα αυτο-αναγωγής είναι μία ειδική κατηγορία τέτοιων δέντρων, στα οποία η απάντηση στο στιγμιότυπο που αντιστοιχεί σε κάποιον κόμβο είναι "ναι" αν και μόνο αν υπάρχει ένα παιδί του στο αντίστοιχο στιγμιότυπο του οποίου η απάντηση είναι "ναι". Με την ίδια λογική ορίζονται και τα *συζευκτικά* δέντρα αυτο-αναγωγής, στα οποία η απάντηση στο στιγμιότυπο κάποιου κόμβου είναι "ναι" αν και μόνο αν η απάντηση για όλα τα παιδιά του είναι "ναι".

Για το πρόβλημα **SAT**, το δέντρο αυτο-αναγωγής που προκύπτει για κάθε λογική έκφραση ϕ είναι διαζευκτικό. Πραγματικά, για να είναι ικανοποιήσιμη η ϕ πρέπει είτε η $\phi\,[x := 1]$ να είναι ικανοποιήσιμη, είτε η $\phi\,[x := 0]$ να είναι ικανοποιήσιμη, όπου x είναι κάποια μεταβλητή της ϕ. Κάθε επίπεδο του δέντρου περιέχει εκφράσεις με μία λιγότερη μεταβλητή από το προηγούμενο επίπεδο. Κάθε ένα από τα μονοπάτια του δέντρου από τη ρίζα προς τα φύλλα αντιστοιχεί σε μία δυνατή ανάθεση τιμών στις μεταβλητές της έκφρασης της ρίζας. Κάθε φύλλο είναι μία έκφραση χωρίς μεταβλητές, που αποτιμώμενη δίνει την τιμή της ρίζας με την ανάθεση τιμών που αντιστοιχεί στο μονοπάτι από τη ρίζα στο συγκεκριμένο φύλλο. Αν έστω και ένα από τα φύλλα του δέντρου αποτιμάται στην τιμή 1, τότε η έκφραση της ρίζας είναι ικανοποιήσιμη.

Ο αλγόριθμος πολυωνυμικού χώρου για το **SAT**, όπως προκύπτει από την απόδειξη του θεωρήματος 2.1.6, στη χειρότερη περίπτωση διατρέχει ολόκληρο το δέντρο αυτο-αναγωγής μιας δοσμένης έκφρασης για να μπορέσει να αποφασίσει αν είναι ικανοποιήσιμη ή όχι. Επομένως έχει εκθετική χρονική πολυπλοκότητα, αφού το δέντρο που αντιστοιχεί σε μία έκφραση με n μεταβλητές έχει 2^n φύλλα (ένα για κάθε δυνατή ανάθεση τιμών στις μεταβλητές) και συνολικά $\sum_{i=0}^{n} 2^i = 2^{n+1} - 1$ κόμβους. Οι αλγόριθμοι 3 και 4 εκμεταλλεύονται την υποτιθέμενη ύπαρξη μίας **NP**-δύσκολης μονοσυμβολικής γλώσσας, ή μίας **coNP**-δύσκολης αραιής γλώσσας αντίστοιχα, ώστε να κατευθύνουν την αναζήτηση σε ορισμένα μονοπάτια (αναθέσεις τιμών) που είναι βέβαιο ότι θα δώσουν την ανάθεση που ικανοποιεί τη δοσμένη έκφραση, αν υπάρχει. Με αυτόν τον τρόπο αποφεύγεται η εξέταση όλων των κόμβων του δέντρου αυτο-αναγωγής και οι αλγόριθμοι λειτουργούν σε πολυωνυμικό χρόνο.

Συγκεκριμένα, στον αλγόριθμο 3, σε κάθε επίπεδο ψαλιδίζονται τα υποδέντρα εκείνων των κόμβων που δεν απεικονίζονται μέσω της συνάρτησης αναγωγής σε συμβολοακολουθίες που ανήκουν στο $\{1\}^\star$. Το σύνολο $\{1\}^\star$

είναι αραιό, ανήκει στην κλάση **P** και είναι υπερσύνολο οποιασδήποτε μονοσυμβολικής γλώσσας. Παρέχει λοιπόν έναν υπολογιστικά εύκολο τρόπο να αποκλειούμε κάποιες αναθέσεις που σίγουρα δεν θα οδηγήσουν σε αποτίμηση της αρχικής έκφρασης στην τιμή 1. Επίσης, για τον περαιτέρω περιορισμό των εξεταζόμενων υποδέντρων χρησιμοποιείται το γεγονός ότι, από όλους τους κόμβους του τρέχοντος επιπέδου που απεικονίζονται στην ίδια συμβολοακολουθία, αρκεί να εξεταστεί το υποδέντρο του ενός μόνο. Στον αλγόριθμο αυτό διευκολυνόμαστε κατά πολύ από το γεγονός ότι έχουμε υποθέσει ότι υπάρχει **NP**-δύσκολη *μονοσυμβολική* γλώσσα. Οι συμβολοακολουθίες που ανήκουν σε μία τέτοια γλώσσα αναγνωρίζονται πολύ εύκολα και έτσι οι συνθήκες ψαλιδισμού είναι αρκετά απλές. Στον αλγόριθμο 4, ο ψαλιδισμός δυσχεραίνεται από το γεγονός ότι οι αραιές γλώσσες δεν έχουν απαραίτητα κάποιο εύκολα αναγνωρίσιμο υπερσύνολο, επομένως δεν μπορούμε να χρησιμοποιήσουμε κάποια συνθήκη ψαλιδισμού αντίστοιχη της "$g(\psi) \notin \{1\}^\star$". Για το λόγο αυτό ο μόνος ψαλιδισμός που γίνεται αφορά τις εκφράσεις που απεικονίζονται στο ίδιο στοιχείο του S. Από μόνη της η συνθήκη αυτή δεν εξασφαλίζει ότι θα εξεταστεί το πολύ πολυωνυμικός, ως προς το μήκος της εισόδου, αριθμός κόμβων του δέντρου αυτο-αναγωγής. Όμως, λόγω της συνθήκης της γραμμής 4.13, ο αριθμός των εξεταζόμενων κόμβων σε κάθε επίπεδο του δέντρου αυτο-αναγωγής περιορίζεται ρητά σε πολυωνυμικό. Αποδεικνύεται ότι αν ξεπεραστεί αυτό το όριο, τότε στο τρέχον επίπεδο κάποια υπο-έκφραση είναι ικανοποιήσιμη, άρα και η αρχική έκφραση είναι ικανοποιήσιμη.

Παρατηρούμε, τέλος, ότι ο αλγόριθμος 3 μπορεί να τροποποιηθεί πολύ εύκολα και χωρίς σπουδαίο κόστος στο χρόνο εκτέλεσης ώστε, όταν ολοκληρωθεί η εκτέλεση και αν η δοσμένη έκφραση είναι ικανοποιήσιμη, να παρέχει ως έξοδο και την ίδια την ανάθεση τιμών που οδήγησε στην ικανοποίηση της έκφρασης, εκτός από την απόφαση ότι η έκφραση είναι ικανοποιήσιμη. Αυτό μπορεί να γίνει αν τα σύνολα C και S περιέχουν, αντί για εκφράσεις ψ, ζεύγη $\langle \psi, w \rangle$, όπου w είναι μία συμβολοακολουθία που αποτελείται από τα σύμβολα 0 και 1, έχει μήκος $0 \leq |w| \leq m$ και το i-οστό της σύμβολο είναι η τιμή που έχει ανατεθεί στη μεταβλητή v_i στο συγκεκριμένο κόμβο. Έτσι, όταν οι επαναλή-

Αλγόριθμος 5 Τροποποιημένος ντετερμινιστικός αλγόριθμος πολυωνυμικού χρόνου για το **SAT**, υπό τις προϋποθέσεις του θεωρήματος 2.2.4, ώστε να φτάνει μέχρι το τελευταίο επίπεδο του δέντρου αυτο-αναγωγής

Είσοδος: λογική έκφραση ϕ με μεταβλητές $v_1, v_2, \ldots, v_m$, $m \geq 0$

$C \leftarrow \{\phi\}$
for $i \leftarrow 1$ **to** m **do**
    $R \leftarrow \emptyset$
    for all $\psi \in C$ **do**
        $R \leftarrow R \cup \{\psi\,[v_i := 1]\,, \psi\,[v_i := 0]\}$
    end for
    $C \leftarrow \emptyset$
    for all $\psi \in R$ **do**
        if $|C| < p_d(p_k(|\phi|)) + 1$ και δεν υπάρχει άλλη έκφραση $\psi' \in C$ με $g(\psi') = g(\psi)$ **then**
            $C \leftarrow C \cup \{\psi\}$
        end if
    end for
end for
αν υπάρχει έκφραση $\psi \in C$ με τιμή 1 τότε αποδέξου, αλλιώς απόρριψε

ψεις του αλγορίθμου ολοκληρωθούν, για κάθε έκφραση με τιμή 1 που υπάρχει στο C είναι γνωστή και η αντίστοιχη ανάθεση τιμών για όλες τις μεταβλητές της αρχικής έκφρασης. Η μετατροπή αυτή είναι εφικτή γιατί ο αλγόριθμος φτάνει πάντα μέχρι το τελευταίο επίπεδο του δέντρου αυτο-αναγωγής. Αντίθετα, επειδή ο αλγόριθμος 4 περιέχει τη συνθήκη της γραμμής 4.13 είναι δυνατόν για ορισμένες εισόδους η εκτέλεση να μη φτάσει μέχρι το τελευταίο επίπεδο. Ο αλγόριθμος 5 είναι μία τροποποιημένη εκδοχή του αλγορίθμου 4, ώστε να φτάνει πάντα μέχρι το τελευταίο επίπεδο του δέντρου. Η ιδέα είναι απλή: κάθε φορά που προστίθεται μία έκφραση στο σύνολο C, ελέγχεται αν η πληθικότητα του C έχει φτάσει την τιμή $p_d(p_k(|\phi|)) + 1$. Αν αυτό ισχύει, τότε δεν προστίθενται άλλες εκφράσεις στο C και ο αλγόριθμος συνεχίζει με την επόμενη επανάληψη. Ο νέος αλγόριθμος είναι πολυωνυμικού χρόνου, αφού εξετάζει το πολύ $|\phi| \cdot (p_d(p_k(|\phi|)) + 1)$ κόμβους του δέντρου αυτο-αναγωγής. Η ορθότητά του προκύπτει εύκολα με επιχειρήματα παρόμοια με αυτά που περιέχονται στην απόδειξη του θεωρήματος 2.2.4. Ο αλγόριθμος 5 φτάνει πάντα μέχρι το τελευταίο επίπεδο του δέντρου. Άρα μπορούμε να τον τροποποιήσουμε και αυτόν ώστε να δίνει την ικανοποιούσα αποτίμηση, αν υπάρχει, στο τέλος της εκτέλεσης.

2.3 Η τεχνική του αριστερού συνόλου

Είδαμε στις προηγούμενες παραγράφους πώς αποδεικνύεται, μέσω αλγορίθμων που ψαλιδίζουν το δέντρο αυτοαναγωγής του **SAT**, ότι σύνολα με "λίγα" στοιχεία δεν μπορούν να είναι δύσκολα για τις κλάσεις **NP** και **coNP**. Ο Mahaney επέκτεινε τα αποτελέσματα αυτά αποδεικνύοντας το ακόλουθο κομψό και αναμενόμενο, από μία άποψη, θεώρημα.

Θεώρημα 2.3.1 *[Mah82] Αν υπάρχει αραιή γλώσσα που είναι* **NP***-πλήρης, τότε* $\mathbf{P} = \mathbf{NP}$.

Αν προσπαθούσαμε να χρησιμοποιήσουμε την ιδέα της απόδειξης του θεωρήματος 2.2.4 για την απόδειξη του θεωρήματος 2.3.1, θα καταλήγαμε σε έναν αλγόριθμο παρόμοιο με τον αλγόριθμο 4 όπου όμως η συνθήκη $|C| \geq p_d(p_k(|\phi|))+1$

δεν μπορεί να εξασφαλίσει κάτι καλύτερο από το ότι υπάρχει στο C τουλάχιστον μία μη ικανοποιήσιμη έκφραση. Άρα αυτή η προσέγγιση δεν μας παρέχει εύκολο τρόπο ώστε να ψαλιδίσουμε το δέντρο αυτο-αναγωγής.

Για την επίδειξη της τεχνικής του αριστερού συνόλου θα δείξουμε ένα πιο γενικό θεώρημα, που οφείλεται στους Ogiwara και Watanabe και αφορά την ύπαρξη γλωσσών που είναι δύσκολες για την κλάση **NP** υπό την έννοια αναγωγής με φραγμένο πίνακα αληθείας (bounded truth-table reduction). Στη συνέχεια δίνουμε τη διατύπωση του θεωρήματος των Ogiwara και Watanabe, τους απαραίτητους ορισμούς για την ανάπτυξη της απόδειξης, και τέλος την ίδια την απόδειξη.

Θεώρημα 2.3.2 *[OW91] Αν υπάρχει αραιή γλώσσα που είναι* $\mathbf{NP}$-$\leq^{\mathrm{p}}_{\mathrm{btt}}$*-δύσκολη, τότε* $\mathbf{P} = \mathbf{NP}$.

Η απόδειξη του θεωρήματος χρησιμοποιεί το γεγονός ότι το αριστερό σύνολο κάθε γλώσσας του **NP** ανάγεται σε ένα αραιό σύνολο, ώστε να μπορέσει να ψαλιδίσει το σύνολο των πιθανών μαρτύρων μίας συμβολοακολουθίας σε πολυωνυμικό μέγεθος. Ουσιαστικά καθορίζει bit προς bit το πρόθεμα των υποψήφιων μαρτύρων, φροντίζοντας να απορρίπτει κάποια προθέματα ώστε κάθε φορά να έχει το πολύ πολυωνυμικό πλήθος υποψήφιων προθεμάτων. Ο αλγόριθμος που αποφασίζει την **NP** γλώσσα σε πολυωνυμικό χρόνο δεν χρησιμοποιεί άμεσα την ιδιότητα αυτο-αναγωγής κάποιου συνόλου, αλλά συνδυάζει ένα είδος ψαλιδισμού με δυαδική αναζήτηση στο χώρο των διαστημάτων επί του $\Sigma^{p(|x|)}$, όπου $p(|x|)$ είναι το μήκος των μαρτύρων του x.

2.3.1 Βασικές έννοιες και ορισμοί

Αναγωγές με πίνακα αληθείας

Ορισμός 2.3.3 *(Αναγωγή Turing) Λέμε ότι η γλώσσα A ανάγεται κατά Turing στη γλώσσα B, και γράφουμε $A \leq^{\mathrm{p}}_{\mathrm{T}} B$, αν και μόνο αν $A \in \mathbf{P}^B$.*

Ορισμός 2.3.4 *(Αναγωγή πίνακα αληθείας) Λέμε ότι η γλώσσα A ανάγεται με πίνακα αληθείας στη γλώσσα B, και γράφουμε $A \leq^{\mathrm{p}}_{\mathrm{tt}} B$, αν και μόνο αν*

$(\exists g \in \mathbf{FP})(\exists L \in \mathbf{P})(\forall x)\,[(\exists k)(\exists z_1, z_2, \ldots, z_k)\,[g(x) = z_1\#z_2\# \cdots z_k\#\wedge$
$(x \in A \iff x\#\chi_B(z_1)\chi_B(z_2)\cdots\chi_B(z_k) \in L)]].$

Ορισμός 2.3.5 *(Αναγωγή $f(n)$-πίνακα αληθείας) Λέμε ότι η γλώσσα A ανάγεται με $f(n)$-πίνακα αληθείας στη γλώσσα B, και γράφουμε $A \leq^{\mathrm{p}}_{f(n)-\mathrm{tt}} B$ αν και μόνο αν $(\exists g \in \mathbf{FP})(\exists L \in \mathbf{P})(\forall x)\,[(\exists k : k \leq f(|x|))(\exists z_1, z_2, \ldots, z_k)$ $[g(x) = z_1\#z_2\# \cdots z_k\# \wedge (x \in A \iff x\#\chi_B(z_1)\chi_B(z_2)\cdots\chi_B(z_k) \in L)]].$*

Ορισμός 2.3.6 *(Αναγωγή φραγμένου πίνακα αληθείας) Λέμε ότι η γλώσσα A ανάγεται με φραγμένο πίνακα αληθείας στη γλώσσα B, και γράφουμε $A \leq^{\mathrm{p}}_{\mathrm{btt}} B$, αν και μόνο αν $\exists k\; A \leq^{\mathrm{p}}_{k-\mathrm{tt}} B$.*

Υπενθυμίζουμε ότι στην αναγωγή κατά Karp, αν $A \leq^{\mathrm{p}}_{\mathrm{m}} B$, τότε μία μηχανή Turing που αποφασίζει την A υπολογίζει σε πολυωνυμικό χρόνο μία συμβολοακολουθία $g(x)$, όπου x η είσοδος, και αποφαίνεται για το αν $x \in A$ ανάλογα με το αν $g(x) \in B$. Αντίστοιχα, αν μία γλώσσα A ανάγεται κατά Turing στη γλώσσα B, τότε η μηχανή Turing που αποφασίζει την A λειτουργεί σε πολυωνυμικό χρόνο και μπορεί να χρησιμοποιεί τη γλώσσα B ως μαντείο. Στην περίπτωση της αναγωγής πίνακα αληθείας, αν υποθέσουμε ότι $A \leq^{\mathrm{p}}_{\mathrm{tt}} B$, τότε μία μηχανή Turing που αποφασίζει την A, για την είσοδο x υπολογίζει σε πολυωνυμικό χρόνο τις ερωτήσεις που θα θέσει στο μαντείο B και στη συνέχεια αποφασίζει, επίσης σε πολυωνυμικό χρόνο, χρησιμοποιώντας τόσο την είσοδό της όσο και τις απαντήσεις του μαντείου, αν $x \in A$. Σημειώνουμε ότι οι ερωτήσεις προς το μαντείο B, σε αντίθεση με την αναγωγή Turing, δεν επιτρέπεται να είναι προσαρμοστικές, δηλαδή για κάθε ερώτηση j, καμία τιμή $\chi_B(z_i)$, $i < j$, δεν συμμετέχει στον προσδιορισμό της συμβολοακολουθίας z_j. Η αναγωγή $f(n)$-πίνακα αληθείας είναι μία αναγωγή πίνακα αληθείας όπου το πλήθος των ερωτήσεων στο μαντείο φράσσεται άνω από μία τιμή που εξαρτάται από το μήκος της εισόδου, ενώ στην αναγωγή φραγμένου πίνακα αληθείας υπάρχει σταθερό φράγμα ανεξάρτητο από την είσοδο. Από τους ορισμούς έπεται άμεσα ότι $A \leq^{\mathrm{p}}_{\mathrm{m}} B \Rightarrow A \leq^{\mathrm{p}}_{\mathrm{btt}} B \Rightarrow A \leq^{\mathrm{p}}_{\mathrm{tt}} B \Rightarrow A \leq^{\mathrm{p}}_{\mathrm{T}} B$.

Για να δικαιολογήσουμε κατά κάποιον τρόπο την ονομασία των αναγωγών με πίνακα αληθείας, δίνουμε χωρίς απόδειξη έναν χαρακτηρισμό της αναγωγής φραγμένου πίνακα αληθείας που θα μας χρησιμεύσει στην απόδειξη του θεωρήματος 2.3.2. Για το σκοπό αυτό, ορίζουμε την *k-συνθήκη* ως μία πλειάδα $k+1$ στοιχείων, όπου το πρώτο στοιχείο είναι μία boolean συνάρτηση που δέχεται k ορίσματα από το σύνολο $\Sigma^\star$ των συμβολοακολουθιών επί ενός αλφαβήτου Σ, και τα υπόλοιπα στοιχεία είναι συμβολοακολουθίες του $\Sigma^\star$. Αν $C = (\alpha, v_1, v_2, \ldots, v_k)$ είναι μία k-συνθήκη, η συνάρτηση α είναι ο *πίνακας αληθείας* της C, το σύνολο $\{v_1, v_2, \ldots, v_k\}$ είναι το σύνολο των *ερωτήσεων* της C και για κάθε i με $1 \leq i \leq k$ η v_i είναι η *i-οστή ερώτηση* της C. Τέλος, αν $D \subseteq \Sigma^\star$ είναι μία γλώσσα επί του αλφαβήτου Σ και $C = (\alpha, v_1, v_2, \ldots, v_k)$ είναι μία k-συνθήκη επί του ίδιου αλφαβήτου, λέμε ότι η C *ικανοποιείται* από την D αν $\alpha\left(\chi_D(v_1), \chi_D(v_2), \ldots, \chi_D(v_k)\right) = 1$.

Θεώρημα 2.3.7 *Αν $C, D \subseteq \Sigma^\star$ και $C \leq^{\mathrm{p}}_{\mathrm{btt}} D$, τότε υπάρχει ένας ακέραιος $k \geq 1$ και μία πολυωνυμικού χρόνου συνάρτηση f από το $\Sigma^\star$ στο σύνολο των k-συνθηκών, έτσι ώστε για κάθε $u \in \Sigma^\star$ να ισχύει:*

$$u \in C \iff \text{η } f(u) \text{ ικανοποιείται από την } D.$$

Η έννοια του αριστερού συνόλου

Στο εισαγωγικό κεφάλαιο είδαμε τον ορισμό της κλάσης πολυπλοκότητας **NP**, ως $\mathbf{NP} = \bigcup_{k\geq 0} \mathbf{NTIME}(n^k)$. Ένας ισοδύναμος εναλλακτικός ορισμός είναι ο ακόλουθος:

Θεώρημα 2.3.8 *Η κλάση **NP** περιέχει ακριβώς εκείνες τις γλώσσες L για τις οποίες υπάρχει ένα πολυώνυμο p και μία γλώσσα $A \in \mathbf{P}$, έτσι ώστε για κάθε x να ισχύει:*

$$x \in L \iff (\exists w \in \Sigma^{p(|x|)})[\langle x, w\rangle \in A]$$

Διαισθητικά, αν $x \in L$ τότε το w είναι μία κωδικοποίηση όλων των μη ντετερμινιστικών επιλογών της μηχανής Turing που αποδέχεται την L, και αντιστοιχεί σε κάποιο μονοπάτι που καταλήγει σε κατάσταση αποδοχής για την είσοδο x. Το γεγονός ότι η γλώσσα A αποφασίζεται σε ντετερμινιστικό πολυωνυμικό χρόνο σημαίνει ότι αν δίνεται μία είσοδος $x \in L$ και η κωδικοποίηση w των μη ντετερμινιστικών επιλογών σε κάποιο μονοπάτι που καταλήγει σε κατάσταση αποδοχής, τότε μία μηχανή Turing μπορεί να επαληθεύσει σε ντετερμινιστικό πολυωνυμικό χρόνο ότι $x \in L$. Για το λόγο αυτό, η συμβολοακολουθία w ονομάζεται *μάρτυρας του x ως προς τη γλώσσα A και το πολυώνυμο p*. Από το θεώρημα 2.3.8 προκύπτει ότι μπορούμε χωρίς βλάβη της γενικότητας να θεωρούμε ότι κάθε μη ντετερμινιστική μηχανή Turing πολυωνυμικού χρόνου έχει υπολογιστικά μονοπάτια μήκους ακριβώς $p(|x|)$, όπου p κάποιο πολυώνυμο, και κάθε μη ντετερμινιστική επιλογή γίνεται μεταξύ δύο δυνατών μεταβάσεων. Έτσι το δέντρο του υπολογισμού της μηχανής για κάθε είσοδο x είναι ένα πλήρες δυαδικό δέντρο με ύψος $p(|x|)$.

Ορισμός 2.3.9 *Για κάθε $x \in \Sigma^\star$, για κάθε γλώσσα $A \in \mathbf{P}$ και για κάθε πολυώνυμο p, ορίζουμε το σύνολο των μαρτύρων του x ως προς A, p:*

$$^{A,p}\mathrm{witness}(x) = \{w \in \Sigma^{p(|x|)} : \langle x, w\rangle \in A\}$$

Ορίζουμε ακόμη το συμβολισμό $^{A,p}L$:

$$^{A,p}L = \{x : (\exists w \in \Sigma^{p(|x|)})[\langle x, w\rangle \in A]\}$$

Για κάθε γλώσσα $A \in \mathbf{P}$ και για κάθε πολυώνυμο p φαίνεται εύκολα ότι $^{A,p}L \in \mathbf{NP}$ και για κάθε x ισχύει $x \in {}^{A,p}L \iff {}^{A,p}\mathrm{witness}(x) \neq \emptyset$. Αν θεωρήσουμε τη λεξικογραφική διάταξη $\prec$ μεταξύ συμβολοακολουθιών του $\Sigma^\star$ του ιδίου μήκους[2], τότε έχει νόημα η έννοια του *μέγιστου μάρτυρα του x ως προς A, p*.

Ορισμός 2.3.10 *Για κάθε $x \in \Sigma^\star$, για κάθε γλώσσα $A \in \mathbf{P}$ και για κάθε πολυώνυμο p, ορίζουμε τον μέγιστο μάρτυρα του x ως προς A, p:*

$$^{A,p}\mathrm{maxwitness}(x) = \max {}^{A,p}\mathrm{witness}(x)$$

[2] Για τον τυπικό ορισμό της $\prec$, υποθέτουμε ότι έχει οριστεί μία ολική διάταξη $\prec_\Sigma$ επί του αλφαβήτου Σ. Τότε η $\prec$ ορίζεται ως $\prec = \{(\alpha w, \beta v) : \alpha \prec_\Sigma \beta \wedge w, v \in \Sigma^\star \wedge |w| = |v|\} \cup \{(\alpha w, \alpha v) : \alpha \in \Sigma \wedge w \prec v\}$.

Για κάθε x, οι μάρτυρες ως προς A, p έχουν σταθερό μήκος $p(|x|)$. Επομένως έχει νόημα να θεωρήσουμε όλες τις συμβολοακολουθίες που είναι μικρότερες ή ίσες (κατά τη διάταξη $\prec$) από το μέγιστο μάρτυρα του x ως προς A, p. Αν θεωρήσουμε τα στοιχεία του $\Sigma^{p(|x|)}$ διατεταγμένα σε μία γραμμή κατά αύξουσα σειρά, οι συμβολοακολουθίες αυτές θα βρίσκονται *στα αριστερά* του μέγιστου μάρτυρα. Για κάθε μία από αυτές, υπάρχει κάποιος μάρτυρας λεξικογραφικά μεγαλύτερός της, ή ίσος με αυτήν. Ορίζουμε λοιπόν το *αριστερό σύνολο του x ως προς A, p.*

Ορισμός 2.3.11 *Για κάθε $x \in \Sigma^\star$, για κάθε γλώσσα $A \in \mathbf{P}$ και για κάθε πολυώνυμο p, ορίζουμε το αριστερό σύνολο του x ως προς A, p:*

$$^{A,p}\mathrm{Left}(x) = \{y \in \Sigma^{p(|x|)} : (\exists w \in {}^{A,p}\mathrm{witness}(x))[y \preceq w]\}$$

Προφανώς αν $x \in {}^{A,p}L$ τότε το σύνολο $^{A,p}\mathrm{Left}(x)$ περιέχει όλες τις συμβολοακολουθίες που βρίσκονται στα αρίστερα του μέγιστου μάρτυρα του x ως προς A, p. Αντίθετα, αν $x \notin {}^{A,p}L$, τότε δεν υπάρχουν μάρτυρες του x ως προς A, p, κατά συνέπεια δεν ορίζεται μέγιστος μάρτυρας και $^{A,p}\mathrm{Left}(x) = \emptyset$. Άρα έχουμε ότι:

$$^{A,p}\mathrm{Left}(x) = \begin{cases} \{y : 0^{p(|x|)} \preceq y \preceq {}^{A,p}\mathrm{maxwitness}(x)\} & \text{, αν } x \in {}^{A,p}L \\ \emptyset & \text{, αν } x \notin {}^{A,p}L \end{cases}$$

Τέλος, ορίζουμε το *αριστερό σύνολο ως προς A, p* ως το σύνολο που περιέχει όλες τις κωδικοποιήσεις ζευγών $\langle x, y\rangle$ για τις οποίες ισχύει ότι το y είναι μικρότερο ή ίσο λεξικογραφικά από το μέγιστο μάρτυρα του x, ως προς A, p.

Ορισμός 2.3.12 *Για κάθε γλώσσα $A \in \mathbf{P}$ και για κάθε πολυώνυμο p, ορίζουμε το αριστερό σύνολο ως προς A, p:*

$$^{A,p}\mathrm{Left} = \bigcup_{x \in \Sigma^\star} \{\langle x, y\rangle : y \in {}^{A,p}\mathrm{Left}(x)\}$$

Αν $\langle x, y\rangle \in {}^{A,p}\mathrm{Left}$, τότε από τους παραπάνω ορισμούς προκύπτει αμέσως ότι $x \in {}^{A,p}L$ και $y \preceq {}^{A,p}\mathrm{maxwitness}(x)$. Ακόμη, $\langle x, y'\rangle \in {}^{A,p}\mathrm{Left}$, για κάθε

$y' \prec y$. Αυτές τις ιδιότητες, καθώς και την τελευταία ιδιότητα που μας ενδιαφέρει, ότι αν $A \in \mathbf{P}$ και p πολυώνυμο τότε ${}^{A,p}\mathrm{Left} \in \mathbf{NP}$, τις δίνουμε στη συνέχεια με τη μορφή θεωρημάτων.

Θεώρημα 2.3.13 *Για κάθε γλώσσα $A \in \mathbf{P}$, για κάθε πολυώνυμο p και για κάθε $x \in \Sigma^\star$ και $y, y' \in \Sigma^{p(|x|)}$ ισχύουν:*

1. $\langle x, y\rangle \in {}^{A,p}\mathrm{Left} \iff y \preceq {}^{A,p}\mathrm{maxwitness}(x)$.

2. $\langle x, y\rangle \in {}^{A,p}\mathrm{Left} \wedge y' \prec y \Rightarrow \langle x, y'\rangle \in {}^{A,p}\mathrm{Left}$.

Θεώρημα 2.3.14 *Για κάθε γλώσσα $A \in \mathbf{P}$ και για κάθε πολυώνυμο p έχουμε ότι ${}^{A,p}\mathrm{Left} \in \mathbf{NP}$.*

Απόδειξη

Εφόσον η γλώσσα A αποφασίζεται σε ντετερμινιστικό πολυωνυμικό χρόνο, ο αλγόριθμος 6 είναι ένας μη ντετερμινιστικός αλγόριθμος πολυωνυμικού χρόνου για το σύνολο ${}^{A,p}\mathrm{Left}$. Συνεπώς ${}^{A,p}\mathrm{Left} \in \mathbf{NP}$.

■

Διαστήματα επί του Σ^n

Έχοντας ορίσει τη λεξικογραφική διάταξη μεταξύ συμβολοακολουθιών ίσου μήκους, μπορούμε να χρησιμοποιούμε την έννοια του διαστήματος επί του συνόλου των συμβολοακολουθιών που περιέχουν σύμβολα ενός αλφαβήτου Σ και έχουν το ίδιο μήκος.

Ορισμός 2.3.15 *Αν $n \geq 1$ είναι ένας ακέραιος, Σ ένα αλφάβητο και $I \subseteq \Sigma^n$, λέμε ότι το I είναι* διάστημα επί του Σ^n *αν υπάρχουν συμβολοακολουθίες y, z μήκους n τέτοιες ώστε $y \preceq z$ και $I = \{u : y \preceq u \preceq z\}$. Σε αυτήν την περίπτωση το I γράφεται $I = [y, z]$, η y ονομάζεται* αριστερό άκρο *του διαστήματος I και η z ονομάζεται* δεξιό άκρο *του διαστήματος I.*

Αλγόριθμος 6 Μη ντετερμινιστικός αλγόριθμος πολυωνυμικού χρόνου για το σύνολο ${}^{A,p}\mathrm{Left}$, όπου $A \in \mathbf{P}$ και p πολυώνυμο

Είσοδος: $\langle x, y \rangle$

if $|y| \neq p(|x|)$ **then**
    τερμάτισε την εκτέλεση σε κατάσταση απόρριψης
end if
επέλεξε μη ντετερμινιστικά μία συμβολοακολουθία w με μήκος ακριβώς $p(|x|)$
if $y \succ w$ **then**
    τερμάτισε την εκτέλεση σε κατάσταση απόρριψης
end if
if $\langle x, w \rangle \notin A$ **then**
    τερμάτισε την εκτέλεση σε κατάσταση απόρριψης
end if
τερμάτισε την εκτέλεση σε κατάσταση αποδοχής

Ορισμός 2.3.16 *Αν $I = [u, v]$, $J = [y, z]$ είναι δύο διαστήματα επί του Σ^n, λέμε ότι τα I, J είναι* ξένα *αν είναι ξένα ως σύνολα συμβολοακολουθιών, δηλαδή ισχύει $z \prec u$ ή $v \prec y$. Αν κάθε στοιχείο του I είναι λεξικογραφικά μικρότερο από όλα τα στοιχεία του J, τότε λέμε ότι το διάστημα I είναι* λεξικογραφικά μικρότερο *από το διάστημα J και γράφουμε $I \prec J$.*

Ορισμός 2.3.17 *Για κάθε γλώσσα $A \in \mathbf{P}$, για κάθε πολυώνυμο p, για κάθε $x \in \Sigma^\star$ και για κάθε σύνολο Λ διαστημάτων επί του $\Sigma^{p(|x|)}$, λέμε ότι το Λ είναι* πιστό *για το x ως προς A, p αν ισχύει:*

$$x \in {}^{A,p}L \Rightarrow (\exists I \in \Lambda)[{}^{A,p}\mathrm{maxwitness}(x) \in I].$$

Ορισμός 2.3.18 *Για κάθε γλώσσα $A \in \mathbf{P}$, για κάθε πολυώνυμο p, για κάθε $x \in \Sigma^\star$ και για οποιαδήποτε σύνολα Λ, Γ διαστημάτων επί του $\Sigma^{p(|x|)}$, λέμε ότι το Γ είναι* εκλέπτυνση *του Λ για το x ως προς A, p, αν ισχύει:*

το Λ είναι πιστο για το x ως προς A, p $\Rightarrow$
το Γ είναι πιστο για το x ως προς A, p.

Στα επόμενα θα περιοριστούμε σε γλώσσες που ορίζονται στο αλφάβητο $\Sigma = \{0,1\}$, δηλαδή οι συμβολοακολουθίες επί του Σ μπορούν να θεωρούνται ως ακολουθίες δυαδικών ψηφίων. Στο Σ ορίζουμε τη συνήθη διάταξη $\prec_\Sigma = \{(0,1)\}$. Παρατηρούμε απευθείας από τον ορισμό του πιστού συνόλου διαστημάτων ότι για κάθε $A \in \mathbf{P}$ και για κάθε πολυώνυμο p, το μονοσύνολο που περιέχει ένα διάστημα που καλύπτει όλες τις συμβολοσειρές του $\Sigma^{p(|x|)}$, συγκεκριμένα το $\{[0^{p(|x|)}, 1^{p(|x|)}]\}$, είναι πιστό για οποιοδήποτε $x \in \Sigma^\star$ ως προς A, p. Επιπρόσθετα, αν $x \notin {}^{A,p}L$ τότε οποιοδήποτε σύνολο ξένων ανά δύο διαστημάτων επί του $\Sigma^{p(|x|)}$ είναι πιστό για το x ως προς A, p.

Λίστες υποθέσεων

Το τελευταίο εργαλείο που θα μας χρειαστεί για την απόδειξη του θεωρήματος 2.3.2 είναι οι λίστες υποθέσεων. Διαισθητικά, μία λίστα υποθέσεων για κάποιο σύνολο S περιέχει ζεύγη συμβολοακολουθίας-απάντησης όπου η απάντηση είναι ένα σύμβολο 0 ή 1 και εκφράζει την υπόθεση για το αν η συμβολοακολουθία ανήκει στο S ή όχι. Εφόσον αναφερόμαστε σε λίστα και όχι σε σύνολο υποθέσεων, έχει σημασία η σειρά με την οποία έχουν εισαχθεί στη λίστα οι υποθέσεις.

Ορισμός 2.3.19 *Μία (πιθανώς κενή) λίστα ζευγών της μορφής (w, b), όπου $w \in \Sigma^\star$ και $b \in \{0,1\}$, ονομάζεται* λίστα υποθέσεων.

Ορισμός 2.3.20 *Για κάθε σύνολο $S \subseteq \Sigma^\star$ και για κάθε λίστα υποθέσεων τ, λέμε ότι η τ είναι* ορθή *ως προς το S αν για κάθε στοιχείο (w, b) της τ ισχύει ότι $\chi_S(w) = b$.*

Ορισμός 2.3.21 *Για κάθε γλώσσα $A \in \mathbf{P}$, για κάθε πολυώνυμο p, για κάθε $x \in \Sigma^\star$, για οποιαδήποτε σύνολα Λ, Γ διαστημάτων επί του $\Sigma^{p(|x|)}$, για κάθε λίστα υποθέσεων τ και για κάθε σύνολο S, λέμε ότι το Γ είναι τ,S-*εκλέπτυνση *του Λ για το x ως προς A, p, αν ισχύει:*

η τ είναι ορθή ως προς το S $\Rightarrow$
το Γ είναι εκλέπτυνση του Λ για το x ως προς A, p.

Θεώρημα 2.3.22 *Ιδιότητες των εκλεπτύνσεων:*

1. *Αν* $\Lambda = \Gamma_1 \cup \Gamma_2 \cup \ldots \cup \Gamma_m$ *και τα* $\Gamma'_1, \Gamma'_2, \ldots, \Gamma'_m$ *είναι* τ,S*-εκλεπτύνσεις των* $\Gamma_1, \Gamma_2, \ldots, \Gamma_m$*, αντίστοιχα, για το* x *ως προς* A*,* p*, τότε το* $\Gamma'_1 \cup \Gamma'_2 \cup \ldots \cup \Gamma'_m$ *είναι* τ,S*-εκλέπτυνση του* Λ *για το* x *ως προς* A*,* p*.*

2. *Αν το* Θ *είναι* τ,S*-εκλέπτυνση του* Γ *για το* x *ως προς* A*,* p *και το* Θ' *είναι* τ,S*-εκλέπτυνση του* Θ *για το* x *ως προς* A*,* p*, τότε το* Θ' *είναι* τ,S*-εκλέπτυνση του* Γ *για το* x *ως προς* A*,* p*.*

2.3.2 NP-$\leq^{\mathrm{p}}_{\mathrm{btt}}$-δύσκολες αραιές γλώσσες

Απόδειξη (του θεωρήματος 2.3.2)

Από την υπόθεση του θεωρήματος, υπάρχει ένα αραιό σύνολο S που είναι **NP**-$\leq^{\mathrm{p}}_{\mathrm{btt}}$-δύσκολο. Θα δείξουμε ότι οποιαδήποτε γλώσσα που ανήκει στο **NP** μπορεί να αποφασιστεί σε πολυωνυμικό χρόνο.

Έστω $L \in \mathbf{NP}$. Από το θεώρημα 2.3.8 και τον ορισμό 2.3.9 προκύπτει άμεσα ότι υπάρχει μία γλώσσα $A \in \mathbf{P}$ και ένα πολυώνυμο p ώστε $L = {}^{A,p}L$. Επομένως, από το θεώρημα 2.3.14 έχουμε ότι ${}^{A,p}\mathrm{Left} \in \mathbf{NP}$. Εφόσον το S είναι **NP**-$\leq^{\mathrm{p}}_{\mathrm{btt}}$-δύσκολο, το ${}^{A,p}\mathrm{Left}$ ανάγεται κατά $\leq^{\mathrm{p}}_{\mathrm{btt}}$ στο S. Άρα από το θεώρημα 2.3.7, υπάρχει ένας ακέραιος $k \geq 1$ και μία πολυωνυμικού χρόνου συνάρτηση f από το $\Sigma^\star$ στο σύνολο των k-συνθηκών ώστε για κάθε $u \in \Sigma^\star$ να ισχύει:

$$u \in {}^{A,p}\mathrm{Left} \iff \text{η } f(u) \text{ ικανοποιείται από το } S.$$

Ας υποθέσουμε ότι έχουμε διαθέσιμη μία διαδικασία **SUPPLYMAXWITNESS** πολυωνυμικού χρόνου, η οποία για κάθε είσοδο x παράγει ένα σύνολο συμβολοακολουθιών κάθε μία από τις οποίες έχει μήκος $p(|x|)$, με την επιπλέον ιδιότητα ότι αν $x \in L$ τότε ο μέγιστος μάρτυρας ${}^{A,p}\mathrm{maxwitness}(x)$ περιέχεται στο σύνολο αυτό. Τότε ο αλγόριθμος 7 αποφασίζει την L:

- Αν $x \in L$ τότε στο σύνολο που επιστρέφει η διαδικασία **SUPPLYMAXWITNESS** με είσοδο x θα υπάρχει η συμβολοακολουθία

Αλγόριθμος 7 Ντετερμινιστικός αλγόριθμος πολυωνυμικού χρόνου που αποφασίζει μία αυθαίρετη γλώσσα $L \in \mathbf{NP}$, υπό τις προϋποθέσεις του θεωρήματος 2.3.2

Είσοδος: x

$Y \leftarrow$ **SUPPLYMAXWITNESS**(x)
for all $y \in Y$ **do**
  if $\langle x, y \rangle \in A$ **then**
    τερμάτισε την εκτέλεση σε κατάσταση αποδοχής
  end if
end for
τερμάτισε την εκτέλεση σε κατάσταση απόρριψης

$^{A,p}\text{maxwitness}(x)$, για την οποία ισχύει $\langle x, {}^{A,p}\text{maxwitness}(x)\rangle \in A$. Άρα ο αλγόριθμος θα αποδεχτεί την είσοδο.

- Αν $x \notin L$ το επιστρεφόμενο σύνολο δεν μπορεί να περιέχει y που ικανοποιεί την $\langle x, y \rangle \in A$, διότι $^{A,p}\text{witness}(x) = \emptyset$. Άρα ο αλγόριθμος θα απορρίψει την είσοδο.

Εφόσον η διαδικασία **SUPPLYMAXWITNESS** εκτελείται σε πολυωνυμικό χρόνο, το σύνολο που επιστρέφει περιέχει το πολύ πολυωνυμικό αριθμό συμβολοακολουθιών. Άρα ο έλεγχος για το αν κάποιο στοιχείο του συνόλου είναι μάρτυρας του x ως προς A, p μπορεί να γίνει σε πολυωνυμικό χρόνο. Τελικά, ο αλγόριθμος 7 εκτελείται σε χρόνο πολυωνυμικό ως προς το μήκος της εισόδου του και αποφασίζει την L. Άρα $L \in \mathbf{P}$ και επειδή η L είναι αυθαίρετη γλώσσα της κλάσης $\mathbf{NP}$, προκύπτει $\mathbf{P} = \mathbf{NP}$.

Στο υπόλοιπο της απόδειξης θα επικεντρωθούμε στην κατασκευή της διαδικασίας **SUPPLYMAXWITNESS** ώστε να πληροί τις προϋποθέσεις που τέθηκαν παραπάνω. Σε αυτό το σημείο παρατηρούμε ότι το πλήθος των συμβολοακολουθιών με μήκος $p(|x|)$ είναι $2^{p(|x|)}$, άρα για να λειτουργήσει η **SUPPLYMAXWITNESS** σε πολυωνυμικό χρόνο πρέπει με κάποιον τρόπο να ψαλιδίζει το σύνολο $\Sigma^{p(|x|)}$ ώστε να επιστρέφει ένα σύνολο με το πολύ πολυωνυμικό πλήθος συμβολοακολουθιών.

Αρχικά ορίζουμε κάποια πολυώνυμα που χρησιμεύουν για να φράξουν άνω τα μεγέθη διαφόρων συμβολοσειρών και συνόλων που μας ενδιαφέρουν.

Ορισμός 2.3.23

- *Έστω $p_1(n)$ ένα πολυώνυμο που παίρνει γνησίως αύξουσες τιμές και για το οποίο για κάθε $x \in \Sigma^\star$ και για κάθε $y \in \Sigma^{p(|x|)}$ ισχύει $|\langle x, y\rangle| \leq p_1(|x|)$. Εφόσον χρησιμοποιούμε "λογική" συνάρτηση σύζευξης $\langle \cdot, \cdot \rangle$ το μήκος της κωδικοποίησης του ζεύγους είναι πολυωνυμικό ως προς τα μήκη των επιμέρους συμβολοακολουθιών. Επειδή το μήκος της y είναι ήδη πολυωνυμικό ως προς το μήκος της x, το μήκος της αναπαράστασης $\langle x, y\rangle$ είναι πολυωνυμικό ως προς $|x|$. Επίσης για κάθε πολυώνυμο υπάρχει ένα γνησίως αύξον πολυώνυμο που έχει μεγαλύτερη ή ίση τιμή από το αρχικό σε κάθε σημείο[3].*

- *Έστω $p_2(n)$ ένα γνησίως αύξον πολυώνυμο για το οποίο ισχύει ότι για κάθε $u \in \Sigma^\star$, όλες οι ερωτήσεις της συνθήκης $f(u)$ έχουν μήκος το πολύ $p_2(|u|)$. Το πολυώνυμο p_2 υπάρχει γιατί η f υπολογίζεται σε πολυωνυμικό χρόνο, άρα η αναπαράσταση της k-συνθήκης που αντιστοιχεί σε κάθε u έχει το πολύ πολυωνυμικό μήκος.*

- *Έστω $p_3(n)$ ένα γνησίως αύξον πολυώνυμο για το οποίο ισχύει για κάθε φυσικό n, $|S^{\leq n}| \leq p_3(n)$. Το πολυώνυμο p_3 υπάρχει γιατί το σύνολο S έχει υποτεθεί αραιό, συνεπώς η συνάρτηση πυκνότητάς του είναι το πολύ πολυωνυμική.*

- *Έστω $q(n)$ το πολυώνυμο που για κάθε φυσικό n παίρνει τιμή $q(n) = p_3(p_2(p_1(n)))$. Για κάθε κωδικοποίηση ζεύγους $\langle x, y\rangle$, η τιμή $p_2(p_1(|x|))$ φράσσει άνω το μέγεθος οποιασδήποτε ερώτησης της συνθήκης $f(\langle x, y\rangle)$. Το πλήθος των συμβολοακολουθιών που ανήκουν στο S και αποτελούν αναπαράσταση κάποιας ερώτησης που ανήκει σε κάποια συνθήκη $f(\langle x, y\rangle)$ είναι λοιπόν $p_3(p_2(p_1(|x|))) = q(|x|)$. Άρα:*

$$|\{w \in S : (\exists y \in \Sigma^{p(|x|)})[w \text{ είναι ερώτηση της } f(\langle x, y\rangle)]\}| \leq q(|x|)$$

[3] Αναφερόμαστε πάντα στο πεδίο ορισμού των φυσικών αριθμών. Αν $p(x) = \sum_{i=0}^{k} a_i x^i$, τότε το $p'(x) = \sum_{i=0}^{k} |a_i| x^i$ είναι γνησίως αύξον για $x \geq 0$ και για κάθε $n \in \mathbb{N}$ ισχύει $p(n) \leq p'(n)$.

- $r(n) = k!2^k(2q(n)+1)^k$.
- *Για κάθε* d *στο διάστημα* $0 \leq d \leq k$, *ορίζουμε το πολυώνυμο* $r_d(n) = (k-d)!2^{k-d}(2q(n)+1)^{k-d}$. *Ισχύουν* $r_0 \equiv r$ *και* $r_k \equiv 1$.

Για την κατασκευή της **SUPPLYMAXWITNESS** θα θεωρήσουμε προς το παρόν δεδομένη την ύπαρξη μιας διαδικασίας **CULL**, η οποία με είσοδο μία συμβολοακολουθία x, ένα σύνολο διαστημάτων Γ και μία λίστα υποθέσεων τ έτσι ώστε $|\Gamma| > r_{|\tau|}(|x|)$, επιστρέφει ένα σύνολο διαστημάτων που αποτελεί τ,S-εκλέπτυνση του Γ για το x ως προς A, p και έχει πληθικότητα το πολύ $r_{|\tau|}(|x|)$.

Η **SUPPLYMAXWITNESS** δουλεύει εσωτερικά με διαστήματα και περιγράφεται από τον αλγόριθμο 8. Ξεκινώντας με ένα σύνολο διαστημάτων Λ που περιλαμβάνει ένα διάστημα που καλύπτει όλο το $\Sigma^{p(|x|)}$, εκτελεί $p(|x|)$ φορές μία διαδικασία δύο βημάτων:

- **Διαμέριση** Αρχικά διαμερίζει κάθε διάστημα που ανήκει στο Λ σε δύο ισομήκη διαστήματα.
- **Εκλέπτυνση** Στη συνέχεια, αν το Λ περιέχει περισσότερα από $r(|x|)$ στοιχεία, καλεί τη διαδικασία **CULL** με κενή λίστα υποθέσεων ώστε να λάβει ένα υποσύνολο του Λ που είναι πιστό για το x ως προς A, p και περιέχει το πολύ $r(|x|)$ διαστήματα.

Μετά από $p(|x|)$ επαναλήψεις των βημάτων αυτών, κάθε στοιχείο του Λ είναι ένα εκφυλισμένο διάστημα της μορφής $[u, u]$. Η διαδικασία επιστρέφει ως αποτέλεσμα το σύνολο $\{u \in \Sigma^{p(|x|)} : [u, u] \in \Lambda\}$.

Παρατηρούμε ότι αν το Λ είναι πιστό για το x ως προς A, p στην αρχή του βήματος διαμέρισης, εξακολουθεί να διατηρεί την ιδιότητα αυτή και στο τέλος του βήματος διαμέρισης. Επίσης, επειδή η **CULL** επιστρέφει ένα σύνολο Y που είναι τ,S-εκλέπτυνση του Λ για το x ως προς A, p και επειδή η κενή λίστα υποθέσεων είναι ορθή ως προς το S, προκύπτει ότι αν το Λ είναι πιστό για το x ως προς A, p στην αρχή του βήματος εκλέπτυνσης, τότε εξακολουθεί να διατηρεί αυτήν την ιδιότητα και στο τέλος του βήματος εκλέπτυνσης. Το

Αλγόριθμος 8 Η διαδικασία **SUPPLYMAXWITNESS**, μέρος της απόδειξης του θεωρήματος 2.3.2

Είσοδος: x

Έξοδος: σύνολο συμβολοακολουθιών που περιλαμβάνει το μέγιστο μάρτυρα του x, αν υπάρχει

procedure SUPPLYMAXWITNESS(x)
    $\Lambda \leftarrow \{[0^{p(|x|)}, 1^{p(|x|)}]\}$
    loop $p(|x|)$ φορές
        διαμέρισε κάθε στοιχείο του Λ σε δύο ισομήκη διαστήματα
        if $|\Lambda| > r(|x|)$ **then**
            $\tau \leftarrow []$ ▶ κενή λίστα υποθέσεων
            $Y \leftarrow$ **CULL**(x, Λ, τ)
            $\Lambda \leftarrow Y$
        end if
    end loop
    return $\{u \in \Sigma^{p(|x|)} : [u, u] \in \Lambda\}$
end procedure

Λ στον αλγόριθμο 8 αρχικοποιείται με $\Lambda \leftarrow \{[0^{p(|x|)}, 1^{p(|x|)}]\}$, το οποίο είναι πιστό για το x ως προς A, p. Με βάση τις προηγούμενες παρατηρήσεις, το Λ εξακολουθεί να είναι πιστό για το x ως προς A, p και μετά την ολοκλήρωση των $p(|x|)$ επαναλήψεων των βημάτων διαμέρισης και εκλέπτυνσης. Συνεπώς το σύνολο που επιστρέφει ο αλγόριθμος για την **SUPPLYMAXWITNESS** όντως περιέχει τον ${}^{A,p}\mathrm{maxwitness}(x)$, αν αυτός υπάρχει. Τέλος, επειδή τα p, r είναι πολυώνυμα και η **CULL** έχουμε υποθέσει ότι εκτελείται σε πολυωνυμικό χρόνο, ολόκληρη η διαδικασία **SUPPLYMAXWITNESS** εκτελείται σε πολυωνυμικό χρόνο.

Στη συνέχεια της απόδειξης κατασκευάζουμε την τελευταία διαδικασία που χρειαζόμαστε, την **CULL**. Αρχικά εισάγουμε κάποιους συμβολισμούς. Η συνάρτηση f απεικονίζει κάθε συμβολοακολουθία του $\Sigma^\star$ σε μία k-συνθήκη. Από αυτήν την απεικόνιση προκύπτει και μία απεικόνιση των διαστημάτων επί του $\Sigma^{p(|x|)}$ σε k-συνθήκες. Την εικόνα ενός διαστήματος I υπό αυτήν την απεικόνιση θα τη συμβολίζουμε με $f[I]$, και το διάστημα $I = [u, v]$ θα απεικονίζεται στην k-συνθήκη $f[I] = f(\langle x, u\rangle)$. Αν $f[I] = (\alpha, w_1, w_2, \ldots, w_k)$, θα συμβολίζουμε το w_i με $Q[I, i]$, για κάθε i με $1 \leq i \leq k$.

Ακόμη, έστω ότι έχουμε μία λίστα υποθέσεων με μέγεθος $|\tau| = d$, $\tau = [(w_1, b_1), (w_2, b_2), \ldots, (w_d, b_d)]$, όπου $d \leq k$, και ένα διάστημα I για το οποίο η k-συνθήκη $f[I] = (\alpha, v_1, v_2, \ldots, v_k)$ περιλαμβάνει όλα τα $\{w_i\}_{1\leq i\leq d}$ ως ερωτήσεις. Έστω $\mathcal{P} \subseteq \{1, 2, \ldots, k\}$ το σύνολο των δεικτών που αποτελούν τις θέσεις στις οποίες εμφανίζονται οι ερωτήσεις w_i στην k-συνθήκη. Ο πίνακας αληθείας της $f[I]$ είναι μία boolean συνάρτηση k ορισμάτων. Αν σταθεροποιήσουμε d από αυτά τα ορίσματα και συγκεκριμένα αυτά που βρίσκονται στις θέσεις που περιέχονται στο σύνολο $\mathcal{P}$, κατά τέτοιο τρόπο ώστε για κάθε $i \in \{1, \ldots, d\}$ το όρισμα που βρίσκεται στη θέση που αντιστοιχεί στο ερώτημα w_i να σταθεροποιείται στην τιμή b_i, προκύπτει μία boolean συνάρτηση $k - d$ ορισμάτων που θα την συμβολίζουμε με $\beta_\tau[I]$. Αν $\{v_{j_1}, v_{j_2}, \ldots, v_{j_{k-d}}\}$ είναι το υποσύνολο των ερωτήσεων της $f[I]$ που δεν περιέχονται στη λίστα υποθέσεων τ, κατασκευάζουμε την $(k-d)$-συνθήκη $f_\tau[I] = (\beta_\tau[I], v_{j_1}, v_{j_2}, \ldots, v_{j_{k-d}})$. Το σύνολο των ερωτήσεων της νέας συνθήκης θα το συμβολίζουμε με $Q_\tau[I]$. Παρατηρούμε ότι αν η λίστα υποθέσεων τ είναι ορθή ως προς S, τότε η $f[I]$ ικανοποιείται από το S αν και μόνο αν η $f_\tau[I]$ ικανοποιείται από το S.

Η **CULL** πρέπει να είναι μία διαδικασία πολυωνυμικού, ως προς $|x|$, χρόνου η οποία δέχεται ως είσοδο το x, ένα σύνολο διαστημάτων Γ επί του $\Sigma^{p(|x|)}$ και μία λίστα υποθέσεων τ, για τα οποία ισχύουν οι εξής παραδοχές:

- Η λίστα υποθέσεων τ περιέχει το πολύ k στοιχεία: $|\tau| \leq k$.

- Επειδή το Γ προκύπτει από το $\{[0^{p(|x|)}, 1^{p(|x|)}]\}$ με τα βήματα διαμέρισης και εκλέπτυνσης που έχουμε δεί στη διαδικασία **SUPPLYMAXWITNESS**, μπορούμε να υποθέσουμε ότι το Γ περιέχει διαστήματα επί του $\Sigma^{p(|x|)}$ που είναι ανά δύο ξένα μεταξύ τους. Επίσης, $|\Gamma| > r_{|\tau|}(|x|)$. Η **SUPPLYMAXWITNESS** καλεί την **CULL** με $|\tau| = 0$ και $|\Gamma| > r(|x|) = r_0(|x|)$.

- Αν $\tau = [(w_1, b_1), (w_2, b_2), \ldots, (w_{|\tau|}, b_{|\tau|})]$, τότε κάθε συνθήκη $f[I]$, όπου $I \in \Gamma$, περιέχει τις $\{w_i\}_{1\leq i\leq|\tau|}$ ως ερωτήσεις.

Δεδομένης μίας τέτοιας εισόδου, πρέπει να επιστρέφει μία τ,S-εκλέπτυνση του Γ για το x, ως προς A, p. Η εκλέπτυνση αυτή πρέπει να έχει μέγεθος το πολύ $r_{|\tau|}(|x|)$.

Η διαδικασία **CULL** είναι αναδρομική ως προς το μέγεθος της λίστας υποθέσεων της εισόδου και περιγράφεται από τον αλγόριθμο 9, όπου χρησιμοποιούνται ως υπορουτίνες οι αλγόριθμοι 10 και 11.

Η βάση της αναδρομής είναι η περίπτωση $|\tau| = k$. Σε αυτήν την περίπτωση, για κάθε διάστημα $I \in \Gamma$ η συνάρτηση του πίνακα αληθείας $\beta_\tau[I]$ είναι σταθερά 0 ή 1. Η **CULL** επιστρέφει το μονοσύνολο που περιέχει το μέγιστο λεξικογραφικά διάστημα του Γ που απεικονίζεται μέσω της $f_\tau[\cdot]$ σε 0-συνθήκη με πίνακα αληθείας σταθερά ίσο με 1. Αν δεν υπάρχει τέτοιο διάστημα στο Γ, επιστρέφει το κενό σύνολο. Δηλαδή αν R' είναι το σύνολο που επιστρέφει η **CULL**, έχουμε:

$$R' = \begin{cases} \emptyset & \text{, αν } R = \emptyset \\ \{maxR\} & \text{, αν } R \neq \emptyset \end{cases}$$

Αλγόριθμος 9 Η διαδικασία **CULL**, μέρος της απόδειξης του θεωρήματος 2.3.2

Είσοδος: x, Γ, τ με $|\Gamma| > r_{|\tau|}(|x|)$ και $|\tau| \leq k$

Έξοδος: τ,S-εκλέπτυνση του Γ για το x, ως προς A, p με μέγεθος το πολύ $r_{|\tau|}(|x|)$

procedure CULL(x, Γ, τ)
    ▶ Βάση της αναδρομής
    if $|\tau| = k$ **then**
        $R \leftarrow \{I \in \Gamma : \beta_\tau[I] \equiv 1\}$
        if $|R| = 0$ **then**
            return $\emptyset$
        else
            return $\{\max R\}$ ▶ κατά τη λεξικογραφική διάταξη
        end if
    else if $|\tau| < k$ **then**
        ▶ Φάση 1η: απαλοιφή διαστημάτων με τον ίδιο "μειωμένο" πίνακα αληθείας
        while $(\exists I, I' \in \Gamma)[I \prec I' \wedge f_\tau[I] = f_\tau[I']]$ **do**
            βρες το μικρότερο λεξικογραφικά I για το οποίο υπάρχει τέτοιο ζεύγος (I, I')
            $\Gamma \leftarrow \Gamma \setminus \{I\}$
        end while
        ▶ Φάση 2η: Εύρεση μίας εκλέπτυνσης του Γ που απομένει
        $\Delta_0 \leftarrow \{I \in \Gamma : \beta_\tau[I](0, \ldots, 0) = 0\}$
        $\Delta_1 \leftarrow \{I \in \Gamma : \beta_\tau[I](0, \ldots, 0) = 1\}$
        ▶ Εκλέπτυνση του Δ_0
        $\mathcal{Y}_0 \leftarrow$ **AUX_REF0**(Δ_0) ▶ βλ. αλγόριθμο 10
        ▶ Εκλέπτυνση του Δ_1
        $\mathcal{Y}_1 \leftarrow$ **AUX_REF1**(Δ_1) ▶ βλ. αλγόριθμο 11
        return $\mathcal{Y}_0 \cup \mathcal{Y}_1$
    end if
end procedure

Αλγόριθμος 10 Η διαδικασία **AUX_REF0**, μέρος της απόδειξης του θεωρήματος 2.3.2

procedure AUX_REF0(Δ_0)
  $m \leftarrow 0$
  repeat
    $S_{m+1} \leftarrow \{I \in \Delta_0 : (\forall j : 1 \leq j \leq m)[Q_\tau[I] \cap Q_\tau[I_j] = \emptyset]\}$
    if $S_{m+1} \neq \emptyset$ **then**
      $I_{m+1} \leftarrow \min S_{m+1}$
      $m \leftarrow m + 1$
    end if
  until $S_{m+1} = \emptyset$
  if $m > q(|x|)$ **then**
    $\Delta_0 \leftarrow \{J \in \Delta_0 : J \prec I_{q(|x|)+1}\}$
    $m \leftarrow q(|x|)$
  end if
  $R \leftarrow \bigcup_{1 \leq j \leq m} Q_\tau[I_j]$
  $h \leftarrow |R|$
  έστω $y_1, y_2, \ldots, y_h$ μία απαρίθμηση του R κατά τη λεξικογραφική διάταξη
  $\mathcal{Y}_0 \leftarrow \emptyset$
  for $j \leftarrow 1$ **to** h **do**
    $\Theta_j \leftarrow \{I \in \Delta_0 : (\forall s < j)[I \notin \Theta_s] \wedge y_j \in Q_\tau[I]\}$
    $\Theta'_{j,0} \leftarrow$ **CULL**$(x, \Theta_j, \tau \circ (y_j, 0))$
    $\Theta'_{j,1} \leftarrow$ **CULL**$(x, \Theta_j, \tau \circ (y_j, 1))$
    $\mathcal{Y}_0 \leftarrow \mathcal{Y}_0 \cup \Theta'_{j,0} \cup \Theta'_{j,1}$
  end for
  return $\mathcal{Y}_0$
end procedure

Αλγόριθμος 11 Η διαδικασία **AUX_REF1**, μέρος της απόδειξης του θεωρήματος 2.3.2

procedure AUX_REF1(Δ_1)
  $m \leftarrow 0$
  repeat
    $S_{m+1} \leftarrow \{I \in \Delta_1 : (\forall j : 1 \leq j \leq m)[Q_\tau[I] \cap Q_\tau[I_j] = \emptyset]\}$
    if $S_{m+1} \neq \emptyset$ **then**
      $I_{m+1} \leftarrow \max S_{m+1}$
      $m \leftarrow m + 1$
    end if
  until $S_{m+1} = \emptyset$
  if $m > q(|x|) + 1$ **then**
    $\Delta_1 \leftarrow \{J \in \Delta_1 : J \succeq I_{q(|x|)+1}\}$
    $m \leftarrow q(|x|) + 1$
  end if
  $R \leftarrow \bigcup_{1 \leq j \leq m} Q_\tau[I_j]$
  $h \leftarrow |R|$
  έστω $y_1, y_2, \ldots, y_h$ μία απαρίθμηση του R κατά τη λεξικογραφική διάταξη
  $\mathcal{Y}_1 \leftarrow \emptyset$
  for $j \leftarrow 1$ **to** h **do**
    $\Theta_j \leftarrow \{I \in \Delta_1 : (\forall s < j)[I \notin \Theta_s] \wedge y_j \in Q_\tau[I]\}$
    $\Theta'_{j,0} \leftarrow$ **CULL**$(x, \Theta_j, \tau \circ (y_j, 0))$
    $\Theta'_{j,1} \leftarrow$ **CULL**$(x, \Theta_j, \tau \circ (y_j, 1))$
    $\mathcal{Y}_1 \leftarrow \mathcal{Y}_1 \cup \Theta'_{j,0} \cup \Theta'_{j,1}$
  end for
  return $\mathcal{Y}_1$
end procedure

όπου $R = \{I \in \Gamma : \beta_\tau[I] \equiv 1\}$ είναι το σύνολο που υπολογίζει αρχικά η **CULL**. Το R' είναι τ,S-εκλέπτυνση του Γ για το x ως προς A, p. Πραγματικά, έστω ότι η τ είναι ορθή ως προς το S και ότι το Γ είναι πιστό για το x ως προς A, p. Διακρίνουμε δύο περιπτώσεις:

- Αν $R = \emptyset$, τότε για κάθε $I \in \Gamma$ ισχύει $\beta_\tau[I] \equiv 0$. Άρα το S δεν ικανοποιεί την $f_\tau[I]$. Εφόσον έχουμε υποθέσει ότι η τ είναι ορθή ως προς το S, προκύπτει ότι το S δεν ικανοποιεί ούτε την $f[I]$. Αν $I = [u, v]$, και επειδή $f[I] = f(\langle x, u\rangle)$, έχουμε ότι το S δεν ικανοποιεί την $f(\langle x, u\rangle)$, άρα $\langle x, u\rangle \notin {}^{A,p}\mathrm{Left}$. Αν $x \in {}^{A,p}L$, τότε $(\forall I = [u, v] \in \Gamma)[u \succ {}^{A,p}\mathrm{maxwitness}(x)]$ το οποίο σημαίνει ότι $(\forall I \in \Gamma)[{}^{A,p}\mathrm{maxwitness}(x) \notin I]$. Όμως έχουμε υποθέσει ότι το Γ είναι πιστό για το x ως προς A, p, άρα $x \notin {}^{A,p}L$. Επομένως οποιοδήποτε σύνολο διαστημάτων, άρα και το $R' = \emptyset$ είναι πιστό για το x ως προς A, p.

- Αν $R \neq \emptyset$, τότε ορίζεται το μέγιστο λεξικογραφικά διάστημα του R: $\max R = [u_m, v_m]$. Αφού $\max R \in R$, έχουμε $\beta_\tau[\max R] \equiv 1$ άρα η $f_\tau[\max R]$ ικανοποιείται από το S και επειδή η τ είναι ορθή ως προς το S, ικανοποιείται και η $f[\max R]$ από το S. Άρα $\langle x, u_m\rangle \in {}^{A,p}\mathrm{Left}$ δηλαδή ορίζεται ο ${}^{A,p}\mathrm{maxwitness}(x)$ και $u_m \preceq {}^{A,p}\mathrm{maxwitness}(x)$. Επειδή το Γ είναι πιστό για το x ως προς A, p, ο μέγιστος μάρτυρας του x ως προς A, p πρέπει να ανήκει σε κάποιο διάστημα του Γ. Πρέπει λοιπόν να ανήκει στο $\max R$, γιατί για όλα τα $J = [u_J, v_J] \in \Gamma$ με $J \succ I$, και κατά συνέπεια $\beta_\tau[J] \equiv 0$, έχουμε $u_J \succ {}^{A,p}\mathrm{maxwitness}(x)$, άρα ${}^{A,p}\mathrm{maxwitness}(x) \notin J$. Επομένως το $R' = \{maxR\}$ είναι πιστό για το x ως προς A, p.

Εάν η **CULL** κληθεί με $|\tau| < k$, αρχικά αφαιρεί διαστήματα από το Γ έτσι ώστε τα διαστήματα που θα παραμείνουν στο Γ να απεικονίζονται ανά δύο σε διαφορετικές $(k - |\tau|)$-συνθήκες μέσω της $f_\tau[\cdot]$. Το σύνολο Γ που απομένει είναι τ,S-εκλέπτυνση του αρχικού Γ για το x ως προς A, p. Συγκεκριμένα, αν υπάρχουν $I = [u, v], I' = [u', v'] \in \Gamma$ με $I \prec I'$ και $f_\tau[I] = f_\tau[I']$, θα δείξουμε ότι το $\Gamma \setminus \{I\}$ είναι τ,S-εκλέπτυνση του Γ για το x ως προς A,

p. Έστω ότι η τ είναι ορθή ως προς το S. Επειδή $f_\tau[I] = f_\tau[I']$, η $f[I]$ ικανοποιείται από το S αν και μόνο αν η $f[I']$ ικανοποιείται από το S. Άρα από το θεώρημα 2.3.7, $\langle x, u\rangle \in {}^{A,p}\mathrm{Left} \iff \langle x, u'\rangle \in {}^{A,p}\mathrm{Left}$. Η τελευταία σχέση γράφεται ισοδύναμα $u \preceq {}^{A,p}\mathrm{maxwitness}(x) \iff u' \preceq {}^{A,p}\mathrm{maxwitness}(x)$. Από αυτό προκύπτει ότι ${}^{A,p}\mathrm{maxwitness}(x) \notin I$, διότι σε αντίθετη περίπτωση ισχύει $u \preceq {}^{A,p}\mathrm{maxwitness}(x) \preceq v \prec u' \Rightarrow u' \preceq {}^{A,p}\mathrm{maxwitness}(x) \prec u'$, που είναι άτοπο. Αφού ${}^{A,p}\mathrm{maxwitness}(x) \notin I$, προφανώς το $\Gamma \setminus I$ είναι εκλέπτυνση του Γ για το x ως προς A, p. Άρα από το θέωρημα 2.3.22, ιδιότητα 2, το σύνολο Γ που προκύπτει στο τέλος της πρώτης φάσης είναι τ,S-εκλέπτυνση του αρχικού Γ για το x ως προς A, p.

Στη δεύτερη φάση, το σύνολο Γ διαμερίζεται σε δύο υποσύνολα Δ_0 και Δ_1, καθένα από τα οποία εκλεπτύνεται ξεχωριστά. Λόγω του θεωρήματος 2.3.22, ιδιότητα 1, η ένωση των δύο εκλεπτύνσεων είναι εκλέπτυνση του Γ. Αρχικά η **CULL** υπολογίζει μία αύξουσα ακολουθία διαστημάτων $\{I_i = [u_i, v_i]\}_{1\le i\le m}$ του $\Delta_0 = \{I \in \Gamma : \beta_\tau[I](0,\ldots,0) = 0\}$ έτσι ώστε να ισχύουν:

- Τα σύνολα $\{Q_\tau[I_i]\}_{1\le i\le m}$ είναι ανά δύο ξένα μεταξύ τους και
- για κάθε t, $0 \le t < m$, το I_{t+1} είναι το μικρότερο λεξικογραφικά διάστημα $I \in \Delta_0$ για το οποίο ισχύει $Q_\tau[I] \cap Q_\tau[I_1] = \emptyset$, $Q_\tau[I] \cap Q_\tau[I_2] = \emptyset$, $\ldots$, $Q_\tau[I] \cap Q_\tau[I_t] = \emptyset$.

Αν το πλήθος m αυτών των διαστημάτων είναι μεγαλύτερο από $q(|x|)$, τότε κρατάει στο Δ_0 μόνο όσα διαστήματα είναι μικρότερα λεξικογραφικά από το $(q(|x|)+1)$-οστό της ακολουθίας. Δηλαδή, αν Δ_0' είναι το νέο σύνολο Δ_0:

$$\Delta_0' = \begin{cases} \Delta_0 & \text{, αν } m \le q(|x|) \\ \{J \in \Delta_0 : J \prec I_{q(|x|)+1}\} & \text{, αν } m \ge q(|x|)+1 \end{cases}$$

Το Δ_0' είναι τ,S-εκλέπτυνση του Δ_0. Αρκεί να δείξουμε ότι αν $m \ge q(|x|)+1$ και αν η τ είναι ορθή ως προς S τότε ο μέγιστος μάρτυρας ${}^{A,p}\mathrm{maxwitness}(x)$ δεν μπορεί να περιέχεται σε κάποιο διάστημα J με $J \succeq I_{q(|x|)+1}$. Προς άτοπο, ας υποθέσουμε ότι $x \in {}^{A,p}L$ και $u_{q(|x|)+1} \preceq {}^{A,p}\mathrm{maxwitness}(x)$. Τότε για κάθε j, $1 \le j \le q(|x|)+1$, έχουμε $u_j \preceq {}^{A,p}\mathrm{maxwitness}(x)$. Άρα το S

ικανοποιεί την $f_\tau[I_j]$. Επειδή όμως $\beta_\tau[I_j](0,\ldots,0) = 0$, πρέπει τουλάχιστον μία από τις ερωτήσεις της $f_\tau[I_j]$ να περιέχεται στο S. Επειδή όπως είδαμε τα σύνολα $\{Q_\tau[I_j]\}_{1\le j\le m}$ είναι ανά δύο ξένα μεταξύ τους, συμπεραίνουμε ότι το S περιέχει τουλάχιστον $q(|x|)+1$ διαφορετικές μεταξύ τους ερωτήσεις k-συνθηκών, οι οποίες προκύπτουν ως απεικόνιση μέσω της $f(\cdot)$ κάποιου $\langle x,y\rangle$, όπου $|y| = p(|x|)$. Όμως αυτό είναι άτοπο, γιατί το πολυώνυμο q φράσσει άνω ακριβώς αυτό το μέγεθος (ορισμός 2.3.23).

Στη συνέχεια, ας είναι m' το πλήθος των διαστημάτων του Δ_0'. Δηλαδή $m' = \min\{m, q(|x|)\}$. Το σύνολο $R = \bigcup_{1\le j\le m'} Q_\tau[I_j]$ όλων των ερωτήσεων που εμφανίζονται στις μειωμένες συνθήκες $\{f_\tau[I_j]\}_{1\le j\le m'}$ έχει πληθάριθμο $h = |R| = (k-|\tau|)m'$. Αν $y_1, y_2, \ldots, y_h$ είναι η διάταξη των στοιχείων του R σε λεξικογραφική σειρά, ορίζουμε τα σύνολα $\{\Theta_j\}_{1\le j\le h}$, όπου για κάθε j, $1\le j\le h$:

$$\Theta_j = \{I \in \Delta_0' : (\forall s : 1 \le s \le j-1)[I \notin \Theta_s] \wedge (y_j \in Q_\tau[I])\}$$

Δηλαδή το σύνολο Θ_j περιέχει διαστήματα που οι αντίστοιχες μειώμενες συνθήκες τους δεν περιλαμβάνουν ως ερωτήσεις καμία από τις $\{y_i\}_{1\le i<j}$ αλλά περιλαμβάνουν την y_j. Η **CULL** υπολογίζει τα $\{\Theta_j\}_{1\le j\le h}$ και χρησιμοποιεί αναδρομικά τον εαυτό της για να υπολογίσει το σύνολο $\mathcal{Y}_0$:

$$\mathcal{Y}_0 = \bigcup_{1\le j\le h}\ \bigcup_{b\in\{0,1\}} \mathbf{CULL}(x, \Theta_j, \tau\circ(y_j,b))$$

Το $\mathcal{Y}_0$ είναι τ,S-εκλέπτυνση του Δ_0' για το x ως προς A, p. Πραγματικά, παρατηρούμε ότι για κάθε j, το $\bigcup_{b\in\{0,1\}} \mathbf{CULL}(x, \Theta_j, \tau\circ(y_j,b))$ είναι τ,S-εκλέπτυνση του Θ_j για το x ως προς A, p. Επειδή $\Delta_0' = \Theta_1 \cup \Theta_2 \cup \cdots \cup \Theta_h$, από το θεώρημα 2.3.22 προκύπτει τελικά ότι το $\mathcal{Y}_0$ είναι τ,S-εκλέπτυνση του Δ_0 για το x ως προς A, p.

Για την εκλέπτυνση του Δ_1 ακολουθούνται παρόμοια βήματα. Αρχικά υπολογίζεται μία φθίνουσα ακολουθία διαστημάτων $\{I_i = [u_i, v_i]\}_{1\le i\le m}$ του $\Delta_1 = \{I \in \Gamma : \beta_\tau[I](0,\ldots,0) = 1\}$, για την οποία ισχύει και πάλι ότι τα σύνολα $\{Q_\tau[I_i]\}_{1\le i\le m}$ είναι ανά δύο ξένα μεταξύ τους, αλλά για κάθε t, $0\le t<m$, το I_{t+1} είναι το μεγαλύτερο λεξικογραφικά διάστημα $I\in\Delta_1$ για

το οποίο ισχύει $Q_\tau[I] \cap Q_\tau[I_1] = \emptyset$, $Q_\tau[I] \cap Q_\tau[I_2] = \emptyset$, ..., $Q_\tau[I] \cap Q_\tau[I_t] = \emptyset$. Αν το πλήθος m αυτών των διαστημάτων είναι μεγαλύτερο από $q(|x|)+1$, τότε κρατάει στο Δ_1 μόνο όσα διαστήματα είναι μεγαλύτερα ή ίσα λεξικογραφικά από το $(q(|x|)+1)$-οστό της ακολουθίας. Δηλαδή, αν Δ_1' είναι το νέο σύνολο Δ_1:

$$\Delta_1' = \begin{cases} \Delta_1 & \text{, αν } m \leq q(|x|)+1 \\ \{J \in \Delta_1 : J \succeq I_{q(|x|)+1}\} & \text{, αν } m \geq q(|x|)+2 \end{cases}$$

Το Δ_1' είναι τ,S-εκλέπτυνση του Δ_1. Αρκεί να δείξουμε ότι αν $m \geq q(|x|)+2$ και αν η τ είναι ορθή ως προς S τότε ο μέγιστος μάρτυρας ${}^{A,p}\mathrm{maxwitness}(x)$ δεν μπορεί να περιέχεται σε κάποιο διάστημα J με $J \prec I_{q(|x|)+1}$. Προς άτοπο, ας υποθέσουμε ότι $x \in {}^{A,p}L$ και $u_{q(|x|)+1} \succ {}^{A,p}\mathrm{maxwitness}(x)$. Τότε για κάθε j, $1 \leq j \leq q(|x|)+1$, έχουμε: $u_j \succ {}^{A,p}\mathrm{maxwitness}(x)$ άρα το S δεν ικανοποιεί την $f_\tau[I_j]$. Επειδή όμως $\beta_\tau[I_j](0,\ldots,0) = 1$, πρέπει τουλάχιστον μία από τις ερωτήσεις της $f_\tau[I_j]$ να περιέχεται στο S. Επειδή και πάλι τα σύνολα $\{Q_\tau[I_j]\}_{1 \leq j \leq m}$ είναι ανά δύο ξένα μεταξύ τους, προκύπτει με τον ίδιο τρόπο όπως και στην περίπτωση του Δ_0' ότι το S περιέχει περισσότερες από $q(|x|)$ διαφορετικές μεταξύ τους ερωτήσεις k-συνθηκών $f(\langle x, y\rangle)$ όπου $|y| = p(|x|)$, που είναι άτοπο.

Έχουμε δείξει μέχρι στιγμής ότι το σύνολο που επιστρέφει η **CULL** σε είσοδο (x, Γ, τ) είναι τ,S-εκλέπτυνση του Γ για το x ως προς A, p. Θα δείξουμε στη συνέχεια ότι το μέγεθος αυτής της εκλέπτυνσης δεν ξεπερνάει την τιμή $r_{|\tau|}(|x|)$. Η απόδειξη θα γίνει με επαγωγή στο μήκος της λίστας υποθέσεων τ. Αν $|\tau| = k$, τότε η **CULL** επιστρέφει ένα σύνολο που περιέχει μόνο ένα στοιχείο. Άρα το ζητούμενο ισχύει, αφού $r_{|\tau|} \equiv r_k \equiv 1$. Έστω τώρα ότι για κάθε d, $0 \leq d_0 < d \leq k$, η **CULL** με είσοδο (x, Γ, τ), όπου $|\tau| = d$, επιστρέφει μία τ,S-εκλέπτυνση του Γ για το x ως προς A, p με μέγεθος το πολύ $r_d(|x|)$. Αν η **CULL** κληθεί με είσοδο λίστα υποθέσεων που έχει μήκος d_0, εκτελεί το πολύ $2q(|x|)(k-d_0)$ αναδρομικές κλήσεις με μήκος λίστας υποθέσεων d_0+1 για την εκλέπτυνση του Δ_0', και το πολύ $2(q(|x|)+1)(k-d_0)$ αναδρομικές κλήσεις με μήκος λίστας υποθέσεων d_0+1 για την εκλέπτυνση του Δ_1'. Τελικά, επιστρέφει ένα σύνολο που είναι η ένωση των αποτελεσμάτων αυτών των κλήσεων, άρα λόγω της επαγωγικής υπόθεσης μπορεί να περιέχει το πολύ $2(k - d_0)(2q(|x|) + 1)r_{d_0+1}(|x|) = r_{d_0}(|x|)$ διαστήματα.

■

Κεφάλαιο 3

Συναρτήσεις μονής κατεύθυνσης

Οι συναρτήσεις μονής κατεύθυνσης αποτελούν αντικείμενο βασικής σημασίας στην κρυπτογραφία. Σε αυτές στηρίζονται τα κρυπτοσυστήματα δημοσίου κλειδιού, όπου ο παραλήπτης ενός κρυπτογραφημένου μηνύματος μπορεί να το αποκρυπτογραφήσει εύκολα χρησιμοποιώντας το ιδιωτικό του κλειδί, ενώ για κάποιον που δεν γνωρίζει αυτό το κλειδί η διαδικασία της εξαγωγής του αρχικού μηνύματος από το κρυπτογραφημένο είναι δύσκολη. Γενικά, λέγεται ότι μία συνάρτηση f είναι μονής κατεύθυνσης όταν είναι υπολογιστικά "προσιτός" ο υπολογισμός της τιμής $f(x)$ δοθέντος του x, ενώ ο αντίστροφος υπολογισμός του x δοθείσης της τιμής $f(x)$ είναι υπολογιστικά "απρόσιτος". Συνήθως ταυτίζουμε τα υπολογιστικά προσιτά προβλήματα με την κλάση πολυπλοκότητας $\mathbf{P}$ και τα υπολογιστικά απρόσιτα προβλήματα με την κλάση $\mathbf{NP} \setminus \mathbf{P}$. Επειδή δεν είναι γνωστή η απάντηση στο ερώτημα $\mathbf{P} \stackrel{?}{=} \mathbf{NP}$, δεν υπάρχουν αποδεδειγμένα μονής κατεύθυνσης συναρτήσεις.

Στα επόμενα θα ορίσουμε πιο αυστηρά τις συναρτήσεις μονής κατεύθυνσης και θα δούμε ότι το ερώτημα της ύπαρξης συναρτήσεων μονής κατεύθυνσης είναι ισοδύναμο με το $\mathbf{P} \stackrel{?}{=} \mathbf{NP}$. Ακόμη, θα εξετάσουμε την ύπαρξη αμφιμονοσήμαντων συναρτήσεων μονής κατεύθυνσης και θα διερευνήσουμε την έννοια της συνάρτησης μονής κατεύθυνσης με δύο ορίσματα.

3.1 Ύπαρξη συναρτήσεων μονής κατεύθυνσης

3.1.1 Αποσαφήνιση των εννοιών

Με βάση τη διαισθητική περιγραφή που δώσαμε για τις συναρτήσεις μονής κατεύθυνσης, θα μπορούσαμε να ορίσουμε ότι πρόκειται για τις συναρτήσεις που είναι υπολογιστές σε πολυωνυμικό χρόνο ενώ η αντίστροφη συνάρτηση κάθε μίας από αυτές δεν μπορεί να υπολογιστεί σε πολυωνυμικό χρόνο. Το πρόβλημα με τον ορισμό αυτό είναι ότι η δυσκολία αντιστροφής ενδέχεται να οφείλεται απλώς στο ότι η συνάρτηση παράγει έξοδο με πολύ μικρότερο μήκος από την είσοδο. Για παράδειγμα, η συνάρτηση $f(x) = 1^{\lceil \log(|x|+1) \rceil}$ παράγει έξοδο με μήκος $|f(x)| = O(\log |x|)$. Άρα για την αντίστροφη συνάρτηση f^{-1} θα έχουμε ότι $|f^{-1}(x)| = O(2^{|x|})$. Δηλαδή μία μηχανή Turing που υπολογίζει την f^{-1} θα απαιτεί εκθετικό χρόνο ως προς το μήκος της εισόδου απλώς και μόνο για την εγγραφή του αποτελέσματος στην ταινία εξόδου. Επειδή τέτοιου είδους συναρτήσεις δεν προσφέρουν τίποτε στην όλη συζήτηση στα πλαίσια της κρυπτογραφίας ή της θεωρίας πολυπλοκότητας, απαιτούμε για κάθε τιμή y μίας συνάρτησης μονής κατεύθυνσης να υπάρχει τουλάχιστον μία προ-εικόνα x με μήκος το πολύ πολυωνυμικό ως προς $|y|$, $|x| \leq q(|y|)$. Αν μία συνάρτηση ικανοποιεί αυτή τη συνθήκη, τότε λέγεται *τίμια.*

Ορισμός 3.1.1 *Μία μερική συνάρτηση f λέγεται* τίμια, *αν υπάρχει ένα πολυώνυμο q ώστε για κάθε y στο σύνολο τιμών της f ($\mathrm{range}(f)$) να υπάρχει μία συμβολοακολουθία x με μήκος το πολύ $q(|y|)$ για την οποία ισχύει $f(x) = y$.*

Ορισμός 3.1.2 *Μία μερική συνάρτηση f λέγεται* αντιστρέψιμη σε πολυωνυμικό χρόνο *αν υπάρχει μία συνάρτηση $g \in \mathbf{FP}$ έτσι ώστε για κάθε y στο σύνολο τιμών της f, η y να ανήκει στο πεδίο ορισμού της g ($\mathrm{domain}(g)$), η $g(y)$ να ανήκει στο πεδίο ορισμού της f και $f(g(y)) = y$. Η g λέγεται* αντίστροφη *της f.*

Ορισμός 3.1.3 *Μία μερική συνάρτηση f ονομάζεται συνάρτηση* μονής κατεύθυνσης *αν ισχύουν όλα τα παρακάτω:*

1. η f είναι υπολογιστή σε πολυωνυμικό χρόνο,

2. η f δεν είναι αντιστρέψιμη σε πολυωνυμικό χρόνο και

3. η f είναι τίμια.

Με αυτόν τον ορισμό αποκλείουμε την περίπτωση κατά την οποία η f δεν είναι αντιστρέψιμη σε πολυωνυμικό χρόνο για τετριμμένους λόγους και εξασφαλίζουμε ότι η αντιστροφή των συναρτήσεων μονής κατεύθυνσης που χρησιμοποιούμε παρουσιάζει ουσιαστική υπολογιστική δυσκολία.

3.1.2 Η τεχνική των συναρτήσεων μονής κατεύθυνσης

Θεώρημα 3.1.4 *[Wat88] Υπάρχουν συναρτήσεις μονής κατεύθυνσης αν και μόνο αν* $\mathbf{P} \neq \mathbf{NP}$.

Απόδειξη

Έστω ότι $\mathbf{P} \neq \mathbf{NP}$ και ας είναι A μία γλώσσα που ανήκει στο $\mathbf{NP}$ αλλά όχι στο $\mathbf{P}$. Υπάρχει μία μη ντετερμινιστική μηχανή Turing N πολυωνυμικού χρόνου που αποδέχεται την A, δηλαδή $A = L(N)$. Θεωρούμε το σύνολο των ζευγών των υπολογιστικών μονοπατιών της N που καταλήγουν σε κατάσταση αποδοχής και των αντίστοιχων εισόδων, $\mathrm{comp}(N)$:

$$\mathrm{comp}(N) = \left\{ \langle x, y \rangle : \begin{array}{c} \text{η } x \text{ είναι κωδικοποίηση υπολογιστικού μονοπατιού της} \\ N \text{ με είσοδο } y \text{ που καταλήγει σε κατάσταση αποδοχής} \end{array} \right\}$$

Επειδή η N λειτουργεί σε πολυωνυμικό χρόνο και υποτίθεται ότι η $\langle \cdot, \cdot \rangle$ είναι λογική[1] συνάρτηση σύζευξης, κάθε στοιχείο $\langle x, y \rangle$ του $\mathrm{comp}(N)$ θα έχει το πολύ πολυωνυμικό μήκος ως προς $|y|$. Επίσης είναι φανερό ότι $\mathrm{comp}(N) \in \mathbf{P}$, αφού σε είσοδο $\langle x, y \rangle$ αρκεί να εξομοιωθεί η N με είσοδο y και με τις μη ντετερμινιστικές επιλογές που καθορίζονται από το x και να ελεγχθεί αν καταλήγει σε κατάσταση αποδοχής ή όχι.

[1]Μία λογική συνάρτηση σύζευξης είναι μία ένα-προς-ένα και επί συνάρτηση από το $\Sigma^\star \times \Sigma^\star$ στο $\Sigma^\star$, η οποία είναι υπολογιστή σε πολυωνυμικό χρόνο και αντιστρέψιμη σε πολυωνυμικό χρόνο. Αυτό έχει ως αποτέλεσμα για κάθε $x, y \in \Sigma^\star$ τα μεγέθη $|x| + |y|$ και $|\langle x, y \rangle|$ να είναι πολυωνυμικά συσχετισμένα.

Επιπλέον, θεωρούμε τη συνάρτηση f:

$$f(\langle x, y\rangle) = \begin{cases} y & , \text{ αν } \langle x, y\rangle \in \mathrm{comp}(N) \\ \text{απροσδιόριστη} & , \text{ αλλιώς} \end{cases}$$

Η f είναι μία μερική συνάρτηση που έχει τις ακόλουθες ιδιότητες:

- Είναι υπολογιστή σε πολυωνυμικό χρόνο, αφού $\mathrm{comp}(N) \in \mathbf{P}$.
- $\mathrm{range}(f) = L(N) = A$. Πραγματικά, αν $y \in \mathrm{range}(f)$ τότε υπάρχει ένα υπολογιστικό μονοπάτι x της N με είσοδο y που καταλήγει σε κατάσταση αποδοχής, άρα $y \in L(N)$. Αντίστροφα, αν $y \in L(N)$ και x είναι κάποιο υπολογιστικό μονοπάτι της N με είσοδο y που καταλήγει σε κατάσταση αποδοχής, τότε $f(\langle x, y\rangle) = y$, άρα $y \in \mathrm{range}(f)$.
- Είναι τίμια: αν y ανήκει στο σύνολο τιμών της, τότε υπάρχει μία συμβολοακολουθία $\langle x, y\rangle \in \mathrm{comp}(N)$ ώστε $f(\langle x, y\rangle) = y$, και μάλιστα το μήκος της $\langle x, y\rangle$ είναι το πολύ πολυωνυμικό ως προς $|y|$.

Θα δείξουμε επιπλέον ότι η f δεν είναι αντιστρέψιμη σε πολυωνυμικό χρόνο, άρα είναι συνάρτηση μονής κατεύθυνσης.

Προς άτοπο, έστω ότι η f είναι αντιστρέψιμη σε πολυωνυμικό χρόνο και έστω $g \in \mathbf{FP}$ η αντίστροφή της. Εξ ορισμού, για κάθε $y \in \mathrm{range}(f)$ ισχύει ότι $y \in \mathrm{domain}(g)$, $g(y) \in \mathrm{domain}(f)$ και $f(g(y)) = y$. Άρα για κάθε συμβολοακολουθία y έχουμε την ισοδυναμία:

$$y \in \mathrm{range}(f) \Leftrightarrow f(g(y)) = y$$

Επειδή οι συναρτήσεις f και g είναι υπολογιστές σε πολυωνυμικό χρόνο, προκύπτει $\mathrm{range}(f) \in \mathbf{P} \Rightarrow A \in \mathbf{P}$. Η τελευταία σχέση οδηγεί σε άτοπο, αφού έχουμε υποθέσει ότι $A \in \mathbf{NP} \setminus \mathbf{P}$.

Για την αντίστροφη κατεύθυνση, ας υποθέσουμε ότι υπάρχει μία συνάρτηση μονής κατεύθυνσης f, και προς άτοπο έστω ότι $\mathbf{P} = \mathbf{NP}$. Θα δείξουμε ότι η f είναι αντιστρέψιμη σε πολυωνυμικό χρόνο, γεγονός που αντίκειται στον ορισμό της συνάρτησης μονής κατεύθυνσης.

Αλγόριθμος 12 Μη ντετερμινιστικός αλγόριθμος πολυωνυμικού χρόνου που αποφασίζει το σύνολο prefix-inv(f), όπου η f είναι συνάρτηση μονής κατεύθυνσης

Είσοδος: $\langle x, y \rangle$

επέλεξε μη ντετερμινιστικά μία συμβολοακολουθία z μήκους το πολύ $q(|y|)$
if η x δεν είναι πρόθεμα της z **then**
    τερμάτισε την εκτέλεση σε κατάσταση απόρριψης
end if
if $f(z) \neq y$ **then**
    τερμάτισε την εκτέλεση σε κατάσταση απόρριψης
end if
τερμάτισε την εκτέλεση σε κατάσταση αποδοχής

Η f είναι τίμια, άρα υπάρχει ένα πολυώνυμο q ώστε για κάθε $y \in \mathrm{range}(f)$ να υπάρχει x με μήκος το πολύ $q(|y|)$, τέτοιο ώστε $f(x) = y$. Θεωρούμε το σύνολο "προθεμάτων" της αντίστροφης συνάρτησης της f, prefix-inv(f):

$$\text{prefix-inv}(f) = \{\langle x, y \rangle : (\exists w)[|xw| \leq q(|y|) \wedge f(xw) = y]\}$$

Δηλαδή το σύνολο prefix-inv(f) περιέχει όλα τα ζεύγη $\langle x, y \rangle$ όπου το x είναι πρόθεμα κάποιας προ-εικόνας του y με μέγεθος το πολύ πολυωνυμικό ως προς $|y|$. Ο μη ντετερμινιστικός αλγόριθμος 12 είναι πολυωνυμικού χρόνου, αφού η f ως συνάρτηση μονής κατεύθυνσης υπολογίζεται σε πολυωνυμικό χρόνο, και αποφασίζει το prefix-inv(f). Άρα prefix-inv(f) $\in$ **NP**, και επειδή έχουμε υποθέσει ότι $\mathbf{P} = \mathbf{NP}$ προκύπτει τελικά ότι prefix-inv(f) $\in$ **P**.

Χρησιμοποιώντας τη χαρακτηριστική συνάρτηση του prefix-inv(f) ως υπορουτίνα πολυωνυμικού χρόνου, μπορούμε πλέον να αντιστρέψουμε τη συνάρτηση f σε (ντετερμινιστικό) πολυωνυμικό χρόνο με τον αλγόριθμο 13: ο βρόχος του αλγορίθμου εκτελείται το πολύ $q(|y|)$ φορές και η f είναι υπολογιστή σε πολυωνυμικό χρόνο, ως συνάρτηση μονής κατεύθυνσης.

Ο βρόχος διατηρεί την αναλλοίωτη συνθήκη:

$$y \in \mathrm{range}(f) \Rightarrow \langle x, y \rangle \in \text{prefix-inv}(f)$$

Αλγόριθμος 13 Ντετερμινιστικός αλγόριθμος πολυωνυμικού χρόνου που υπολογίζει την αντίστροφη μίας συνάρτησης μονής κατεύθυνσης, υπό τις προϋποθέσεις του θεωρήματος 3.1.4

Είσοδος: y

$x \leftarrow \varepsilon$
loop
    if $\langle x0, y\rangle \in \text{prefix-inv}(f)$ **then**
        $x \leftarrow x0$
    else if $\langle x1, y\rangle \in \text{prefix-inv}(f)$ **then**
        $x \leftarrow x1$
    else
        break
    end if
end loop
if $f(x) = y$ **then**
    return x
else
    τερμάτισε την εκτέλεση σε κατάσταση απόρριψης ▶ $y \notin \text{range}(f)$
end if

Στο τέλος του βρόχου, αν $y \in \mathrm{range}(f)$, ισχύουν $\langle x, y\rangle \in \text{prefix-inv}(f)$, $\langle x0, y\rangle \notin \text{prefix-inv}(f)$ και $\langle x1, y\rangle \notin \text{prefix-inv}(f)$. Άρα εξ ορισμού του prefix-inv(f) ισχύει $f(x) = y$. Αντίστροφα, αν $y \notin \mathrm{range}(f)$ τότε προφανώς $f(x) \neq y$. Άρα στο τέλος του βρόχου ισχύει:

$$y \in \mathrm{range}(f) \Leftrightarrow f(x) = y$$

Αν g είναι η συνάρτηση που υπολογίζει ο αλγόριθμος 13, έχουμε ότι για κάθε $y \in \mathrm{range}(f)$:

- $y \in \mathrm{domain}(g)$, αφού η g επιστρέφει την τιμή της μεταβλητής x στο τέλος του βρόχου.
- $g(y) \in \mathrm{domain}(f)$, αφού $g(y) = x$ και $f(x) = y$.
- $f(g(y)) = f(x) = y$.

Άρα η f είναι αντιστρέψιμη σε πολυωνυμικό χρόνο.

■

3.2 Ένα-προς-ένα και k-προς-ένα συναρτήσεις μονής κατεύθυνσης

Σχετικά με την αντιστροφή των συναρτήσεων μονής κατεύθυνσης, θα ήταν γενικά επιθυμητό κάθε κρυπτογραφημένο μήνυμα να έχει μόνο μία δυνατή αποκωδικοποίηση, χωρίς όμως να τίθεται σε κίνδυνο η ασφάλεια του χρησιμοποιούμενου κρυπτοσυστήματος. Έτσι οδηγούμαστε στην έννοια της ένα-προς-ένα συνάρτησης μονής κατεύθυνσης:

Ορισμός 3.2.1 *Μία μερική συνάρτηση $f : \Sigma^\star \to \Sigma^\star$ ονομάζεται* ένα-προς-ένα συνάρτηση *ή* αμφιμονοσήμαντη συνάρτηση *αν για κάθε $y \in \Sigma^\star$ υπάρχει το πολύ ένα $x \in \Sigma^\star$ για το οποίο ισχύει $f(x) = y$.*

Ορισμός 3.2.2 *Μία μερική συνάρτηση ονομάζεται* ένα-προς-ένα συνάρτηση μονής κατεύθυνσης *αν είναι ταυτόχρονα ένα-προς-ένα συνάρτηση και συνάρτηση μονής κατεύθυνσης.*

Παρατηρούμε ότι οι ένα-προς-ένα συναρτήσεις είναι εντελώς μονοσήμαντες ως προς την αντιστροφή. Αν η f είναι ένα-προς-ένα συνάρτηση, κάθε στοιχείο του $\mathrm{range}(f)$ έχει ακριβώς μία προ-εικόνα. Θα δούμε στα επόμενα ότι η υπάρξη ένα-προς-ένα συναρτήσεων μονής κατεύθυνσης συνδέεται άμεσα με τους μονοσήμαντους υπολογισμούς μηχανών Turing που αποφασίζουν γλώσσες της κλάσης **UP**. Ακόμη, θα συνδέσουμε άρρηκτα την ύπαρξη ένα-προς-ένα συνάρτησεων μονής κατεύθυνσης με την ύπαρξη k-προς-ένα συναρτήσεων μονής κατεύθυνσης, όπου $k \geq 1$ φυσική σταθερά.

Ορισμός 3.2.3 *Για κάθε φυσικό $k \geq 1$, μία μερική συνάρτηση f ονομάζεται* k-προς-ένα συνάρτηση *αν για κάθε $y \in \mathrm{range}(f)$ υπάρχουν το πολύ k το πλήθος x για τα οποία ισχύει $f(x) = y$.*

Ορισμός 3.2.4 *Αν υπάρχει φυσικός αριθμός $k \geq 1$ ώστε κάποια μερική συνάρτηση f να είναι k-προς-ένα συνάρτηση, τότε η f λέγεται* συνάρτηση φραγμένης αμφισημίας.

3.2.1 Υπολογισμοί περιορισμένης αμφισημίας

Μέχρι στιγμής έχουμε χρησιμοποιήσει για τους μη ντετερμινιστικούς υπολογισμούς ένα υπολογιστικό μοντέλο στο οποίο η μηχανή Turing αποδέχεται την είσοδό της αν και μόνο αν υπάρχει τουλάχιστον ένα μονοπάτι του δέντρου υπολογισμού που καταλήγει σε κατάσταση αποδοχής. Όπως έχουμε δει, οι γλώσσες που γίνονται αποδεκτές από τέτοιες μηχανές που λειτουργούν σε πολυωνυμικό χρόνο αποτελούν την κλάση πολυπλοκότητας **NP**.

Τροποποιούμε το μοντέλο αυτό, απαιτώντας από τον υπολογισμό της μηχανής Turing να έχει για κάθε είσοδο το πολύ ένα μονοπάτι που καταλήγει σε κατάσταση αποδοχής. Οι μηχανές αυτές ονομάζονται *μονοσήμαντες*. Η κλάση των γλωσσών που γίνονται αποδεκτές από μονοσήμαντες μη ντετερμινιστικές μηχανές Turing πολυωνυμικού χρόνου ονομάζεται **UP** [Val76]. Φαίνεται εύκολα ότι $\mathbf{P} \subseteq \mathbf{UP} \subseteq \mathbf{NP}$. Πραγματικά, ο υπολογισμός μίας ντετερμινιστικής μηχανής Turing έχει πάντα το πολύ ένα μονοπάτι που καταλήγει σε κατάσταση αποδοχής. Για τον εγκλεισμό $\mathbf{UP} \subseteq \mathbf{NP}$ αρκεί να παρατηρήσουμε ότι μία μονοσήμαντη μη ντετερμινιστική μηχανή Turing εξακολουθεί να είναι μη ντετερμινιστική μηχανή Turing.

Δεν είναι γνωστό αν κάποιος από τους εγκλεισμούς αυτούς είναι γνήσιος, όμως πιστεύεται ότι $\mathbf{P} \subset \mathbf{UP} \subset \mathbf{NP}$. Θα δούμε στη συνέχεια ότι το ερώτημα $\mathbf{P} \stackrel{?}{=} \mathbf{UP}$ συνδέεται άμεσα με την ύπαρξη ένα-προς-ένα συναρτήσεων μονής κατεύθυνσης. Γενικεύοντας τις μονοσήμαντες μηχανές Turing, μπορούμε να ορίσουμε, για κάθε φυσικό $k \geq 1$, τις k-αμφίσημες μηχανές Turing στις οποίες ο μη ντετερμινιστικός υπολογισμός επιτρέπεται να έχει το πολύ k μονοπάτια που καταλήγουν σε αποδοχή. Η κλάση των γλωσσών που γίνονται αποδεκτές από k-αμφίσημες μηχανές Turing ονομάζεται $\mathbf{UP}_k$ [Bei89]. Προφανώς η $\mathbf{UP}_1$ ταυτίζεται με την **UP**. Αντίστοιχα, αν επιτρέψουμε για κάθε είσοδο x το πολύ πολυωνυμικό ως προς $|x|$ αριθμό μαρτύρων, λαμβάνουμε την κλάση **FewP** [AR88], η οποία βρίσκεται ανάμεσα στις **UP** και **NP**: $\mathbf{P} \subseteq \mathbf{UP} \subseteq \mathbf{FewP} \subseteq \mathbf{NP}$. Ακολουθούν οι ορισμοί των κλάσεων στις οποίες αναφερόμαστε:

Ορισμός 3.2.5 *Μία γλώσσα L ανήκει στην κλάση* **UP** *αν και μόνο αν υπάρχει μη ντετερμινιστική μηχανή Turing N πολυωνυμικού χρόνου, τέτοια ώστε να ισχύει $L = L(N)$ και για κάθε x να υπάρχει το πολύ ένα υπολογιστικό μονοπάτι της N που καταλήγει σε κατάσταση αποδοχής.*

Για τις κλάσεις $\mathbf{UP}_k$ και **FewP** δίνουμε τον ισοδύναμο ορισμό σε όρους μαρτύρων, σε αντιστοιχία με τον ισοδύναμο ορισμό για την κλάση **NP** που προκύπτει από το θεώρημα 2.3.8:

Ορισμός 3.2.6 *Για κάθε φυσικό αριθμό $k \geq 1$, μία γλώσσα L ανήκει στην κλάση $\mathbf{UP}_k$ αν και μόνο αν υπάρχει ένα πολυώνυμο p και μία γλώσσα $A \in \mathbf{P}$, έτσι ώστε για κάθε x να ισχύουν τα εξής:*

1. $x \notin L \Rightarrow |\{y : |y| \leq q(|x|) \wedge \langle x, y \rangle \in A\}| = 0$ *και*

2. $x \in L \Rightarrow 1 \leq |\{y : |y| \leq q(|x|) \wedge \langle x, y \rangle \in A\}| \leq k$

Ορισμός 3.2.7 *Μία γλώσσα L ανήκει στην κλάση* **FewP** *αν και μόνο αν υπάρχουν δύο πολυώνυμα q, p και μία γλώσσα $A \in \mathbf{P}$, έτσι ώστε για κάθε x να ισχύουν τα εξής:*

1. $x \notin L \Rightarrow |\{y : |y| \leq q(|x|) \wedge \langle x, y \rangle \in A\}| = 0$ *και*

2. $x \in L \Rightarrow 1 \leq |\{y : |y| \leq q(|x|) \wedge \langle x, y \rangle \in A\}| \leq p(|x|)$

Ο ορισμός της κλάσης **UP** θέτει έναν περιορισμό στη λειτουργία των μηχανών Turing που αποδέχονται τις γλώσσες της κλάσης. Συγκεκριμένα, απαιτεί να έχουν για κάθε είσοδο το πολύ ένα υπολογιστικό μονοπάτι που καταλήγει σε κατάσταση αποδοχής. Αυτή η κατάσταση είναι διαφορετική από την κατάσταση κατά την οποία η μηχανή Turing επιτρέπεται να έχει οποιονδήποτε αριθμό μονοπατιών που καταλήγουν σε κατάσταση αποδοχής, αλλά τελικά η είσοδος γίνεται αποδεκτή αν και μόνο αν υπάρχει ακριβώς ένα τέτοιο μονοπάτι. Σε αυτήν την περίπτωση έχουμε μία καθ'όλα κανονική μη ντετερμινιστική μηχανή Turing πολυωνυμικού χρόνου, αλλά διαφορετική (σε σχέση με την κλάση **NP**) συνθήκη αποδοχής της εισόδου με βάση το πλήθος των μονοπατιών που οδηγούν σε κατάσταση αποδοχής. Η κλάση των γλωσσών L για κάθε μία από τις οποίες υπάρχει μη ντετερμινιστική μηχανή Turing N πολυωνυμικού χρόνου ώστε να ισχύει

$$L = \left\{ x : \begin{array}{l} \text{ο υπολογισμός της } N \text{ με είσοδο } x \text{ έχει ακριβώς ένα μονοπάτι} \\ \text{που οδηγεί σε κατάσταση αποδοχής} \end{array} \right\}$$

ονομάζεται **US** [BG82]. Για να τονίσουμε τη σχέση μεταξύ των κλάσεων **NP** και **US**, θα αντιπαραβάλουμε τους ορισμούς τους με όρους μαρτύρων:

Ορισμός 3.2.8 *Μία γλώσσα L ανήκει στην κλάση* **NP** *αν και μόνο αν υπάρχει ένα πολυώνυμο q και μία γλώσσα $A \in \mathbf{P}$ ώστε για κάθε x να ισχύει:*

$$x \in L \Leftrightarrow |\{y : |y| \leq q(|x|) \wedge \langle x, y \rangle \in A\}| \geq 1$$

Ορισμός 3.2.9 *Μία γλώσσα L ανήκει στην κλάση* **US** *αν και μόνο αν υπάρχει ένα πολυώνυμο q και μία γλώσσα $A \in \mathbf{P}$ ώστε για κάθε x να ισχύει:*

$$x \in L \Leftrightarrow |\{y : |y| \leq q(|x|) \wedge \langle x, y\rangle \in A\}| = 1$$

Είναι γνωστό ότι η κλάση **US** έχει πλήρη προβλήματα, και ένα από αυτά είναι το πρόβλημα μοναδικής ικανοποιησιμότητας

$$\mathbf{USAT} = \{\phi : \text{υπάρχει ακριβώς μία ανάθεση που ικανοποιεί την } \phi\}.$$

Αντίθετα, δεν γνωρίζουμε αν η κλάση **UP** έχει πλήρη προβλήματα. Έχει όμως αποδειχτεί ότι αν υπάρχουν πλήρεις γλώσσες για την **UP**, τότε έχει μία πλήρη γλώσσα της μορφής $L = \mathbf{SAT} \cap A$, όπου $A \in \mathbf{P}$ [HH88]. Τέλος, οι κλάσεις **UP**, **NP** και **US** σχετίζονται ως εξής:

$$\mathbf{UP} \subseteq \mathbf{NP} \cap \mathbf{US} \subseteq \mathbf{NP} \subseteq \mathbf{coUS}$$

3.2.2 Αμφιμονοσήμαντες συναρτήσεις μονής κατεύθυνσης

Θεώρημα 3.2.10 *[Ber77, Ko85, GS88] Υπάρχουν ένα-προς-ένα συναρτήσεις μονής κατεύθυνσης αν και μόνο αν* $\mathbf{P} \neq \mathbf{UP}$.

Απόδειξη

Έστω ότι $\mathbf{P} \neq \mathbf{UP}$ και έστω A μία γλώσσα που ανήκει στην κλάση **UP** αλλά όχι στην **P**. Υπάρχει λοιπόν μία μη ντετερμινιστική μηχανή Turing N που λειτουργεί σε πολυωνυμικό χρόνο, για κάθε είσοδο ο υπολογισμός της παράγει το πολύ ένα μονοπάτι που καταλήγει σε κατάσταση αποδοχής, και ισχύει $A = L(N)$.

Θεωρούμε τη συνάρτηση f:

$$f(\langle x, y\rangle) = \begin{cases} 0y & , \text{ αν } \langle x, y\rangle \in \mathrm{comp}(N) \\ 1\langle x, y\rangle & , \text{ αλλιώς} \end{cases}$$

όπου το σύνολο $\mathrm{comp}(N)$ ορίζεται με τον ίδιο τρόπο που ορίστηκε και στο θεώρημα 3.1.4. Η f έχει τις ακόλουθες ιδιότητες:

Αλγόριθμος 14 Ντετερμινιστικός αλγόριθμος πολυωνυμικού χρόνου που αποφασίζει μία γλώσσα $A \in \mathbf{UP} \setminus \mathbf{P}$, υπό τις προϋποθέσεις του θεωρήματος 3.2.10

Είσοδος: y

$z \leftarrow g(0y)$
ερμήνευσε την z ως κωδικοποίηση ενός ζεύγους $\langle x, w \rangle$
if $\langle x, w \rangle \in \mathrm{comp}(N)$ **then**
    τερμάτισε την εκτέλεση σε κατάσταση αποδοχής
else
    τερμάτισε την εκτέλεση σε κατάσταση απόρριψης
end if

- Είναι ένα-προς-ένα, γιατί η f απεικονίζει κάθε μονοπάτι του υπολογισμού που καταλήγει σε κατάσταση απόρριψης, σε διαφορετική συμβολοακολουθία. Αν $0y \in \mathrm{range}(f)$, τότε υπάρχει x ώστε $\langle x, y \rangle \in \mathrm{comp}(N)$. Η μηχανή Turing N είναι μονοσήμαντη, άρα το x αυτό είναι μοναδικό, και η μοναδική προ-εικόνα του $0y$ είναι η $\langle x, y \rangle$. Αν $1z \in \mathrm{range}(f)$, η μοναδική προ-εικόνα του $1z$ είναι το ίδιο το z.

- Είναι υπολογιστή σε πολυωνυμικό χρόνο, αφού $\mathrm{comp}(N) \in \mathbf{P}$.

- Είναι τίμια: αν y ανήκει στο σύνολο τιμών της, τότε υπάρχει μία συμβολοακολουθία $\langle x, y \rangle \in \mathrm{comp}(N)$ ώστε $f(\langle x, y \rangle) = y$, και μάλιστα το μήκος της $\langle x, y \rangle$ είναι το πολύ πολυωνυμικό ως προς $|y|$.

Για να δείξουμε το ζητούμενο, αρκεί να δείξουμε ότι η f δεν είναι αντιστρέψιμη σε πολυωνυμικό χρόνο. Προς άτοπο, έστω ότι $g \in \mathbf{FP}$ είναι η αντίστροφή της. Τότε ο αλγόριθμος 14 αποφασίζει την A σε ντετερμινιστικό πολυωνυμικό χρόνο. Επειδή έχουμε υποθέσει ότι $A \notin \mathbf{P}$, προκύπτει τελικά ότι η f είναι ένα-προς-ένα συνάρτηση μονής κατεύθυνσης.

Για την αντίστροφη κατεύθυνση, ας είναι η f μία ένα-προς-ένα συνάρτηση μονής κατεύθυνσης, και προς άτοπο έστω ότι $\mathbf{P} = \mathbf{UP}$. Θεωρούμε το σύνολο $\mathrm{prefix\text{-}inv}(f)$ που κατασκευάστηκε στην απόδειξη του θεωρήματος 3.1.4 και τον αλγόριθμο 12 που το αποφασίζει σε πολυωνυμικό χρόνο. Επειδή η f

είναι ένα-προς-ένα, για κάθε $y \in \mathrm{range}(f)$ υπάρχει ακριβώς ένα z τέτοιο ώστε $f(z) = y$. Άρα το δέντρο υπολογισμού του αλγορίθμου 12 θα έχει το πολύ ένα υπολογιστικό μονοπάτι που καταλήγει σε κατάσταση αποδοχής, γεγονός που σημαίνει ότι prefix-inv$(f) \in \mathbf{UP}$, άρα prefix-inv$(f) \in \mathbf{P}$. Επομένως ο αλγόριθμος 13 μπορεί να χρησιμοποιηθεί για την αντιστροφή της f, που είναι άτοπο γιατί έχουμε υποθέσει ότι η f είναι συνάρτηση μονής κατεύθυνσης. Συνεπώς $\mathbf{P} \neq \mathbf{UP}$.

■

3.2.3 Συναρτήσεις μονής κατεύθυνσης φραγμένης αμφισημίας

Η ομοιότητα μεταξύ των αποδείξεων των θεωρημάτων 3.1.4 και 3.2.10 είναι πολύ εμφανής. Και στις δύο αποδείξεις εφαρμόζεται η τεχνική των συναρτήσεων μονής κατεύθυνσης, για την οποία θα έχουμε να πούμε περισσότερα αφού δούμε την εφαρμογή της στο βασικό αποτέλεσμα της ενότητας που ασχολείται με συναρτήσεις μονής κατεύθυνσης δύο ορισμάτων. Αποδεικνύεται με παρόμοιο τρόπο και το ακόλουθο θεώρημα, που χαρακτηρίζει για κάθε $k \geq 1$ την ύπαρξη k-προς-ένα συναρτήσεων μονής κατεύθυνσης:

Θεώρημα 3.2.11 *[HZ93] Για κάθε $k \geq 1$, υπάρχουν k-προς-ένα συναρτήσεις μονής κατεύθυνσης αν και μόνο αν $\mathbf{P} \neq \mathbf{UP}_k$.*

Είδαμε ότι η ύπαρξη ένα-προς-ένα συναρτήσεων μονής κατεύθυνσης χαρακτηρίζεται πλήρως από τη συνθήκη $\mathbf{P} \neq \mathbf{UP}$. Αντίστοιχα, η ύπαρξη συναρτήσεων μονής κατεύθυνσης χωρίς περιορισμό αμφισημίας χαρακτηρίζεται πλήρως από τη συνθήκη $\mathbf{P} \neq \mathbf{NP}$. Είναι γνωστό ότι $\mathbf{P} \neq \mathbf{UP} \Rightarrow \mathbf{P} \neq \mathbf{NP}$, όμως δεν είναι γνωστό αν ισχύει η αντίστροφη συνεπαγωγή. Επομένως δεν αποκλείεται η περίπτωση να υπάρχουν συναρτήσεις μονής κατεύθυνσης, αλλά να μην υπάρχουν ένα-προς-ένα συναρτήσεις μονής κατεύθυνσης. Αντίθετα, όπως θα δούμε αμέσως, η ύπαρξη k-προς-ένα συναρτήσεων μονής κατεύθυνσης, για κάποιο σταθερό k, συνεπάγεται την ύπαρξη ένα-προς-ένα συναρτήσεων μονής κατεύθυνσης.

Θεώρημα 3.2.12 *[Wat88] Υπάρχουν αμφιμονοσήμαντες συναρτήσεις μονής κατεύθυνσης αν και μόνο αν υπάρχουν συναρτήσεις μονής κατεύθυνσης φραγμένης αμφισημίας.*

Απόδειξη

Κάθε ένα-προς-ένα συνάρτηση είναι συνάρτηση φραγμένης αμφισημίας, άρα η ευθεία κατεύθυνση της ισοδυναμίας ισχύει.

Για την αντίστροφη κατεύθυνση, θα αποδείξουμε ότι για κάθε $k \geq 1$ ισχύει η συνεπαγωγή:

$$\mathbf{P} = \mathbf{UP} \Rightarrow \mathbf{P} = \mathbf{UP}_k$$

Έτσι, αν υπάρχει ένα k ώστε να υπάρχει k-προς-ένα συνάρτηση μονής κατεύθυνσης, λόγω του θεωρήματος 3.2.11 έπεται $\mathbf{P} \neq \mathbf{UP}_k$, άρα $\mathbf{P} \neq \mathbf{UP}$, άρα από το θεώρημα 3.2.10 θα υπάρχουν ένα-προς-ένα συναρτήσεις μονής κατεύθυνσης.

Η απόδειξη της πρότασης $(\forall k \geq 1)[\mathbf{P} = \mathbf{UP} \Rightarrow \mathbf{P} = \mathbf{UP}_k]$ θα γίνει με επαγωγή στο k. Η πρόταση ισχύει προφανώς για $k = 1$, αφού εξ ορισμού $\mathbf{UP} \equiv \mathbf{UP}_1$. Έστω ότι για κάποιο $k \geq 1$ ισχύει $\mathbf{P} = \mathbf{UP} \Rightarrow \mathbf{P} = \mathbf{UP}_k$. Θα δείξουμε ότι ισχύει και η συνεπαγωγή $\mathbf{P} = \mathbf{UP} \Rightarrow \mathbf{P} = \mathbf{UP}_{k+1}$.

Προς τούτο, έστω ότι $\mathbf{P} = \mathbf{UP}$ και ας είναι L μία αυθαίρετη γλώσσα της κλάσης $\mathbf{UP}_{k+1}$. Υπάρχει λοιπόν μία μη ντετερμινιστική μηχανή Turing N που λειτουργεί σε πολυωνυμικό χρόνο και αποδέχεται την L, ενώ για κάθε είσοδο έχει το πολύ $k+1$ μονοπάτια που οδηγούν σε κατάσταση αποδοχής. Ας θεωρήσουμε τα εξής σύνολα:

$$B = \left\{ x : \begin{array}{l} \text{το δέντρο υπολογισμού της } N \text{ με είσοδο } x \text{ έχει ακριβώς} \\ k+1 \text{ μονοπάτια που οδηγούν σε κατάσταση αποδοχής} \end{array} \right\}$$

$$D = \{x : x \notin B \wedge x \in L(N)\}$$

Το σύνολο B ανήκει στην κλάση $\mathbf{UP}$. Η μη ντετερμινιστική μηχανή Turing N_B που το αποδέχεται, σε είσοδο x επιλέγει μη ντετερμινιστικά μία διατεταγμένη $(k+1)$-άδα συμβολοσειρών που κωδικοποιούν υπολογιστικά μονοπάτια της N με είσοδο x, και αποδέχεται αν και μόνο αν όλα καταλήγουν σε κατάσταση αποδοχής. Επειδή η N ποτέ δεν έχει περισσότερα από $k+1$ μονοπάτια που καταλήγουν σε κατάσταση αποδοχής, προκύπτει ότι το πολύ μία επιλεγμένη $(k+1)$-άδα θα οδηγήσει την N_B σε κατάσταση αποδοχής. Με βάση την υπόθεση που έχουμε κάνει, ότι $\mathbf{P} = \mathbf{UP}$, έχουμε ότι $B \in \mathbf{P}$.

Εφόσον $B \in \mathbf{P}$, το σύνολο D ανήκει στην κλάση $\mathbf{UP}_k$. Πραγματικά, η μη ντετερμινιστική μηχανή Turing N_D που το αποδέχεται, σε είσοδο x αρχικά ελέγχει σε ντετερμινιστικό πολυωνυμικό χρόνο αν $x \in B$, και αν ισχύει απορρίπτει το x. Αν $x \notin B$, τότε εξομοιώνει απ'ευθείας την N με είσοδο x. Η εξομοίωση αυτή θα δώσει το πολύ k μονοπάτια που οδηγούν σε κατάσταση αποδοχής, διότι αν έδινε $k{+}1$ το x θα άνηκε στο σύνολο B. Εφόσον $D \in \mathbf{UP}_k$, από την επαγωγική υπόθεση και την υπόθεση ότι $\mathbf{P} = \mathbf{UP}$ έπεται $D \in \mathbf{P}$.

Είναι γνωστό ότι η κλάση $\mathbf{P}$ είναι κλειστή ως προς την πράξη της ένωσης συνόλων, άρα $B \cup D \in \mathbf{P}$. Όμως $B \cup D = L$. Άρα $L \in \mathbf{P}$, και επειδή η L ήταν μία αυθαίρετα επιλεγμένη γλώσσα της κλάσης $\mathbf{UP}_{k+1}$, προκύπτει το ζητούμενο $\mathbf{P} = \mathbf{UP}_{k+1}$.

■

3.3 Συναρτήσεις μονής κατεύθυνσης με δύο ορίσματα

Στη μελέτη κρυπτογραφικών πρωτοκόλλων εμφανίζονται με πολύ φυσικό τρόπο συναρτήσεις μονής κατεύθυνσης με δύο ορίσματα. Οι Rabi, Rivest και Sherman ανέπτυξαν τέτοια πρωτόκολλα [She86, RS93, RS97] που χρησιμοποιούν συναρτήσεις μονής κατεύθυνσης με δύο ορίσματα, οι οποίες είναι επιπλέον ολικές, αντιμεταθετικές, προσεταιριστικές και ισχυρά μη αντιστρέψιμες. Το ερώτημα που τίθεται άμεσα είναι κατά πόσο υπάρχουν τέτοιες συναρτήσεις. Οι Rabi, Sherman έδειξαν ότι υπάρχουν (ασθενώς) αντιμεταθετικές και προσεταιριστικές συναρτήσεις μονής κατεύθυνσης με δύο ορίσματα αν και μόνο αν $\mathbf{P} \neq \mathbf{NP}$ [RS93, RS97]. Όμως οι συναρτήσεις αυτές δεν είναι ολικές και δεν είναι ισχυρά μη αντιστρέψιμες, ενώ η έννοια της ασθενούς προσεταιριστικότητας που χρησιμοποιούν δημιουργεί προβλήματα όταν εφαρμόζεται σε μη ολικές συναρτήσεις [HR99].

Σε αυτήν την ενότητα θα παρουσιάσουμε ένα αποτέλεσμα των Hemaspaandra και Rothe, που χαρακτηρίζει πλήρως την ύπαρξη ολικών, αντιμεταθετικών, προσεταιριστικών και ισχυρά μη αντιστρέψιμων συναρτήσεων μονής κατεύθυνσης με δύο ορίσματα. Συγκεκριμένα, θα δούμε ότι η ύπαρξη τέτοιων συναρτήσεων είναι ισοδύναμη με την ύπαρξη συναρτήσεων μονής κατεύθυνσης, η οποία όπως έχουμε δει (θεώρημα 3.1.4) ισοδυναμεί με $\mathbf{P} \neq \mathbf{NP}$. Επομένως, υποθέτοντας ότι ισχύει η ευρέως αποδεκτή εικασία $\mathbf{P} \neq \mathbf{NP}$, όντως υπάρχουν συναρτήσεις με τις ιδιότητες που απαιτούνται για τα πρωτόκολλα των Rabi, Rivest και Sherman. Για την απόδειξη αυτού του αποτελέσματος χρησιμοποιείται και πάλι η τεχνική των συναρτήσεων μονής κατεύθυνσης. Αρχικά, ας επεκτείνουμε την έννοια των συναρτήσεων μονής κατεύθυνσης σε συναρτήσεις με δύο ορίσματα.

Ορισμός 3.3.1 *Μία μερική συνάρτηση* $f : \Sigma^\star \times \Sigma^\star \to \Sigma^\star$ *με δύο ορίσματα λέγεται* τίμια *αν υπάρχει ένα πολυώνυμο* q *τέτοιο ώστε για κάθε* $y \in \mathrm{range}(f)$ *να υπάρχει ένα ζεύγος* (x, x') *με* $|x| + |x'| \leq q(|y|)$ *για το οποίο ισχύει* $f(x, x') = y$.

Ορισμός 3.3.2 *Μία μερική συνάρτηση* $f : \Sigma^\star \times \Sigma^\star \to \Sigma^\star$ *με δύο ορίσματα λέγεται* αντιστρέψιμη σε πολυωνυμικό χρόνο *αν υπάρχει συνάρτηση* $g \in \mathbf{FP}$ *ώστε για κάθε* $y \in \mathrm{range}(f)$ *να ισχύουν όλα τα παρακάτω:*

- $y \in \mathrm{domain}(g)$,
- $(\mathrm{first}(g(y)), \mathrm{second}(g(y))) \in \mathrm{domain}(f)$ *και*
- $f(\mathrm{first}(g(y)), \mathrm{second}(g(y))) = y$,

όπου οι προβολές $\mathrm{first}(z)$ *και* $\mathrm{second}(z)$ *επιστρέφουν το πρώτο και το δεύτερο, αντίστοιχα, στοιχείο του μοναδικού διατεταγμένου ζεύγους* (x, y) *για το οποίο ισχύει* $\langle x, y \rangle = z$.

Ορισμός 3.3.3 *Μία μερική συνάρτηση* $f : \Sigma^\star \times \Sigma^\star \to \Sigma^\star$ *με δύο ορίσματα ονομάζεται* συνάρτηση μονής κατεύθυνσης *αν ισχύουν όλα τα παρακάτω:*

1. *η f είναι υπολογιστή σε πολυωνυμικό χρόνο,*

2. *η f δεν είναι αντιστρέψιμη σε πολυωνυμικό χρόνο και*

3. *η f είναι τίμια.*

Όπως φαίνεται, οι ορισμοί αυτοί επεκτείνουν με τον αναμενόμενο τρόπο τους αντίστοιχους ορισμούς που δώσαμε για συναρτήσεις με ένα όρισμα. Η έννοια της ισχυρά μη αντιστρέψιμης συνάρτησης αναφέρεται μόνο στις συναρτήσεις με δύο ορίσματα. Διαισθητικά, το γεγονός ότι μία συνάρτηση είναι ισχυρά μη αντιστρέψιμη εξασφαλίζει ότι αν είναι γνωστή η τιμή της συνάρτησης και η τιμή ενός εκ των δύο ορισμάτων της, τότε η τιμή του δεύτερου ορίσματος δεν είναι δυνατόν να υπολογιστεί σε πολυωνυμικό χρόνο.

Ορισμός 3.3.4 *Μία μερική συνάρτηση $f : \Sigma^\star \times \Sigma^\star \to \Sigma^\star$ με δύο ορίσματα ονομάζεται* s-τίμια *αν ισχύουν οι εξής συνθήκες:*

1. *Υπάρχει πολυώνυμο q τέτοιο ώστε, για κάθε a, y για τα οποία ισχύει $f(a,b) = y$ για κάποιο b, να υπάρχει ένα b' με μήκος το πολύ $q(|y|+|a|)$ το οποίο να ικανοποιεί τη σχέση $f(a,b') = y$.*

2. *Υπάρχει πολυώνυμο q τέτοιο ώστε, για κάθε b, y για τα οποία ισχύει $f(a,b) = y$ για κάποιο a, να υπάρχει ένα a' με μήκος το πολύ $q(|y|+|b|)$ το οποίο να ικανοποιεί τη σχέση $f(a',b) = y$.*

Ορισμός 3.3.5 *Μία μερική συνάρτηση $f : \Sigma^\star \times \Sigma^\star \to \Sigma^\star$ με δύο ορίσματα ονομάζεται* ισχυρά μη αντιστρέψιμη, *εννοείται σε πολυωνυμικό χρόνο, αν είναι s-τίμια αλλά δεν ισχύει καμία από τις παρακάτω συνθήκες:*

1. *Υπάρχει συνάρτηση δύο ορισμάτων $g \in \mathbf{FP}$ ώστε για κάθε $y \in \mathrm{range}(f)$ και για κάθε προ-εικόνα (x_1, x_2) του y να ισχύουν $(y, x_1) \in \mathrm{domain}(g)$ και $f(x_1, g(y, x_1)) = y$.*

2. *Υπάρχει συνάρτηση δύο ορισμάτων $g \in \mathbf{FP}$ ώστε για κάθε $y \in \mathrm{range}(f)$ και για κάθε προ-εικόνα (x_1, x_2) του y να ισχύουν $(y, x_2) \in \mathrm{domain}(g)$ και $f(g(y, x_2), x_2) = y$.*

Οι ισχυρά μη αντιστρέψιμες συναρτήσεις απαιτείται να είναι s-τίμιες για να εξασφαλιστεί ότι η δυσκολία εύρεσης της τιμής ενός ορίσματος, δεδομένων των τιμών του άλλου ορίσματος και της εξόδου της συνάρτησης, δεν οφείλεται σε τετριμένους λόγους που έχουν σχέση με την παραγωγή τιμών εξόδου πολύ μικρού μήκους σε σχέση με τα ορίσματα, αλλά αντίθετα οφείλεται στην πραγματική υπολογιστική δυσκολία του προβλήματος της αντιστροφής. Σημειώνουμε εδώ ότι μία συνάρτηση όπως η

$$f(a,b) = \begin{cases} 1^{\lceil \log\log\log\max\{|a|,4\}\rceil} & , \text{ αν } |a| = |b| \\ \text{απροσδιόριστη} & , \text{ αλλιώς} \end{cases}$$

που συρρικνώνει ταυτόχρονα και τα δύο της ορίσματα σε μία τιμή εξόδου με λογαριθμικά μικρότερο μήκος είναι s-τίμια (το πολυώνυμο $q(n) = n$ ικανοποιεί και τις δύο συνθήκες του ορισμού 3.3.4), παρά το γεγονός ότι προφανώς δεν είναι τίμια. Τέλος, η αντιμεταθετικότητα και η προσεταιριστικότητα ορίζονται με το γνωστό τρόπο:

Ορισμός 3.3.6 *Μία ολική συνάρτηση $f : \Sigma^\star \times \Sigma^\star \to \Sigma^\star$ με δύο ορίσματα λέγεται* αντιμεταθετική *αν για κάθε $x, y \in \Sigma^\star$ ισχύει:*

$$f(x,y) = f(y,x)$$

Ορισμός 3.3.7 *Μία ολική συνάρτηση $f : \Sigma^\star \times \Sigma^\star \to \Sigma^\star$ με δύο ορίσματα λέγεται* προσεταιριστική *αν για κάθε $x, y, z \in \Sigma^\star$ ισχύει:*

$$f(f(x,y),z) = f(x, f(y,z))$$

Πριν περάσουμε στην απόδειξη του αποτελέσματος των Hemaspaandra και Rothe, θα δείξουμε ότι υπάρχουν συναρτήσεις μονής κατεύθυνσης με δύο ορίσματα αν και μόνο αν υπάρχουν συναρτήσεις μονής κατεύθυνσης με ένα όρισμα. Στη συνέχεια, χρησιμοποιώντας αυτό το ενδιάμεσο αποτέλεσμα, θα κατασκευάσουμε μία ολική, αντιμεταθετική, προσεταιριστική, ισχυρά μη αντιστρέψιμη συνάρτηση μονής κατεύθυνσης με δύο ορίσματα.

Θεώρημα 3.3.8 *Οι παρακάτω προτάσεις είναι ισοδύναμες:*

1. *Υπάρχουν συναρτήσεις μονής κατεύθυνσης (με ένα όρισμα).*

2. *Υπάρχουν συναρτήσεις μονής κατεύθυνσης με δύο ορίσματα.*

3. $\mathbf{P} \neq \mathbf{NP}$.

Απόδειξη

Η ισοδυναμία (1) $\Leftrightarrow$ (3) ισχύει λόγω του θεωρήματος 3.1.4.

Για τη συνεπαγωγή (2) $\Rightarrow$ (1), έστω $\langle \cdot, \cdot \rangle$ μία λογική συνάρτηση σύζευξης, η οποία έχει την επιπλέον ιδιότητα ότι αν σταθεροποιηθεί το ένα όρισμά της τότε είναι αύξουσα ως προς το άλλο όρισμά της (ως προς τη σχέση λεξικογραφικής διάταξης $\prec$). Ας είναι $f : \Sigma^\star \times \Sigma^\star \to \Sigma^\star$ μία συνάρτηση μονής κατεύθυνσης με δύο ορίσματα, και ας θεωρήσουμε τη συνάρτηση $g : \Sigma^\star \to \Sigma^\star$ που ορίζεται ως εξής:

$$g(z) = f(\text{first}(z), \text{second}(z))$$

Η g υπολογίζεται σε πολυωνυμικό χρόνο. Επιπλέον φαίνεται εύκολα ότι είναι τίμια και δεν είναι αντιστρέψιμη σε πολυωνυμικό χρόνο. Άρα η g είναι συνάρτηση μονής κατεύθυνσης.

Για τη συνεπαγωγή (1) $\Rightarrow$ (2), έστω ότι η $h : \Sigma^\star \to \Sigma^\star$ είναι συνάρτηση μονής κατεύθυνσης. Θεωρούμε την $h' : \Sigma^\star \times \Sigma^\star \to \Sigma^\star$ που ορίζεται ως εξής:

$$h'(x, y) = \langle h(x), h(y) \rangle$$

Φαίνεται εύκολα ότι η h' είναι συνάρτηση μονής κατεύθυνσης με δύο ορίσματα. Μάλιστα, είναι επιπλέον και ισχυρά μη αντιστρέψιμη.

■

Θεώρημα 3.3.9 *[HR99] Υπάρχουν ολικές, αντιμεταθετικές, προσεταιριστικές, ισχυρά μη αντιστρέψιμες συναρτήσεις μονής κατεύθυνσης με δύο ορίσματα αν και μόνο αν υπάρχουν συναρτήσεις μονής κατεύθυνσης.*

Απόδειξη

Αν υπάρχουν ολικές, αντιμεταθετικές, προσεταιριστικές, ισχυρά μη αντιστρέψιμες συναρτήσεις μονής κατεύθυνσης με δύο ορίσματα τότε ασφαλώς υπάρχουν συναρτήσεις μονής κατεύθυνσης με δύο ορίσματα. Άρα λόγω του θεωρήματος 3.3.8 υπάρχουν συναρτήσεις μονής κατεύθυνσης.

Για την αντίστροφη κατεύθυνση, και πάλι λόγω του θεωρήματος 3.3.8, αρκεί να δείξουμε ότι αν $\mathbf{P} \neq \mathbf{NP}$ τότε υπάρχουν ολικές, αντιμεταθετικές, προσεταιριστικές, ισχυρά μη αντιστρέψιμες συναρτήσεις μονής κατεύθυνσης με δύο ορίσματα.

Έστω λοιπόν ότι $\mathbf{P} \neq \mathbf{NP}$, και ας είναι η N μία μη ντετερμινιστική μηχανή Turing πολυωνυμικού χρόνου που αποδέχεται μία γλώσσα στο $\mathbf{NP} \setminus \mathbf{P}$: $L(N) \in \mathbf{NP} \setminus \mathbf{P}$. Μπορούμε να υποθέσουμε, χωρίς βλάβη της γενικότητας, ότι υπάρχει ένα γνησίως αύξον πολυώνυμο p για το οποίο ισχύει $(\forall n)[p(n) > n]$ και για κάθε είσοδο x όλα τα υπολογιστικά μονοπάτια της N έχουν μήκος ακριβώς $p(|x|)$. Όπως έχουμε συζητήσει και στην ενότητα 2.3.1, κάθε υπολογιστικό μονοπάτι μπορεί να θεωρηθεί ως εν δυνάμει μάρτυρας για το γεγονός ότι η συμβολοακολουθία εισόδου ανήκει στη γλώσσα που αποδέχεται η μηχανή Turing, και κάθε τέτοιο μονοπάτι που καταλήγει όντως σε κατάσταση αποδοχής για είσοδο x λέγεται μάρτυρας για το $x \in L(N)$. Αν $W(x)$ είναι το σύνολο όλων των μαρτύρων για μία συμβολοακολουθία εισόδου x τότε ισχύει η ισοδυναμία $x \in L(N) \Leftrightarrow W(x) \neq \emptyset$.

Ορίζουμε τη συνάρτηση δύο ορισμάτων f, με τύπο

$$f(u,v) = \begin{cases} \langle x, \min\{w_1, w_2\}\rangle & \text{αν } u = \langle x, w_1\rangle,\ v = \langle x, w_2\rangle \text{ και } w_1, w_2 \in W(x) \\ \langle x, x\rangle & \text{αν } \{u,v\} = \{\langle x,x\rangle, \langle x,w\rangle\},\ \text{όπου } w \in W(x) \\ \langle t, t1\rangle & \text{αλλιώς} \end{cases}$$

όπου t είναι μία σταθερή συμβολοακολουθία που δεν ανήκει στην $L(N)$ [2] και $t1$ συμβολίζει την παράθεση της συμβολοακολουθίας t και της συμβολοακολουθίας 1. Η f είναι καλώς ορισμένη, γιατί έχουμε υποθέσει ότι για κάθε x και για κάθε $w \in W(x)$ ισχύει $|w| > |x|$, άρα η περίπτωση $x \in W(x)$ δεν υφίσταται. Διαισθητικά, η f περιμένει τα δύο ορίσματά της να είναι συζεύξεις της ίδιας συμβολοακολουθίας x, είτε με τον εαυτό της είτε με κάποιον μάρτυρά της. Αν δεν ισχύει κάτι τέτοιο, η f δίνει ως έξοδο μία συγκεκριμένη συμβολοακολουθία που σηματοδοτεί το "λάθος". Αν η είσοδος έχει την

[2] Υπάρχει τέτοια συμβολοακολουθία, ειδάλλως θα ίσχυε $L(N) = \Sigma^\star \in \mathbf{P}$.

αναμενόμενη μορφή, τότε η f μειώνει κατά ένα τον αριθμό των εμφανίσεων μαρτύρων. Συγκεκριμένα, αν και τα δύο ορίσματα είναι ζεύγη της x και κάποιου μάρτυρα, επιστρέφει το ένα από αυτά και μάλιστα αυτό που περιέχει τον μικρότερο λεξικογραφικά μάρτυρα. Αντίθετα, αν το ένα όρισμα είναι σύζευξη της x με τον εαυτό της και το άλλο είναι σύζευξη της x με κάποιον μάρτυρα, τότε απορρίπτει το ζεύγος που περιέχει τον μάρτυρα και επιστρέφει την τιμή $\langle x, x\rangle$.

Επειδή, όπως έχουμε ήδη δει στην ενότητα 2.3.1, το σύνολο $\{\langle x, w\rangle : w \in W(x)\}$ ανήκει στην κλάση $\mathbf{P}$, η f είναι υπολογιστή σε πολυωνυμικό χρόνο. Επίσης, από τον ορισμό της είναι φανερό ότι είναι ολική συνάρτηση. Θα δείξουμε στη συνέχεια ότι είναι επιπλέον αντιμεταθετική, ισχυρά μη αντιστρέψιμη, μη αντιστρέψιμη σε πολυωνυμικό χρόνο, τίμια και προσεταιριστική, ολοκληρώνοντας έτσι την απόδειξη.

Η f είναι αντιμεταθετική:
Αν u, v είναι τέτοια ώστε να ισχύει η συνθήκη του δεύτερου ή του τρίτου κλάδου του ορισμού της f, τότε προφανώς $f(u, v) = f(v, u)$. Αν ισχύει η συνθήκη του πρώτου κλάδου, τότε επειδή $\min\{w_1, w_2\} = \min\{w_2, w_1\}$ και πάλι ισχύει $f(u, v) = f(v, u)$. Άρα για κάθε $u, v \in \Sigma^\star$ ισχύει $f(u, v) = f(v, u)$.

Η f είναι ισχυρά μη αντιστρέψιμη:
Οι δυνατές περιπτώσεις για τα ζεύγη (a, y) για τα οποία υπάρχει b ώστε $f(a, b) = y$ είναι οι εξής:

- $y = \langle x, w\rangle$ και $a = \langle x, w_1\rangle$, όπου $w, w_1 \in W(x)$ και $w \preceq w_1$. Τότε για την $b = \langle x, w\rangle$ ισχύει $f(a, b) = y$ και το μήκος της b είναι το πολύ πολυωνυμικό ως προς $|y| + |a|$, εφόσον χρησιμοποιούμε λογική συνάρτηση σύζευξης.

- $y = \langle x, x\rangle$ και $a = \langle x, x\rangle$. Τότε υπάρχει κάποιος μάρτυρας w ώστε για την $b = \langle x, w\rangle$ να ισχύει $f(a, b) = y$. Το μήκος της b είναι το πολύ πολυωνυμικό ως προς $|x| + |w|$, άρα και ως προς $|y| + |a|$.

- $y = \langle x, x\rangle$ και $a = \langle x, w\rangle$, όπου $w \in W(x)$. Τότε για την $b = \langle x, x\rangle$ ισχύει $f(a, b) = y$ και προφανώς η b έχει μήκος το πολύ πολυωνυμικό ως προς $|y| + |a|$.

- $y = \langle t, t1 \rangle$ και $a = \langle x, w \rangle$, όπου $|w| = p(|x|)$. Τότε για $b = \langle x', x' \rangle$, όπου $|x'| = |x|$ και $x' \neq x$, ισχύει $f(a, b) = y$ και η b έχει μήκος το πολύ πολυωνυμικό ως προς $|y| + |a|$.

- $y = \langle t, t1 \rangle$ και $a = \langle x, z \rangle$, όπου $|z| \neq p(|x|)$. Τότε για $b = \langle x, z \rangle$ ισχύει $f(a, b) = y$ και προφανώς η b έχει μήκος το πολύ πολυωνυμικό ως προς $|y| + |a|$.

Συνδυάζοντας τα πολυώνυμα που φράσσουν άνω το μήκος του δεύτερου ορίσματος της f ως προς τα μήκη του πρώτου ορίσματος και του αποτελέσματος, προκύπτει ότι ισχύει η πρώτη συνθήκη του ορισμού 3.3.4. Με παρόμοιο τρόπο αποδεικνύεται ότι ισχύει και η δεύτερη συνθήκη του ορισμού αυτού, επομένως η f είναι s-τίμια. Για να δείξουμε ότι η f είναι ισχυρά μη αντιστρέψιμη απομένει να δείξουμε ότι δεν ισχύουν οι δύο συνθήκες του ορισμού 3.3.5. Προς άτοπο, έστω ότι ισχύει η πρώτη από αυτές. Δηλαδή υπάρχει μία συνάρτηση $g \in \mathbf{FP}$ ώστε για κάθε $y \in \mathrm{range}(f)$ και για κάθε προ-εικόνα (x_1, x_2) του y να ισχύει $(y, x_1) \in \mathrm{domain}(g)$ και $f(x_1, g(y, x_1)) = y$. Συγκεκριμένα, εστιάζοντας στον δεύτερο κλάδο του ορισμού της f, αν $y = \langle x, x \rangle$ και $x \in L(N)$, η τιμή της $g(y, \langle x, x \rangle)$ θα είναι της μορφής $\langle x, w \rangle$, όπου $w \in W(x)$. Χρησιμοποιώντας τη συνάρτηση $g \in \mathbf{FP}$, ο αλγόριθμος 15 αποφασίζει το σύνολο $L(N)$ σε πολυωνυμικό χρόνο, εφόσον ο έλεγχος αν μια συμβολοακολουθία w είναι μάρτυρας μιας συμβολοακολουθίας x μπορεί να γίνει σε πολυωνυμικό χρόνο.Επομένως καταλήγουμε σε άτοπο γιατί έχουμε υποθέσει ότι $L(N) \in \mathbf{NP} \setminus \mathbf{P}$. Με παρόμοιο τρόπο αποδεικνύεται ότι και η δεύτερη συνθήκη του ορισμού 3.3.5 δεν ισχύει. Άρα η f είναι ισχυρά μη αντιστρέψιμη.

Η f δεν είναι αντιστρέψιμη σε πολυωνυμικό χρόνο:
Το γεγονός ότι η f είναι ισχυρά μη αντιστρέψιμη δεν συνεπάγεται ότι δεν είναι αντιστρέψιμη σε πολυωνυμικό χρόνο. Μάλιστα, είναι γνωστό ότι αν $\mathbf{P} \neq \mathbf{NP}$ τότε υπάρχουν τίμιες και ισχυρά μη αντιστρέψιμες συναρτήσεις που είναι υπολογιστές σε πολυωνυμικό χρόνο και ταυτόχρονα είναι αντιστρέψιμες σε πολυωνυμικό χρόνο [HPR01]. Έστω ότι η f είναι αντιστρέψιμη σε

Αλγόριθμος 15 Ντετερμινιστικός αλγόριθμος πολυωνυμικού χρόνου που αποφασίζει μία γλώσσα $L(N) \in \mathbf{NP} \setminus \mathbf{P}$, υπό την προϋπόθεση ότι η συνάρτηση f του θεωρήματος 3.3.9 δεν είναι ισχυρά μη αντιστρέψιμη

Είσοδος: x

1: $z \leftarrow g(\langle x, x\rangle, \langle x, x\rangle)$
2: ερμήνευσε την z ως κωδικοποίηση ενός ζεύγους $\langle u, w\rangle$
3: **if** $u = x$ και $w \in W(x)$ **then**
4: τερμάτισε την εκτέλεση σε κατάσταση αποδοχής
5: **else**
6: τερμάτισε την εκτέλεση σε κατάσταση απόρριψης
7: **end if**

πολυωνυμικό χρόνο, δηλαδή υπάρχει $g \in \mathbf{FP}$ ώστε για κάθε $y \in \mathrm{range}(f)$ να ισχύουν: $y \in \mathrm{domain}(g)$, $(\mathrm{first}(g(y)), \mathrm{second}(g(y))) \in \mathrm{domain}(f)$ και $f(\mathrm{first}(g(y)), \mathrm{second}(g(y))) = y$. Αν $x \in L(N)$, τότε $\langle x, x\rangle \in \mathrm{range}(f)$, άρα ισχύει $f(\mathrm{first}(g(\langle x, x\rangle)), \mathrm{second}(g(\langle x, x\rangle))) = \langle x, x\rangle$. Από τον ορισμό της f προκύπτει τότε ότι είτε το $\mathrm{first}(g(\langle x, x\rangle))$, είτε το $\mathrm{second}(g(\langle x, x\rangle))$ θα είναι ένα ζεύγος της μορφής $\langle x, w\rangle$, όπου $w \in W(x)$. Επομένως μπορούμε να αποφασίσουμε τη γλώσσα $L(N)$ σε ντετερμινιστικό πολυωνυμικό χρόνο, χρησιμοποιώντας έναν αλγόριθμο παρόμοιο με τον αλγόριθμο 15. Καταλήγουμε σε άτοπο, άρα η f δεν είναι αντιστρέψιμη σε πολυωνυμικό χρόνο.

Η f είναι τίμια:
Οι περιπτώσεις των δύο πρώτων κλάδων της f καλύπτονται από την ανάλυση περιπτώσεων που έγινε για την απόδειξη της s-τιμιότητας της f. Για τον τρίτο κλάδο, αρκεί να επιλέξουμε το πολυώνυμο q του ορισμού 3.3.1 ώστε ταυτόχρονα να καλύπτει τις δύο πρώτες περιπτώσεις, αλλά και να ισχύει $|z_1| + |z_2| \leq q(|\langle t, t1\rangle|)$, όπου z_1, z_2 είναι οι συμβολοακολουθίες με το μικρότερο άθροισμα μηκών για τις οποίες ισχύει $f(z_1, z_2) = \langle t, t1\rangle$.

Η f είναι προσεταιριστική:
Θα δείξουμε ότι για κάθε $z, z', z'' \in \Sigma^\star$ ισχύει:

$$f(z, f(z', z'')) = f(f(z, z'), z'')$$

Ως συντόμευση, θα λέμε ότι μία συμβολοακολουθία a είναι *νόμιμη* αν είναι της μορφής $a = \langle x, w\rangle$, όπου $w \in W(x)$. Από τον ορισμό της f προκύπτει ότι αν δύο ή παραπάνω από τις z, z', z'' δεν είναι νόμιμες τότε ισχύει $f(z, f(z', z'')) = f(f(z, z'), z'') = \langle t, t1\rangle$. Επίσης, αν οι z, z', z'', ως συζεύξεις, δεν έχουν όλες το ίδιο πρώτο στοιχείο τότε και πάλι $f(z, f(z', z'')) = f(f(z, z'), z'') = \langle t, t1\rangle$. Η σχέση αυτή εξακολουθεί να ισχύει και στην περίπτωση $\text{first}(z) = \text{first}(z') = \text{first}(z'')$ όπου ακριβώς μία από τις z, z', z'' δεν είναι νόμιμη, εκτός αν αυτή που δεν είναι νόμιμη είναι η $\langle \text{first}(z), \text{first}(z)\rangle$. Απομένουν οι εξής δύο περιπτώσεις:

- Ακριβώς μία από τις z, z', z'' δεν είναι νόμιμη, $\text{first}(z) = \text{first}(z') = \text{first}(z'')$ και αυτή που δεν είναι νόμιμη έχει τη μορφή $\langle \text{first}(z), \text{first}(z)\rangle$. Έστω ότι $z = \langle x, x\rangle$, $z' = \langle x, w'\rangle$, $z'' = \langle x, w''\rangle$. Εκ του ορισμού της f προκύπτουν:

$$f(z, f(z', z'')) = f(z, \langle x, \min\{w', w''\}\rangle) = \langle x, x\rangle$$

$$f(f(z, z'), z'') = f(\langle x, x\rangle, z'') = \langle x, x\rangle$$

  Άρα $f(z, f(z', z'')) = f(f(z, z'), z'')$.

- Οι z, z', z'' είναι όλες νόμιμες και έχουν όλες το ίδιο πρώτο στοιχείο. Έστω $z = \langle x, w\rangle$, $z' = \langle x, w'\rangle$, $z'' = \langle x, w''\rangle$. Εκ του ορισμού της f προκύπτουν:

$$f(z, f(z', z'')) = f(z, \langle x, \min\{w', w''\}\rangle) = \langle x, \min\{w, w', w''\}\rangle$$

$$f(f(z, z'), z'') = f(\langle x, \min\{w, w'\}\rangle, z'') = \langle x, \min\{w, w', w''\}\rangle$$

■

Τα θεωρήματα 3.1.4, 3.2.10 και 3.3.9 κινούνται σε έναν κοινό άξονα. Η διατύπωση του καθενός ισχυρίζεται την ισοδυναμία μεταξύ της ανισότητας μίας κλάσης πολυπλοκότητας με την κλάση **P** και της ύπαρξης ενός συγκεκριμένου τύπου συναρτήσεων μονής κατεύθυνσης. Έχουμε ήδη αναφέρει ότι για τις αποδείξεις των τριών αυτών θεωρημάτων χρησιμοποιείται η κοινή τεχνική των "συναρτήσεων μονής κατεύθυνσης", την οποία θα οριοθετήσουμε λίγο ακριβέστερα.

Γενικά, η τεχνική βασίζεται στο ότι η υποτιθέμενη δυνατότητα πολυωνυμικής αντιστροφής συναρτήσεων μονής κατεύθυνσης παρέχει έναν αποδοτικό τρόπο υπολογισμού μαρτύρων για την αποδοχή συμβολοακολουθιών από μηχανές Turing συγκεκριμένου τύπου. Οι αποδείξεις των αποτελεσμάτων ακολουθούν τον εξής σκελετό:

- Υποθέτουμε ότι ισχύει ένας γνήσιος εγκλεισμός της μορφής $\mathbf{P} \subset \mathcal{C}$, όπου η $\mathcal{C}$ ορίζεται με περιορισμό χρόνου. Θεωρούμε μία γλώσσα $A \in \mathcal{C} \setminus \mathbf{P}$.

- Ορίζουμε μία συνάρτηση f που να είναι υπολογιστή σε πολυωνυμικό χρόνο, τίμια και να έχει όποιες άλλες ιδιότητες χρειαζόμαστε. Ο ορισμός της f γίνεται έτσι ώστε να απεικονίζει, σε γενικές γραμμές, μάρτυρες w για το $x \in A$ στην ίδια τη συμβολοακολουθία x.

- Προς άτοπο, υποθέτουμε ότι η f είναι αντιστρέψιμη σε πολυωνυμικό χρόνο. Η αντίστροφή της, g, με είσοδο $x \in A$ υπολογίζει σε πολυωνυμικό χρόνο ένα μάρτυρα για το x. Έτσι, ελέγχοντας σε πολυωνυμικό χρόνο αν η τιμή που επιστρέφει η g είναι όντως μάρτυρας για την είσοδο, κατασκευάζουμε έναν πολυωνυμικό αλγόριθμο που αποφασίζει το A.

- Καταλήγουμε σε άτοπο, άρα η f προκύπτει μη αντιστρέψιμη σε πολυωνυμικό χρόνο. Εφόσον έχουμε ήδη αποδείξει ότι η f είναι υπολογιστή σε πολυωνυμικό χρόνο και τίμια, η f είναι τελικά συνάρτηση μονής κατεύθυνσης.

Οι αποδείξεις των θεωρημάτων 3.1.4 και 3.2.10 ακολουθούν κατά γράμμα το σκελετό αυτό. Λαμβάνεται μέριμνα ώστε η συνάρτηση f που κατασκευάζεται στην απόδειξη του θεωρήματος 3.2.10 να είναι ένα-προς-ένα, το οποίο επιτυγχάνεται με έναν πολύ απλό χειρισμό. Στο θεώρημα 3.3.9 η κατάσταση είναι λίγο πιο πολύπλοκη, λόγω των επιπλέον ιδιοτήτων που πρέπει να έχει η συνάρτηση f. Όμως ο παραπάνω σκελετός αναγνωρίζεται και εδώ εύκολα, με τη διαφορά ότι χρειάζεται να δώσουμε δύο αποδείξεις μη αντιστρεψιμότητας: την πρώτη για την ισχυρή μη αντιστρεψιμότητα και τη δεύτερη για τη μη πολυωνυμική αντιστρεψιμότητα.

Ένα άλλο σημείο που μπορεί να τονιστεί στις αποδείξεις των θεωρημάτων 3.1.4 και 3.2.10 είναι η απόδειξη της ύπαρξης ενός συνόλου prefix-inv$(f) \in \mathcal{C} \setminus \mathbf{P}$, δεδομένης μίας συνάρτησης μονής κατεύθυνσης f. Υποθέτοντας ότι prefix-inv$(f) \in \mathbf{P}$, καταλήγουμε σε άτοπο γιατί κατασκευάζουμε πολυωνυμικό αλγόριθμο που αντιστρέφει την f. Η τεχνική που χρησιμοποιείται για το σκοπό αυτό ονομάζεται "αναζήτηση προθέματος" και περιγράφεται καθαρά από τον αλγόριθμο 13. Από το άτοπο στο οποίο καταλήγουμε προκύπτει τελικά ότι $\mathbf{P} \neq \mathcal{C}$.

Κεφάλαιο 4

Τουρνουά και ημιαποδοτικοί υπολογισμοί

Για λόγους που έχουμε περιγράψει ήδη από το εισαγωγικό κεφάλαιο, θεωρούμε ως αποδοτικούς τους υπολογισμούς που μπορούν να γίνουν σε πολυωνυμικό χρόνο από κάποια ντετερμινιστική μηχανή Turing. Επομένως, έχουμε αποδοτικό αλγόριθμο απόφασης για ένα σύνολο αν η χαρακτηριστική του συνάρτηση υπολογίζεται σε πολυωνυμικό χρόνο. Σε αυτό το κεφάλαιο ασχολούμαστε με ημιαποδοτικώς υπολογιστά σύνολα, τα όποια ενδεχομένως να μην έχουν αποδεδειγμένα πολυωνυμικού χρόνου χαρακτηριστική συνάρτηση, αλλά η συνάρτηση επιλογής τους υπολογίζεται σε πολυωνυμικό χρόνο. Όπως θα δούμε, η συνάρτηση επιλογής είναι μία γενίκευση της χαρακτηριστικής συνάρτησης.

Από την άποψη των αποδεικτικών τεχνικών, το ενδιαφέρον επικεντρώνεται στη συσχέτιση που υπάρχει ανάμεσα σε δύο φαινομενικά άσχετες έννοιες. Από τη μία πλευρά, έχουμε δύο θεωρήματα από τη θεωρία γραφημάτων που εξασφαλίζουν την ύπαρξη "μικρών" υποσυνόλων του συνόλου των κόμβων ενός γράφου τουρνουά, τα οποία έχουν συγκεκριμένες ιδιότητες. Από την άλλη πλευρά, εμφανίζεται η αναζήτηση για άνω φράγματα στη μη ομοιόμορφη πολυπλοκότητα των ημιαποδοτικώς υπολογιστών συνόλων. Προκύπτει ότι τα αποτελέσματα για τους γράφους τουρνουά εφαρμόζονται άμεσα και κομψά για να δώσουν απαντήσεις σε αυτήν την αναζήτηση.

Θα δούμε ακόμη πώς σχετίζονται τα ημιαποδοτικώς υπολογιστά σύνολα με υπολογισμούς με συμβουλές, και θα εξετάσουμε κατά πόσο είναι δυνατόν να ισχυροποιήσουμε το διερμηνέα συμβουλών ώστε να απαιτεί συμβουλές με μικρότερο μήκος. Τέλος, μέσω του μη ντετερμινιστικού ανάλογου των ημιαποδοτικώς υπολογιστών συνόλων θα δείξουμε ότι η δυνατότητα υπολογισμού μοναδικών λύσεων σε **NP** προβλήματα συνεπάγεται την κατάρρευση της πολυωνυμικής ιεραρχίας.

4.1 Εισαγωγικά

4.1.1 Υπολογισμοί με συμβουλές

Τροποποιούμε για μία ακόμη φορά το υπολογιστικό μοντέλο της μηχανής Turing, προσθέτοντας μία ταινία εισόδου που προορίζεται μόνο για ανάγνωση και λέγεται ταινία συμβουλών. Υποθέτουμε ότι μαζί με κάθε συμβολοακολουθία x που παρέχεται ως είσοδος στην αρχική ταινία εισόδου, εμφανίζεται στην ταινία συμβουλών μία συμβολοακολουθία $A(|x|)$, όπου A μία συνάρτηση που απεικονίζει ακέραιους σε συμβολοακολουθίες. Λέμε ότι η μηχανή αποφασίζει τη γλώσσα L, αν για κάθε $x \in L$ αποδέχεται την είσοδο $(x, A(|x|)$ και για κάθε $x \notin L$ απορρίπτει την είσοδο $(x, A(|x|)$. Παρατηρούμε ότι η συμβουλή που λαμβάνει η μηχανή είναι η ίδια για όλες τις συμβολοακολουθίες που έχουν το ίδιο μήκος, αφού η τιμή της συνάρτησης A εξαρτάται μόνο από το μήκος της εισόδου x. Από την άποψη της πολυπλοκότητας, μας ενδιαφέρουν δύο παράμετροι της λειτουργίας της μηχανής. Αφενός, το υπολογιστικό μοντέλο που θεωρούμε ότι ακολουθεί η μηχανή Turing (π.χ. ντετερμινιστικό, μη ντετερμινιστικό, πιθανοτικό) και οι περιορισμοί χρόνου ή χώρου που τίθενται στο μοντέλο αυτό, και αφετέρου το μήκος που έχει η συμβουλή που δέχεται η μηχανή ως είσοδο.

Ακολουθούν οι ορισμοί των εννοιών, που οφείλονται στους Karp και Lipton [KL80].

Ορισμός 4.1.1 *Για κάθε σύνολο $A \subseteq \Sigma^\star$ και κάθε συνάρτηση $f : \mathbb{N} \to \mathbb{N}$, ο συμβολισμός A/f υποδηλώνει την κλάση όλων των γλωσσών L, για κάθε μία από τις οποίες υπάρχει μία συνάρτηση $h : \mathbb{N} \to \Sigma^\star$ που ικανοποιεί τα εξής:*

1. *Για κάθε φυσικό n, $|h(n)| = f(n)$.*
2. *$L = \{x : \langle x, h(|x|)\rangle \in A\}$.*

Η συνάρτηση h λέγεται συνάρτηση συμβουλών *και το σύνολο A λέγεται* ερμηνευτής συμβουλών.

Ορισμός 4.1.2 *Για κάθε κλάση $\mathcal{C}$ και κάθε συνάρτηση $f : \mathbb{N} \to \mathbb{N}$, ορίζουμε το συμβολισμό $\mathcal{C}/f$:*

$$\mathcal{C}/f = \bigcup_{A \in \mathcal{C}} A/f$$

Ορισμός 4.1.3 *Για κάθε σύνολο $A \subseteq \Sigma^\star$ και κάθε κλάση συναρτήσεων $\mathcal{F}$, ορίζουμε το συμβολισμό $A/\mathcal{F}$:*

$$A/\mathcal{F} = \bigcup_{f \in \mathcal{F}} A/f$$

Ορισμός 4.1.4 *Για κάθε κλάση $\mathcal{C}$ και κάθε κλάση συναρτήσεων $\mathcal{F}$, ορίζουμε το συμβολισμό $\mathcal{C}/\mathcal{F}$:*

$$\mathcal{C}/\mathcal{F} = \bigcup_{A \in \mathcal{C}} \bigcup_{f \in \mathcal{F}} A/f$$

Τα μήκη των συμβουλών που θα μας απασχολήσουν στη συνέχεια είναι γραμμικά, τετραγωνικά, ή γενικά πολυωνυμικά. Άρα μας ενδιαφέρουν οι ακόλουθες κλάσεις συναρτήσεων:

Ορισμός 4.1.5

1. $\text{poly} = \bigcup_{k \in \mathbb{N}} \{f : f(n) = O(n^k)\}$.
2. $\text{linear} = \{f : f(n) = O(n)\}$.
3. $\text{quadratic} = \{f : f(n) = O(n^2)\}$.

Προφανώς linear $\subset$ quadratic $\subset$ poly.

Για παράδειγμα, η κλάση **P**/poly περιλαμβάνει τις γλώσσες που αποφασίζονται από ντετερμινιστική μηχανή Turing σε πολυωνυμικό χρόνο, με το πολύ πολυωνυμικό μήκος συμβουλής. Η συγκεκριμένη κλάση φαίνεται να είναι αρκετά "κοντά" στην **P**, με την έννοια ότι οι γλώσσες που περιλαμβάνει αποφασίζονται αποδοτικά με μικρό μήκος συμβουλής. Οι ορισμοί όμως δεν παρέχουν κάποιο άνω φράγμα για την πολυπλοκότητα υπολογισμού της συνάρτησης συμβουλών.

4.1.2 Ημιαποδοτικοί υπολογισμοί

Γνωρίζουμε ότι με κάθε σύνολο $A \subseteq \Omega$, όπου Ω είναι το θεωρούμενο σύμπαν, συνδέεται μία συνάρτηση $\chi_A : \Omega \to \{0,1\}$ για την οποία ισχύει, για κάθε $x \in \Omega$:

$$x \in A \Leftrightarrow \chi_A(x) = 1$$

Η συνάρτηση χ_A είναι η χαρακτηριστική συνάρτηση του συνόλου A. Την κλάση των συνόλων που έχουν χαρακτηριστική συνάρτηση που υπολογίζεται ντετερμινιστικά σε χρόνο πολυωνυμικό ως προς το μήκος της αναπαράστασης της εισόδου, την έχουμε ονομάσει **P**. Όπως έχουμε δει, το ερώτημα της ύπαρξης αποδοτικά υπολογιστής χαρακτηριστικής συνάρτησης είναι ακόμη ανοιχτό για πολλά και σημαντικά σύνολα όπως το **SAT**. Για το λόγο αυτό, αναζητούμε γενικεύσεις του **P** με σκοπό να διερευνήσουμε κατά πόσο αυτές οι γενικεύσεις είναι δυνατό να περιέχουν τα σύνολα που μας ενδιαφέρουν.

Ο Jockusch όρισε το 1968 την έννοια της συνάρτησης επιλογής, καθώς μελετούσε την κλάση των ημιαναδρομικών συνόλων, δηλαδή των συνόλων που έχουν αναδρομική συνάρτηση επιλογής. Αργότερα, ο Selman θεώρησε την κλάση **P**-sel των ημιαποδοτικών συνόλων, δηλαδή των συνόλων που έχουν συνάρτηση επιλογής υπολογιστή σε πολυωνυμικό χρόνο.

Ορισμός 4.1.6 *[Joc68] Για κάθε σύνολο $A \subseteq \Omega$, όπου Ω είναι το θεωρούμενο σύμπαν,* συνάρτηση επιλογής $\mathrm{sel}_A : \Omega \times \Omega \to \Omega$ *ονομάζεται μία συνάρτηση με τις εξής ιδιότητες:*

1. $(\forall x \in \Omega)(\forall y \in \Omega)[\mathrm{sel}_A(x,y) \in \{x,y\}]$.

2. $(\forall x \in \Omega)(\forall y \in \Omega)[\{x,y\} \cap A \neq \emptyset \Rightarrow \mathrm{sel}_A(x,y) \in A]$.

Δηλαδή η τιμή της συνάρτησης επιλογής ενός συνόλου A είναι πάντα ίση με ένα από τα δύο ορίσματά της. Αν τουλάχιστον ένα από τα ορίσματα ανήκει στο A, τότε η sel_A "επιλέγει" αυτό που ανήκει στο A. Αν κανένα από τα ορίσματα δεν ανήκει στο A, η sel_A έχει αυθαίρετη τιμή, αλλά πάντα ίση με ένα από τα ορίσματά της. Παρατηρούμε ότι ενώ η χαρακτηριστική συνάρτηση είναι μοναδική για κάθε σύνολο, η συνάρτηση επιλογής αντίθετα μπορεί να οριστεί με πολλούς τρόπους. Κατά συνέπεια κάθε σύνολο έχει περισσότερες από μία συναρτήσεις επιλογής, για την ακρίβεια το πλήθος τους είναι $2^{|A|^2-|A|+(|\Omega|-|A|)^2-(|\Omega|-|A|)}$.

Ορισμός 4.1.7 *[Sel79, Sel82a, Sel82b] Η κλάση* $\mathbf{P}$-sel *των* ημιαποδοτικώς υπολογιστών *συνόλων ορίζεται ως η κλάση των συνόλων που έχουν συνάρτηση επιλογής υπολογιστή σε πολυωνυμικό χρόνο.*

Αν ένα σύνολο έχει πολυωνυμικά υπολογιστή συνάρτηση επιλογής μπορούμε χωρίς βλάβη της γενικότητας να θεωρούμε ότι αυτή είναι αντιμεταθετική, λόγω του επόμενου θεωρήματος.

Θεώρημα 4.1.8 *Αν $L \in \mathbf{P}$-sel, τότε η L έχει αντιμεταθετική συνάρτηση επιλογής υπολογιστή σε πολυωνυμικό χρόνο.*

Απόδειξη

Έστω $f : \Sigma^\star \times \Sigma^\star \to \Sigma^\star$ μία πολυωνυμικά υπολογιστή συνάρτηση επιλογής της L. Θεωρούμε τη συνάρτηση $f' : \Sigma^\star \times \Sigma^\star \to \Sigma^\star$ που ορίζεται ως εξής:

$$f'(x,y) = f(\min\{x,y\}, \max\{x,y\})$$

Η f' είναι υπολογιστή σε πολυωνυμικό χρόνο, είναι αντιμεταθετική, και για κάθε $x, y \in \Sigma^\star$ ισχύει $f'(x,y) \in \{x,y\}$. Αν $\{x,y\} \cap L \neq \emptyset$, τότε προφανώς $\{\min\{x,y\}, \max\{x,y\}\} \cap L \neq \emptyset$, άρα $f(\min\{x,y\}, \max\{x,y\}) \in L$, άρα και $f'(x,y) \in L$.

■

4.1.3 Τουρνουά

Ορισμός 4.1.9 *Ένας κατευθυνόμενος γράφος $G(V, E)$ λέγεται* προσανατολισμένος *αν για κάθε $a, b \in V$ τουλάχιστον μία από τις ακμές (a, b), (b, a) δεν ανήκει στο E.*

Ορισμός 4.1.10 *Κάθε πλήρης προσανατολισμένος γράφος ονομάζεται* τουρνουά. *Επιπλέον, κάθε τουρνουά με k κόμβους ονομάζεται* k-τουρνουά.

Η ονομασία "τουρνουά" προέρχεται από το γεγονός ότι ένας γράφος k-τουρνουά μπορεί να αναπαραστήσει ένα διαγωνισμό k παικτών, όπου ο κάθε παίκτης αγωνίζεται από μία φορά εναντίον όλων των υπολοίπων, και δεν επιτρέπονται ισοπαλίες. Για κάθε ζεύγος παικτών, η ακμή που συνδέει τους αντίστοιχους κόμβους στο γράφο έχει κατεύθυνση από το χαμένο προς το νικητή. Τα τουρνουά έχουν μελετηθεί αρκετά, αλλά για τους σκοπούς μας χρειάζονται μόνο δύο βασικά αποτελέσματα. Πρόκειται για τα θεωρήματα 4.1.13 και 4.1.14, που θα δούμε στη συνέχεια.

Ορισμός 4.1.11 *Για κάθε κατευθυνόμενο γράφο $G(V, E)$, για κάθε φυσικό αριθμό i και για κάθε $v \in V$, ορίζουμε το σύνολο $R_{i,G}(v)$:*

$$R_{i,G}(v) = \left\{ u \in V : \begin{array}{l} \textit{υπάρχει κατευθυνόμενο μονοπάτι από} \\ \textit{το } v \textit{ στο } u\textit{, με μήκος το πολύ } i \end{array} \right\}$$

Θεώρημα 4.1.12 *Σε κάθε k-τουρνουά $G(V, E)$, $k \geq 1$, υπάρχει ένας κόμβος $u \in V$ με $|R_{1,G}(u)| \geq \left\lceil \frac{k+1}{2} \right\rceil$.*

Απόδειξη

Θεωρούμε για κάθε $v \in V$, το σύνολο $W_{1,G}(v)$ των κόμβων από κάθε έναν από τους οποίους υπάρχει κατευθυνόμενο μονοπάτι μήκους το πολύ 1 προς τον κόμβο v:

$$W_{1,G}(v) = \{w \in V : v \in R_{1,G}(w)\}$$

Παρατηρούμε ότι $\sum_{v \in V} |R_{1,G}(v)| = \sum_{v \in V} |W_{1,G}(v)| = |V| + |E|$, και για κάθε $u \in V$ ισχύει $|R_{1,G}(u)| + |W_{1,G}(u)| = k + 1$. Υπάρχει λοιπόν ένας κόμβος $u \in V$ με $|R_{1,G}(u)| \geq |W_{1,G}(u)|$, γιατί σε αντίθετη περίπτωση για κάθε κόμβο

$v \in V$ θα ίσχυε $|R_{1,G}(u)| < |W_{1,G}(u)|$ και αθροίζοντας σε όλους τους κόμβους θα είχαμε $\sum_{v\in V} |R_{1,G}(v)| < \sum_{v\in V} |W_{1,G}(v)|$.

Για τον κόμβο u λοιπόν ισχύει:

$$\begin{aligned} |R_{1,G}(u)| \geq |W_{1,G}(u)| &\Rightarrow 2|R_{1,G}(u)| \geq |W_{1,G}(u)| + |R_{1,G}(u)| \\ &\Rightarrow 2|R_{1,G}(u)| \geq k+1 \\ &\Rightarrow |R_{1,G}(u)| \geq \left\lceil \frac{k+1}{2} \right\rceil \end{aligned}$$

■

Θεώρημα 4.1.13 *Αν ο γράφος $G(V,E)$ είναι k-τουρνουά, $k \geq 1$, τότε υπάρχει ένα σύνολο $H \subseteq V$ με τις εξής ιδιότητες:*

1. *$|H| \leq \lfloor \log(k+1) \rfloor$, και*

2. *για κάθε $v \in V$ υπάρχει $g \in H$ ώστε $v \in R_{1,G}(g)$.*

Απόδειξη

Η απόδειξη θα γίνει με επαγωγή στο k. Αν ο $G(V,E)$ είναι 1-τουρνουά, τότε το σύνολο $H = V$ έχει μέγεθος $|H| = 1 = \log 2$. Επιπλέον, για το μοναδικό στοιχείο $v \in V = H$ ισχύει $v \in R_{1,G}(v)$. Άρα το θεώρημα ισχύει για $k = 1$.

Έστω ότι ισχύει για όλα τα $k \leq m$, όπου $m \geq 1$ σταθερά. Δηλαδή για κάθε k-τουρνουά $G(V,E)$ με $k \leq m$, υπάρχει $H \subseteq V$ με μέγεθος $|H| \leq \lfloor \log(k+1) \rfloor$, τέτοιο ώστε για κάθε $v \in V$ να υπάρχει $g \in H$ που ικανοποιεί τη σχέση $v \in R_{1,G}(g)$.

Θα δείξουμε ότι ισχύει για $k = m+1$. Έστω $G(V,E)$ ένα $(m+1)$-τουρνουά. Λόγω του θεωρήματος 4.1.12, υπάρχει ένας κόμβος $u_0 \in V$ με $|R_{1,G}(u_0)| \geq \left\lceil \frac{m+2}{2} \right\rceil$.

Θεωρούμε το γράφο $G'(V',E') = G - R_{1,G}(u_0)$ που προκύπτει αν από τον G αφαιρέσουμε όλους τους κόμβους του συνόλου $R_{1,G}(u_0)$ και τις προσκείμενες ακμές τους. Ο G' έχει πλήθος κόμβων $|V'| = |V| - |R_{1,G}(u_0)| \leq m + 1 - \left\lceil \frac{m+2}{2} \right\rceil = \left\lfloor \frac{m}{2} \right\rfloor$ και είναι προφανώς ένα $\left\lfloor \frac{m}{2} \right\rfloor$-τουρνουά. Λόγω της επαγωγικής υπόθεσης υπάρχει ένα υποσύνολο $H' \subseteq V'$, με $|H'| \leq \left\lfloor \log\left(\left\lfloor \frac{m}{2} \right\rfloor + 1\right) \right\rfloor$, τέτοιο ώστε για κάθε $v \in V'$ να υπάρχει $g \in H'$ ώστε $v \in R_{1,G'}(g)$.

Το σύνολο $H = H' \cup \{u_0\}$ είναι προφανώς υποσύνολο του V και για κάθε $v \in V$ έχουμε:

- Αν $v \in V'$, υπάρχει $g \in H'$ ώστε $v \in R_{1,G'}(g)$. Άρα, επειδή $H' \subseteq H$, υπάρχει $g \in H$ ώστε $v \in R_{1,G}(g)$.

- Αν $v \notin V'$, τότε $v \in R_{1,G}(u_0)$. Άρα υπάρχει $g \in H$ και συγκεκριμένα $g = u_0$, ώστε $v \in R_{1,G}(g)$.

Για να ολοκληρώσουμε την απόδειξη αρκεί να δείξουμε ότι για το μέγεθος του H ισχύει $|H| \leq \lfloor \log(m+2) \rfloor$. Εφόσον $H = H' \cup \{u_0\}$, το μέγεθος του H είναι $|H| = 1 + |H'|$. Λόγω της επαγωγικής υπόθεσης $|H'| \leq \left\lfloor \log\left(\left\lfloor \frac{m}{2} \right\rfloor + 1\right) \right\rfloor$. Άρα $|H| \leq 1 + \left\lfloor \log\left(\left\lfloor \frac{m}{2} \right\rfloor + 1\right) \right\rfloor$.

Θεωρούμε τη μεγαλύτερη δύναμη του 2, 2^{λ_0}, που δεν υπερβαίνει την τιμή $\left\lfloor \frac{m}{2} \right\rfloor + 1$. Προφανώς τότε $\left\lfloor \log\left(\left\lfloor \frac{m}{2} \right\rfloor + 1\right) \right\rfloor = \lambda_0$. Επιπλέον έχουμε:

$$
\begin{aligned}
2^{\lambda_0} \leq \left\lfloor \frac{m}{2} \right\rfloor + 1 < 2^{\lambda_0+1} \quad &\Rightarrow \quad 2^{\lambda_0} - 1 \leq \left\lfloor \frac{m}{2} \right\rfloor < 2^{\lambda_0+1} - 1 \\
&\Rightarrow \quad 2^{\lambda_0+1} - 2 \leq m < 2^{\lambda_0+2} - 2 \\
&\Rightarrow \quad 2^{\lambda_0+1} \leq m + 2 < 2^{\lambda_0+2} \\
&\Rightarrow \quad \lfloor \log(m+2) \rfloor = \lambda_0 + 1
\end{aligned}
$$

Άρα $|H| \leq \lfloor \log(m+2) \rfloor$.

■

Θεώρημα 4.1.14 *Αν ο γράφος $G(V,E)$ είναι k-τουρνουά, $k \geq 1$, τότε υπάρχει ένας κόμβος $v \in V$ τέτοιος ώστε $V = R_{2,G}(v)$.*

Απόδειξη

Η απόδειξη θα γίνει με επαγωγή στο k. Το θεώρημα ισχύει προφανώς για $k = 1$ και $k = 2$. Έστω ότι ισχύει για $k = m \geq 2$. Θα δείξουμε ότι ισχύει για $k = m + 1$.

Έστω $G(V, E)$ ένα $(m+1)$-τουρνουά και $a \in V$ ένας κόμβος του. Θεωρούμε τον γράφο $G'(V', E') = G - \{a\}$. Ο G' είναι m-τουρνουά, άρα λόγω της επαγωγικής υπόθεσης υπάρχει ένας κόμβος $b \in V'$ τέτοιος ώστε $V \setminus \{a\} = V' = R_{2,G'}(b)$. Αν $(b, a) \in E$ ή $a \in R_{2,G}(b)$ τότε το θεώρημα ισχύει, αφού $V = R_{2,G}(b)$.

Στο εξής λοιπόν υποθέτουμε ότι $(a, b) \in E$ και $a \notin R_{2,G}(b)$. Έπεται ότι

$$R_{1,G}(b) \subseteq R_{1,G}(a)$$

γιατί αν $v \in R_{1,G}(b)$ και $v \notin R_{1,G}(a)$, επειδή $v \notin R_{1,G}(a)$ έχουμε $(v, a) \in E$ και επειδή $v \in R_{1,G}(b)$ έχουμε $a \in R_{2,G}(b)$. Επίσης, για κάθε $c \in R_{1,G}(b)$ ισχύει

$$(a, c) \in E$$

γιατί αν για κάποιο $c \in R_{1,G}(b)$ είχαμε $(c, a) \in E$, θα προέκυπτε $a \in R_{2,G}(b)$.

Αρκεί να δείξουμε ότι $V \subseteq R_{2,G}(a)$. Έστω $v \in V$. Αν $v = a$ ή $v = b$, τότε προφανώς $v \in R_{2,G}(a)$, αφού $(a, b) \in E$. Αν $v \neq a$, $v \neq b$ και $v \in R_{1,G}(b)$, τότε $v \in R_{1,G}(a) \subseteq R_{2,G}(a)$. Τέλος, έστω ότι $v \neq a$, $v \neq b$ και $v \notin R_{1,G}(b)$. Επειδή $V \setminus \{a\} = R_{2,G'}(b)$ και $v \neq a$, έχουμε $v \in R_{2,G'}(b)$. Άρα υπάρχει ένας κόμβος $v' \neq a$ ώστε $(b, v') \in E$ και $(v', v) \in E$. $v' \in R_{1,G}(b)$, άρα $(a, v') \in E$. Από τις σχέσεις $(a, v') \in E$ και $(v', v) \in E$ έπεται $v \in R_{2,G}(a)$.

■

4.2 Συμβουλές για ημιαποδοτικώς υπολογιστά σύνολα

Σε αυτήν την ενότητα θα ασχοληθούμε με το να καθορίσουμε το μήκος των συμβουλών που απαιτούνται για τα σύνολα της κλάσης **P**-sel, χρησιμοποιώντας ντετερμινιστικούς και μη ντετερμινιστικούς ερμηνευτές συμβουλών πολυωνυμικού χρόνου. Στα δύο κύρια θεωρήματα της ενότητας θα δούμε ότι για ερμηνευτή συμβουλών από το **P** αρκεί πολυωνυμικό (και μάλιστα τετραγωνικό) μήκος συμβουλών, ενώ για ερμηνευτή από το **NP** αρκεί γραμμικό μήκος συμβουλών. Στην απόδειξη των δύο αποτελεσμάτων θα έχουν κεντρικό ρόλο τα θεωρήματα 4.1.13 και 4.1.14.

Θεώρημα 4.2.1 *[Ko83]* $\mathbf{P}\text{-sel} \subseteq \mathbf{P}/\text{poly}$.

Απόδειξη

Έστω ένα σύνολο $L \in \mathbf{P}\text{-sel}$. Λόγω του θεωρήματος 4.1.8 υπάρχει μία αντιμεταθετική συνάρτηση επιλογής $f(\cdot,\cdot)$ για το L. Για να δείξουμε ότι $L \in \mathbf{P}/\text{poly}$, αρκεί να δώσουμε ένα σύνολο $A \in \mathbf{P}$ και μία συνάρτηση $g : \mathbb{N} \to \Sigma^\star$ για τα οποία να ισχύουν:

1. $(\forall x)[x \in L \Leftrightarrow \langle x, g(|x|)\rangle \in A]$, και

2. υπάρχει πολύωνυμο q ώστε $(\forall n)[|g(n)| \leq q(n)]$.[1]

Για κάθε φυσικό n, θεωρούμε το σύνολο $L^{=n}$ των συμβολοακολουθιών μήκους n που ανήκουν στην L. Αν $L^{=n} \neq \emptyset$, η συνάρτηση επιλογής f ορίζει ένα $|L^{=n}|$-τουρνουά $G(V,E)$ επί του $L^{=n}$. Συγκεκριμένα, $V = L^{=n}$ και $E = \{(a,b) : a, b \in L^{=n} \wedge a \neq b \wedge f(a,b) = b\}$. Δηλαδή μεταξύ δύο διαφορετικών κόμβων του τουρνουά, "κερδίζει" αυτός που επιλέγει η συνάρτηση επιλογής. Από το θεώρημα 4.1.13 προκύπτει ότι υπάρχει ένα σύνολο $H_n \subseteq L^{=n}$ για το οποίο ισχύουν:

- $|H_n| \leq \lfloor \log(|L^{=n}| + 1) \rfloor$. Όμως $|L^{=n}| \leq 2^n$ (θεωρούμε πάντα αλφάβητο $\Sigma = \{0,1\}$), άρα $|H_n| \leq n + 1$.

- για κάθε $x \in L^{=n}$ υπάρχει $h \in H_n$ ώστε $x \in R_{1,G}(h)$. Αν $x \notin H_n$, αυτό σημαίνει ότι $(h,x) \in E$ άρα $f(h,x) = x$. Αν $x \in H_n$, τότε για $h = x$ ισχύει και πάλι $f(h,x) = x$. Άρα για κάθε $x \in L^{=n}$ υπάρχει $h \in H_n$ ώστε $f(h,x) = x$.

[1] Ο ορισμός 4.1.1 απαιτεί για κάθε n να ισχύει $|g(n)| = q(n)$. Όμως αν για κάθε n ισχύει $|g(n)| \leq q(n)$ μπορούμε πάντα να προσθέσουμε στην $g(n)$ επιπλέον σύμβολα άνευ σημασίας έτσι ώστε το μήκος της να γίνει ακριβώς ίσο με $q(n)$.

Αν $L^{=n} = \emptyset$ τότε το σύνολο $H_n = \emptyset$ ικανοποιεί τις παραπάνω ιδιότητες, δηλαδή $H_n \subseteq L^{=n}$, $|H_n| \leq n+1$ και $(\forall x \in L^{=n})(\exists h \in H_n)[f(h,x) = x]$.

Είμαστε έτοιμοι τώρα να δώσουμε τη συνάρτηση συμβουλών $g : \mathbb{N} \to \Sigma^\star$ και τον ερμηνευτή συμβουλών $A \in \mathbf{P}$. Ορίζουμε τη συνάρτηση g για κάθε φυσικό n με τον τύπο:

$$g(n) = H_n$$

όπου H_n είναι το σύνολο που ορίσαμε παραπάνω, το οποίο αναπαρίσταται με μία συμβολοακολουθία επί του $\Sigma^\star$ με κάποια λογική κωδικοποίηση[2]. Επειδή το μέγεθος του H_n είναι το πολύ $n+1$ και κάθε στοιχείο του έχει μήκος n, θα υπάρχει κάποιο πολυώνυμο που για κάθε n φράσσει άνω το μήκος της αναπαράστασης του H_n.

Ο ερμηνευτής συμβουλών είναι το σύνολο

$$A = \left\{ \langle x, y\rangle : \begin{array}{l} \text{η } y \text{ είναι λίστα } z \geq 0 \text{ στοιχείων μήκους } |x|, \\ v_1, v_2, \ldots, v_z, \text{ και } (\exists j)[f(v_j, x) = x] \end{array} \right\}$$

Εφόσον χρησιμοποιούμε λογική συνάρτηση σύζευξης και επειδή η f υπολογίζεται σε πολυωνυμικό χρόνο, είναι σαφές ότι $A \in \mathbf{P}$. Θα δείξουμε ότι για κάθε x, $x \in L \Leftrightarrow \langle x, g(|x|)\rangle \in A$. Έστω $x \in L$. Τότε $\langle x, g(|x|)\rangle \in A$ γιατί, όπως είδαμε, υπάρχει $h \in H_{|x|}$ ώστε $f(h,x) = x$. Αντίστροφα, έστω ότι $\langle x, g(|x|)\rangle \in A$. Εξ ορισμού του A, υπάρχει $h \in H_{|x|}$ ώστε $f(h,x) = x$. Όμως $H_{|x|} \subseteq L^{=|x|}$, άρα $h \in L^{=|x|}$. Από τον ορισμό της συνάρτησης επιλογής προκύπτει $x \in L^{=|x|}$, άρα και $x \in L$.

■

Με το θεώρημα που μόλις δείξαμε, εξασφαλίσαμε ότι αρκεί πολυωνυμικό μήκος συμβουλής για να αποφασιστεί σε πολυωνυμικό χρόνο οποιοδήποτε ημιαποδοτικώς υπολογιστό σύνολο. Στην πραγματικότητα, επειδή η συμβουλή αποτελείται από το πολύ $n+1$ συμβολοακολουθίες μήκους n, όπου n το μήκος της εισόδου, αρκούν $O(n^2)$ bits για να κωδικοποιηθεί. Άρα ουσιαστικά έχουμε αποδείξει ότι $\mathbf{P}$-sel $\subseteq \mathbf{P}/$quadratic.

[2] Δηλαδή υπολογιστή και αντιστρέψιμη σε πολυωνυμικό χρόνο.

Θεώρημα 4.2.2 *[Ko83]* $\mathbf{P}\text{-sel} \subseteq \mathbf{P}/\text{quadratic}$.

Στη συνέχεια θα δούμε ότι μπορούμε να αποφασίσουμε τα ημιαποδοτικώς υπολογιστά σύνολα με ακόμη μικρότερο μήκος συμβουλής, αυξάνοντας όμως την ισχύ του ερμηνευτή συμβουλών. Συγκεκριμένα, αν επιτρέψουμε στον ερμηνευτή να εκτελείται σε μη ντετερμινιστικό πολυωνυμικό χρόνο, αρκεί γραμμικό μήκος συμβουλής.

Θεώρημα 4.2.3 *[HT96, HNP98]* $\mathbf{P}\text{-sel} \subseteq \mathbf{NP}/\text{linear}$.

Απόδειξη

Έστω ένα σύνολο $L \in \mathbf{P}\text{-sel}$. Λόγω του θεωρήματος 4.1.8 υπάρχει μία αντιμεταθετική συνάρτηση επιλογής $f(\cdot,\cdot)$ για το L. Για να δείξουμε ότι $L \in \mathbf{NP}/\text{linear}$, αρκεί να δώσουμε ένα σύνολο $A \in \mathbf{NP}$ και μία συνάρτηση $g : \mathbb{N} \to \Sigma^\star$ για τα οποία να ισχύουν:

1. $(\forall x)[x \in L \Leftrightarrow \langle x, g(|x|)\rangle \in A]$, και
2. υπάρχει $q \in \text{linear}$ ώστε $(\forall n)[|g(n)| = q(n)]$.

Για κάθε $n \geq 1$, αν $L^{=n} \neq \emptyset$, θεωρούμε όπως και στην απόδειξη του θεωρήματος 4.2.1 το τουρνουά $G(V,E)$ επί του $L^{=n}$ που ορίζει η συνάρτηση f. Λόγω του θεωρήματος 4.1.14, υπάρχει $w_n \in L^{=n}$ τέτοιο ώστε $L^{=n} = R_{2,G}(w_n)$. Ορίζουμε, για κάθε $n \geq 1$ την τιμή της συνάρτησης g:

$$g(n) = \begin{cases} 1^{n+1} & , \quad \text{αν } L^{=n} = \emptyset \\ 0w_n & , \quad \text{αν } L^{=n} \neq \emptyset \end{cases}$$

Η τιμή $g(0)$ είναι αυθαίρετη, αλλά έχει μήκος ακριβώς 1. Έτσι, για κάθε n ισχύει $|g(n)| = n + 1 = q(n)$.

Για τον ερμηνευτή συμβουλών, ορίζουμε αρχικά το σύνολο

$$A_0 = \{\langle x, 0w\rangle : |w| = |x| \geq 1 \wedge (\exists c)[|c| = |x| \wedge f(w,c) = c \wedge f(c,x) = x]\}$$

Αν $\varepsilon \in L$, τότε ο ερμηνευτής συμβουλών είναι

$$A = A_0 \cup \bigcup_{\alpha \in \Sigma} \{\langle \varepsilon, \alpha\rangle\}$$

ενώ αν $\varepsilon \notin L$ τότε ο ερμηνευτής συμβουλών είναι

$$A = A_0$$

Σε κάθε περίπτωση, $A \in \mathbf{NP}$. Μία μη ντετερμινιστική μηχανή Turing πολυωνυμικού χρόνου που αποφασίζει το A_0 απλά επιλέγει μη ντετερμινιστικά μία συμβολοακολουθία c με μήκος ίσο με $|x|$ και ελέγχει σε ντετερμινιστικό, πλέον, πολυωνυμικό χρόνο τη συνθήκη $f(w,c) = c \wedge f(c,x) = x$. Ο έλεγχος για την περίπτωση εισόδου $\langle \varepsilon, \alpha \rangle$, $\alpha \in \Sigma$, γίνεται τετριμμένα πριν τη μη ντετερμινιστική επιλογή. Θα δείξουμε στη συνέχεια, ολοκληρώνοντας την απόδειξη, ότι για κάθε x ισχύει $x \in L \Leftrightarrow \langle x, g(|x|) \rangle \in A$. Για $x = \varepsilon$ προφανώς ισχύει η ισοδυναμία. Στο εξής υποθέτουμε ότι $x \neq \varepsilon$.

Έστω $x \in L$. Τότε $x \in R_{2,G}(w_{|x|})$. Διακρίνουμε τις περιπτώσεις:

- $x \in R_{1,G}(w_{|x|})$: όπως και στην απόδειξη του θεωρήματος 4.2.1, ισχύει $f(w_{|x|}, x) = x$. Άρα επιλέγοντας $c = x$ προκύπτει ότι $\langle x, g(|x|) \rangle \in A$.

- $x \in R_{2,G}(w_{|x|}) \setminus R_{1,G}(w_{|x|})$: υπάρχει $c \in R_{1,G}(w_{|x|})$ ώστε $(c,x) \in E \Rightarrow f(c,x) = x$. Ακόμη, επειδή $c \in R_{1,G}(w_{|x|})$ ισχύει $f(w_{|x|}, c) = c$. Άρα $\langle x, g(|x|) \rangle \in A$.

Αντίστροφα, έστω $\langle x, g(|x|) \rangle \in A$. Από τον ορισμό του A προκύπτει $g(|x|) = 0w_{|x|}$, και υπάρχει c με μήκος $|c| = |x| = |w_{|x|}|$ που ικανοποιεί τις σχέσεις $f(w_{|x|}, c) = c$ και $f(c,x) = x$. Επειδή $w_{|x|} \in L^{=|x|}$, από τον ορισμό της συνάρτησης επιλογής προκύπτει ότι $c \in L^{=|x|}$ και $x \in L^{=|x|}$. Άρα προφανώς $x \in L$.

$\blacksquare$

Επομένως γραμμικό μήκος συμβουλής είναι αρκετό για να αποφασιστεί σε μη ντετερμινιστικό πολυωνυμικό χρόνο οποιοδήποτε ημιαποδοτικώς υπολογιστό σύνολο. Μάλιστα, από τον τρόπο με τον οποίο ορίστηκε η συνάρτηση συμβουλής στην απόδειξη του θεωρήματος 4.2.3, προκύπτει ότι αρκούν συμβουλές με μήκος $n+1$, όπου n το μήκος της εισόδου.

Θεώρημα 4.2.4 $\mathbf{P}$-sel $\subseteq \mathbf{NP}/\lambda n.(n+1)$.

4.3 Απομόνωση τιμών μη ντετερμινιστικών συναρτήσεων

Οι μη ντετερμινιστικές συναρτήσεις είναι πλειονότιμες συναρτήσεις που οι τιμές τους για κάθε είσοδο εμφανίζονται ως έξοδοι στα διάφορα υπολογιστικά μονοπάτια κάποιας μη ντετερμινιστικής μηχανής Turing. Παρόλο που είναι εύκολο για μία μη ντετερμινιστική συνάρτηση να υπολογίσει, για παράδειγμα, όλες τις αναθέσεις τιμών που ικανοποιούν μία λογική έκφραση, δεν υπάρχει προφανής τρόπος για τη μηχανή Turing που υπολογίζει αυτή τη συνάρτηση να απομονώσει μία από τις αναθέσεις αυτές. Αυτό ίσως αρχικά φαίνεται παράδοξο, όμως οφείλεται στο γεγονός ότι σε έναν μη ντετερμινιστικό υπολογισμό τα υπολογιστικά μονοπάτια είναι ανεξάρτητα μεταξύ τους και δεν μπορούν να "αποφασίσουν" συλλογικά ποιο από όλα θα δώσει έξοδο, ή ποια κοινή συμβολοακολουθία θα δώσουν στην έξοδό τους. Θα δούμε ότι η υπόθεση πως από μία πλειονότιμη μη ντετερμινιστική συνάρτηση μπορούμε να απομονώσουμε μία τιμή εξόδου και να λάβουμε μία μονότιμη μη ντετερμινιστική συνάρτηση έχει σημαντικές συνέπειες για τη θεωρία πολυπλοκότητας, αφού συνεπάγεται την κατάρρευση της πολυωνυμικής ιεραρχίας. Στην πορεία θα ορίσουμε μία γενίκευση των ημιαποδοτικώς υπολογιστών συνόλων, επιτρέποντας στη συνάρτηση επιλογής να είναι μη ντετερμινιστική.

4.3.1 Η πολυωνυμική ιεραρχία

Η πολυωνυμική ιεραρχία είναι μία ιεραρχία κλάσεων πολυπλοκότητας στην οποία κάθε επίπεδο χρησιμοποιεί την κλάση που ορίζεται στο προηγούμενο επίπεδο ως μαντείο. Η βάση της ιεραρχίας είναι η κλάση **P**. Ορίστηκε από τους Meyer και Stockmeyer [MS72, Sto76] σε αναλογία με την ιεραρχία Kleene που μελετάται στην θεωρία των αναδρομικών συναρτήσεων. Είχε όμως ήδη προταθεί έμμεσα από τον Karp [Kar72].

Ορισμός 4.3.1 *Η* πολυωνυμική ιεραρχία *είναι η δομή που προκύπτει από τις κλάσεις* Σ_k^p, Π_k^p *και* Δ_{k+1}^p *που ορίζονται για κάθε* $k \geq 0$ *ως εξής:*

1. $\Sigma_0^p = \Pi_0^p = \mathbf{P}$.

2. $\Delta_{k+1}^p = \mathbf{P}^{\Sigma_k^p}$.

3. $\Sigma_{k+1}^p = \mathbf{NP}^{\Sigma_k^p}$.

4. $\Pi_{k+1}^p = \mathbf{co}\Sigma_{k+1}^p$.

Ορίζουμε ακόμη την ένωση όλων των Σ_i^p*:*

$$\mathbf{PH} = \bigcup_{i \geq 0} \Sigma_i^p$$

Χωρίς απόδειξη, παραθέτουμε τις ακόλουθες ιδιότητες της πολυωνυμικής ιεραρχίας. Για περισσότερες λεπτομέρειες ο αναγνώστης παραπέμπεται στο κεφ. 8 του [BDG88] και στο κεφ. 17 του [Pap94].

Θεώρημα 4.3.2 *Ιδιότητες της πολυωνυμικής ιεραρχίας.*

1. $\mathbf{PH} = \bigcup_{i \geq 0} \Pi_i^p = \bigcup_{i \geq 1} \Delta_i^p$.

2. $\Delta_1^p = \mathbf{P}$.

3. $\Sigma_{k+1}^p = \mathbf{NP}^{\Pi_k^p}$, *για κάθε* $k \geq 0$.

4. $\Delta_{k+1}^p = \mathbf{P}^{\Pi_k^p}$, *για κάθε* $k \geq 0$.

5. $\Pi_{k+1}^p = \mathbf{coNP}^{\Pi_k^p}$, *για κάθε* $k \geq 0$.

6. $\Sigma_{k+1}^p = \mathbf{NP}^{\Delta_{k+1}^p}$, *για κάθε* $k \geq 0$.

7. $\Pi_{k+1}^p = \mathbf{coNP}^{\Delta_{k+1}^p}$, *για κάθε* $k \geq 0$.

8. $\Sigma_k^p \cup \Pi_k^p \subseteq \Delta_{k+1}^p$, *για κάθε* $k \geq 0$.

9. $\Delta_k^p \subseteq \Sigma_k^p \cap \Pi_k^p$, *για κάθε* $k \geq 0$.

10. $\mathbf{PH} \subseteq \mathbf{PSPACE}$.

11. *Η κλάση* Σ_k^p *έχει πλήρεις γλώσσες, για κάθε* $k \geq 0$.

Από τις ιδιότητες 8 και 9 του θεωρήματος 4.3.2 έπεται ότι κάθε επίπεδο της πολυωνυμικής ιεραρχίας συνδέεται με το επόμενό του μέσω των εγκλεισμών:

$$\Sigma_k^p \cup \Pi_k^p \subseteq \Delta_{k+1}^p \subseteq \Sigma_{k+1}^p \cap \Pi_{k+1}^p$$

Παραμένει ανοιχτό το ερώτημα αν κάποιος από αυτούς τους εγκλεισμούς είναι γνήσιος.

Τα δύο πρώτα επίπεδα της πολυωνυμικής ιεραρχίας, Σ_0^p και Σ_1^p είναι οι κλάσεις $\Sigma_0^p = \mathbf{P}$ και $\Sigma_1^p = \mathbf{NP}$. Ισχύει ότι κάθε επίπεδο είναι υποσύνολο του επομένου του, χωρίς να είναι γνωστό αν κάποιος εγκλεισμός είναι γνήσιος. Πάντως, από τον ορισμό της ιεραρχίας προκύπτει ότι αν για κάποιο k ισχύει $\Sigma_k^p \subset \Sigma_{k+1}^p$, τότε ισχύει $\Sigma_i^p \subset \Sigma_{i+1}^p$ για κάθε $i \leq k$. Αντίστοιχα, αν για κάποιο k ισχύει $\Sigma_k^p = \Sigma_{k+1}^p$, τότε ισχύει $\Sigma_i^p = \Sigma_{i+1}^p$ για κάθε $i \geq k$. Σε αυτήν την περίπτωση λέμε ότι η πολυωνυμική ιεραρχία καταρρέει στο k-οστό επίπεδο. Πιστεύεται ότι η πολυωνυμική ιεραρχία είναι γνήσια. Επομένως αν μία πρόταση συνεπάγεται την κατάρρευση της πολυωνυμικής ιεραρχίας σε κάποιο επίπεδο, θεωρείται εξαιρετικά απίθανο να ισχύει.

4.3.2 Μη ντετερμινιστικές συναρτήσεις. Γενικεύσεις της P-sel

Έχουμε χρησιμοποιήσει μέχρι στιγμής την κλάση **FP** των μερικών συναρτήσεων που υπολογίζονται σε πολυωνυμικό χρόνο από ντετερμινιστική μηχανή Turing. Οι συναρτήσεις της κλάσης **FP** είναι μονότιμες. Οι κλάσεις που θα ορίσουμε σε αυτήν την ενότητα περιλαμβάνουν συναρτήσεις που υπολογίζονται με μία ευρύτερη, μη ντετερμινιστική έννοια.

Ορισμός 4.3.3 *[BLS84, BLS85]*

1. *Αν η f είναι μία πλειονότιμη συνάρτηση που απεικονίζει συμβολοακολουθίες σε συμβολοακολουθίες, ορίζουμε για κάθε x το σύνολο* set-$f(x)$ *των τιμών εξόδου της $f(x)$:*

$$\text{set-}f(x) = \{a : a \text{ είναι έξοδος της } f(x)\}$$

Αν η f είναι μερική και δεν ορίζεται στο x, τότε set-$f(x) = \emptyset$.

2. *Για κάθε μη ντετερμινιστική μηχανή Turing N πολυωνυμικού χρόνου, ορίζουμε την πλειονότιμη, μερική συνάρτηση f_N που θεωρούμε ότι υπολογίζει η N. Για κάθε x, η f_N έχει τιμές:*

   $$\text{set-}f_N(x) = \left\{ y : \begin{array}{l} \textit{υπάρχει μονοπάτι της } N \textit{ που με είσοδο } x \textit{ αποδέχεται} \\ \textit{και παράγει έξοδο } y \end{array} \right\}$$

   Ορίζουμε ακόμη την κλάση **NPMV** *των πλειονότιμων μη ντετερμινιστικών συναρτήσεων που υπολογίζονται με τον παραπάνω τρόπο:*

   $$\mathbf{NPMV} = \{f : (\exists N)(\forall x)[\text{set-}f(x) = \text{set-}f_N(x)]\}$$

3. *Μία μερική, πλειονότιμη συνάρτηση f λέγεται μονότιμη αν για κάθε x έχει το πολύ μία τιμή εξόδου:*

   $$(\forall x)[|\text{set-}f(x)| \leq 1]$$

   Η κλάση **NPSV** *των μονότιμων μη ντετερμινιστικών συναρτήσεων περιλαμβάνει εκείνες τις συναρτήσεις της* **NPMV** *που συμβαίνει να είναι μονότιμες:*

   $$\mathbf{NPSV} = \{f \in \mathbf{NPMV} : \textit{η } f \textit{ είναι μονότιμη}\}$$

Οι κλάσεις **NPMV** και **NPSV** ορίζουν την έννοια του μη ντετερμινιστικού υπολογισμού συναρτήσεων. Ισχύει βέβαια $\mathbf{FP} \subseteq \mathbf{NPSV} \subset \mathbf{NPMV}$. Αν για κάθε είσοδο οι τιμές εξόδου μίας πλειονότιμης συνάρτησης f είναι υποσύνολα των αντίστοιχων τιμών εξόδου μίας πλειονότιμης συνάρτησης g, τότε η f αποτελεί εκλέπτυνση της g.

Ορισμός 4.3.4 *[Sel94] Αν οι f, g είναι πλειονότιμες συναρτήσεις, λέμε ότι η f είναι* εκλέπτυνση *της g αν και μόνο αν ισχύουν οι συνθήκες:*

1. $(\forall x)[\text{set-}f(x) = \emptyset \Leftrightarrow \text{set-}g(x) = \emptyset]$, *και*

2. $(\forall x)[\text{set-}f(x) \subseteq \text{set-}g(x)]$.

Διαισθητικά, η εκλέπτυνση μίας συνάρτησης είναι η ίδια η συνάρτηση με τη διαφορά ότι για κάποιες εισόδους μπορεί να λείπουν κάποιες τιμές εξόδου. Δεν επιτρέπεται όμως να λείπουν όλες οι τιμές εξόδου. Το ερώτημα αν κάθε **NPMV** συνάρτηση έχει **NPSV** εκλέπτυνση παραμένει ανοικτό και θα μας απασχολήσει στη συνέχεια. Εδώ σημειώνουμε απλώς ότι το ερώτημα αυτό δεν είναι ισοδύναμο με το ερώτημα $\mathbf{UP} \stackrel{?}{=} \mathbf{NP}$. Η διαφορά έγκειται στο γεγονός ότι μπορεί κάθε **NPMV** συνάρτηση να έχει **NPSV** εκλέπτυνση, όμως η τιμή εξόδου της εκλέπτυνσης να εμφανίζεται σε περισσότερα από ένα υπολογιστικά μονοπάτια της υποβόσκουσας μηχανής Turing. Έτσι ο υπολογισμός της μηχανής αυτής δεν έχει μοναδικό μάρτυρα, άρα η γλώσσα που αποφασίζει δεν μπορεί να ανήκει στην κλάση **UP**.

Προχωρούμε στη γενίκευση των ημιαποδοτικώς υπολογιστών συνόλων, επιτρέποντάς τους να έχουν συνάρτηση επιλογής που ανήκει στην κλάση **NPSV**. Αρχικά ας ορίσουμε την κλάση $\mathcal{F}$-sel όπου $\mathcal{F}$ είναι μία αυθαίρετη κλάση συναρτήσεων.

Ορισμός 4.3.5 *Έστω $\mathcal{F}$ μία κλάση συναρτήσεων, οι οποίες μπορεί να είναι ολικές ή μερικές, μονότιμες ή πλειονότιμες. Ορίζουμε την κλάση $\mathcal{F}$-sel: μία γλώσσα L ανήκει στην $\mathcal{F}$-sel αν και μόνο αν υπάρχει συνάρτηση $f \in \mathcal{F}$ για την οποία ισχύουν:*

1. $(\forall x, y)[\text{set-}f(x, y) \subseteq \{x, y\}]$, *και*
2. $(\forall x, y)[(x \in L \vee y \in L) \Rightarrow \emptyset \subset \text{set-}f(x, y) \subseteq L]$.

Δηλαδή οι πλειονότιμες συναρτήσεις επιλογής δεν επιτρέπεται να επιλέξουν συμβολοακολουθίες που δεν ανήκουν στα ορίσματά τους. Επιπλέον, αν τουλάχιστον ένα από τα ορίσματά τους ανήκει στο σύνολο, τότε πρέπει να επιλέξουν τουλάχιστον ένα όρισμα και δεν επιτρέπεται να επιλέξουν ορίσματα που δεν ανήκουν στο σύνολο. Με βάση τον παραπάνω γενικό ορισμό προκύπτει και ο ορισμός της κλασης **NPSV**-sel.

Ορισμός 4.3.6 *[HNOS96] Η κλάση* **NPSV**-sel *περιέχει ακριβώς εκείνες τις γλώσσες L για κάθε μία από τις οποίες υπάρχει $f \in$* **NPSV** *με τις εξής ιδιότητες:*

1. $(\forall x, y)[\text{set-}f(x, y) \subseteq \{x, y\}]$, *και*

2. $(\forall x, y)[\{x, y\} \cap L \neq \emptyset \Rightarrow \emptyset \subset \text{set-}f(x, y) \subseteq L]$.

Όπως και στην περίπτωση των ημιαποδοτικώς υπολογιστών συνόλων, η **NPSV** συνάρτηση επιλογής ενός συνόλου της κλάσης **NPSV**-sel μπορεί να θεωρείται αντιμεταθετική χωρίς βλάβη της γενικότητας, όπως φαίνεται από το ακόλουθο θεώρημα.

Θεώρημα 4.3.7 *Αν* $L \in \mathbf{NPSV}\text{-sel}$, *τότε η* L *έχει* **NPSV** *συνάρτηση επιλογής* f *για την οποία ισχύει:*

$$(\forall x, y)[\text{set-}f(x, y) = \text{set-}f(y, x)]$$

4.3.3 Δυσκολία εύρεσης μοναδικών λύσεων

Ας υποθέσουμε ότι υπάρχει μία συνάρτηση $f \in \mathbf{FP}$ η οποία ορίζεται για κάθε ικανοποιήσιμη λογική έκφραση ϕ και η $f(\phi)$ είναι μία ανάθεση τιμών που ικανοποιεί την ϕ. Τότε, επειδή το **SAT** είναι **NP**-πλήρες πρόβλημα, μπορούμε χρησιμοποιώντας την f να αποφασίσουμε σε πολυωνυμικό χρόνο οποιαδήποτε γλώσσα του **NP**. Άρα $\mathbf{P} = \mathbf{NP}$. Αντίστροφα, αν $\mathbf{P} = \mathbf{NP}$ τότε $\mathbf{SAT} \in \mathbf{P}$, άρα δοθείσης μιας λογικής έκφρασης ϕ μπορούμε σε πολυωνυμικό χρόνο να υπολογίσουμε μία ανάθεση τιμών που την ικανοποιεί με έναν απλό αλγόριθμο αναζήτησης προθέματος. Βλέπουμε λοιπόν ότι δεν υπάρχει **FP** συνάρτηση που να υπολογίζει μία ανάθεση τιμών που ικανοποιεί μία λογική έκφραση, παρά μόνο αν $\mathbf{P} = \mathbf{NP}$. Από την άλλη μεριά, αν θεωρήσουμε συναρτήσεις της κλάσης $\mathbf{FP}^{\mathbf{NP}}$, δηλαδή υπολογιστές σε πολυωνυμικό χρόνο με χρήση μαντείου από το **NP**, τότε μπορούμε με αναζήτηση προθέματος να υπολογίσουμε μία ανάθεση τιμών που ικανοποιεί τη δοσμένη έκφραση.

Η περίπτωση να υπάρχει **NPSV** συνάρτηση που υπολογίζει μία ανάθεση τιμών που ικανοποιεί δοθείσα έκφραση βρίσκεται κάπου ενδιάμεσα στα δύο παραπάνω ενδεχόμενα. Συγκεκριμένα, όπως θα δούμε, η ύπαρξή της συνεπάγεται την κατάρρευση της πολυωνυμικής ιεραρχίας στο δεύτερο επίπεδο.

Για την απόδειξη θα χρησιμοποιήσουμε ως λήμματα δύο θεωρήματα. Το πρώτο (θεώρημα 4.3.8) εξασφαλίζει ότι αν κάθε σύνολο της κλάσης **NP** μπορεί να αποφασιστεί με πολυωνυμικό μήκος συμβουλής από ερμηνευτή συμβουλών που ανήκει στο $\mathbf{NP} \cap \mathbf{coNP}$, τότε η πολυωνυμική ιεραρχία καταρρέει στο δεύτερο επίπεδο. Δίνεται χωρίς απόδειξη για να μπορέσουμε να επικεντρώσουμε στην τεχνική που χρησιμοποιείται στην απόδειξη του δεύτερου (θεώρημα 4.3.9), η οποία έχει κοινά σημεία με τις αποδείξεις των θεωρημάτων 4.2.1 και 4.2.3.

Θεώρημα 4.3.8 *[KL80, AFK89, Käm91, HHN+95]*
$\mathbf{NP} \subseteq (\mathbf{NP} \cap \mathbf{coNP})/\mathrm{poly} \Rightarrow \mathbf{PH} = \mathbf{NP}^{\mathbf{NP}}$.

Θεώρημα 4.3.9 *[HNOS96]* $\mathbf{NPSV}\text{-sel} \cap \mathbf{NP} \subseteq (\mathbf{NP} \cap \mathbf{coNP})/\mathrm{poly}$.

Απόδειξη

Έστω $L \in \mathbf{NPSV}\text{-sel} \cap \mathbf{NP}$, και ας είναι $f \in \mathbf{NPSV}$ μία αντιμεταθετική συνάρτηση επιλογής της L, N_f μία μη ντετερμινιστική μηχανή Turing πολυωνυμικού χρόνου που υπολογίζει την f υπό την έννοια του ορισμού 4.3.3 και N_L μία μη ντετερμινιστική μηχανή Turing πολυωνυμικού χρόνου που αποφασίζει την L. Για να δείξουμε ότι $L \in (\mathbf{NP} \cap \mathbf{coNP})/\mathrm{poly}$ αρκεί να δώσουμε μία συνάρτηση g και ένα σύνολο $A \in (\mathbf{NP} \cap \mathbf{coNP})$ για τα οποία να ισχύουν:

1. $(\forall x)[x \in L \Leftrightarrow \langle x, g(|x|)\rangle \in A]$, και

2. υπάρχει πολυώνυμο q ώστε $(\forall n)[|g(n)| \leq q(n)]$.

Για κάθε φυσικό n, αν $L^{=n} \neq \emptyset$, θεωρούμε το $|L^{=n}|$-τουρνουά $G(L^{=n}, E)$ που ορίζει η συνάρτηση επιλογής ως εξής:

$$E = \{(a, b) : a, b \in L^{=n} \wedge a \neq b \wedge \text{set-}f(a, b) = \{b\}\}$$

Λόγω του θεωρήματος 4.1.13 υπάρχει ένα σύνολο $H_n \subseteq L^{=n}$ με μέγεθος $|H_n| \leq \lfloor \log(|L^{=n}| + 1)\rfloor \leq n + 1$, για το οποίο ισχύει:

$$(\forall x \in L^{=n})(\exists h \in H_n)[x \in R_{1,G}(h)]$$

Η συνθήκη $x \in R_{1,G}(h)$ ισχύει αν και μόνο αν $x = h$ ή η ακμή (h, x) ανήκει στο γράφο. Σε κάθε περίπτωση, με βάση τον ορισμό του συνόλου E και της

Αλγόριθμος 16 Μη ντετερμινιστικός αλγόριθμος πολυωνυμικού χρόνου που αποφασίζει τον ερμηνευτή συμβουλών για μία **NP** γλώσσα που έχει **NPSV** συνάρτηση επιλογής, μέρος της απόδειξης του θεωρήματος 4.3.9

Είσοδος: $\langle x, \langle\langle a_1, a_2, \ldots, a_z\rangle, \langle w_1, w_2, \ldots, w_{z'}\rangle\rangle\rangle$

if $z \neq z'$ **then**
    τερμάτισε την εκτέλεση σε κατάσταση απόρριψης
end if
for $i \leftarrow 1$ **to** z **do**
    εξομοίωσε την εκτέλεση της N_L με είσοδο a_i και τις μη ντετερμινιστικές επιλογές που καθορίζονται από το w_i
    if η N δεν τερματίζει την εκτέλεσή της σε κατάσταση αποδοχής **then**
        τερμάτισε την εκτέλεση σε κατάσταση απόρριψης
    end if
end for
for $i \leftarrow 1$ **to** z **do**
    εξομοίωσε μη ντετερμινιστικά την εκτέλεση της N_f με είσοδο $\langle a_i, x\rangle$
    if η N_f τερματίζει την εκτέλεσή της σε κατάσταση αποδοχής με έξοδο x **then**
        τερμάτισε την εκτέλεση σε κατάσταση αποδοχής
    end if
end for
τερμάτισε την εκτέλεση σε κατάσταση απόρριψης

Αλγόριθμος 17 Μη ντετερμινιστικός αλγόριθμος πολυωνυμικού χρόνου που αποφασίζει το συμπλήρωμα του ερμηνευτή συμβουλών για μία **NP** γλώσσα που έχει **NPSV** συνάρτηση επιλογής, μέρος της απόδειξης του θεωρήματος 4.3.9

Είσοδος: y

if το y δεν είναι της μορφής $y = \langle x, \langle \langle a_1, a_2, \ldots, a_z \rangle, \langle w_1, w_2, \ldots, w_z \rangle \rangle \rangle$ **then**
    τερμάτισε την εκτέλεση σε κατάσταση αποδοχής
end if
for $i \leftarrow 1$ **to** z **do**
    εξομοίωσε την εκτέλεση της N_L με είσοδο a_i και τις μη ντετερμινιστικές επιλογές που καθορίζονται από το w_i
    if η N δεν τερματίζει την εκτέλεσή της σε κατάσταση αποδοχής **then**
        τερμάτισε την εκτέλεση σε κατάσταση αποδοχής
    end if
end for
if $x \in \{a_1, a_2, \ldots, a_z\}$ **then**
    τερμάτισε την εκτέλεση σε κατάσταση απόρριψης
end if
for $i \leftarrow 1$ **to** z **do**
    εξομοίωσε μη ντετερμινιστικά την εκτέλεση της N_f με είσοδο $\langle a_i, x \rangle$
    if η N_f τερματίζει την εκτέλεσή της σε κατάσταση απόρριψης ή τερματίζει την εκτέλεσή της σε κατάσταση αποδοχής με έξοδο x **then**
        τερμάτισε την εκτέλεση σε κατάσταση απόρριψης
    end if
end for
τερμάτισε την εκτέλεση σε κατάσταση αποδοχής

συναρτησης επιλογής, ισχύει set-$f(h, x) = \{x\}$. Άρα το σύνολο H_n έχει την ιδιότητα:

$$(\forall x \in L^{=n})(\exists h \in H_n)[\text{set-}f(h, x) = \{x\}]$$

Αν $L^{=n} = \emptyset$, ορίζουμε $H_n = \emptyset$. Για κάθε $n \geq 0$ ορίζουμε τη συνάρτηση συμβουλών g:

$$g(n) = \langle\langle h_1, h_2, \dots, h_{|H_n|}\rangle, \langle w_1, w_2, \dots, w_{|H_n|}\rangle\rangle$$

όπου $h_1, h_2, \dots, h_{|H_n|}$ είναι ακριβώς τα στοιχεία του H_n, και για κάθε i το w_i είναι μάρτυρας της αποδοχής του h_i από την N_L.

Ο ερμηνευτής συμβουλών είναι το σύνολο:

$$\begin{aligned} A = \{\langle x, \langle\langle a_1, a_2, \dots, a_z\rangle, \langle w_1, w_2, \dots, w_{z'}\rangle\rangle\rangle : z = z' \wedge \\ (\forall i : 1 \leq i \leq z)[w_i \text{ είναι μάρτυρας του } a_i \in L] \wedge \\ (\exists i : 1 \leq i \leq z)[\text{set-}f(a_i, x) = \{x\}]\} \end{aligned}$$

Οι αλγόριθμοι 16 και 17 είναι μη ντετερμινιστικοί αλγόριθμοι πολυωνυμικού χρόνου που αποφασίζουν τα A και $\overline{A}$, αντίστοιχα. Άρα $A \in \mathbf{NP} \cap \mathbf{coNP}$.

Θα δείξουμε ότι για κάθε x, $x \in L \Leftrightarrow \langle x, g(|x|)\rangle \in A$. Έστω $x \in L$. Τότε $H_{|x|} \subseteq L^{=|x|}$ και έχουμε δει ότι υπάρχει $h \in H_{|x|}$ ώστε set-$f(h, x) = \{x\}$. Άρα $\langle x, g(|x|)\rangle \in A$. Για την αντίστροφη κατεύθυνση, έστω ότι $g(|x|) = \langle\langle h_1, h_2, \dots, h_{|x|}\rangle, \langle w_1, w_2, \dots, w_{|x|}\rangle\rangle$ και $\langle x, g(|x|)\rangle \in A$. Από τον ορισμό του A, όλα τα h_i ανήκουν στο $L^{=|x|}$ και υπάρχει h_i ώστε set-$f(h_i, x) = \{x\}$. Από τον ορισμό της συνάρτησης επιλογής έπεται ότι $x \in L$.

■

Θεώρημα 4.3.10 *[HNOS96] Αν κάθε* **NPMV** *συνάρτηση έχει* **NPSV** *εκλέπτυνση, τότε* $\mathbf{PH} = \mathbf{NP}^{\mathbf{NP}}$.

Απόδειξη

Λόγω των θεωρημάτων 4.3.8 και 4.3.9, αρκεί να δείξουμε ότι αν κάθε **NPMV** συνάρτηση έχει **NPSV** εκλέπτυνση, τότε $\mathbf{NP} \subseteq \mathbf{NPSV}\text{-sel} \cap \mathbf{NP}$.

Θεωρούμε την πλειονότιμη συνάρτηση $f_{\mathbf{SAT}}$ η οποία ορίζεται για κάθε x, y από τη σχέση

$$\text{set-}f_{\mathbf{SAT}}(x,y) = \{x,y\} \cap \mathbf{SAT}$$

Ισχύει ότι $f_{\mathbf{SAT}} \in \mathbf{NPMV}$. Η μηχανή Turing που την υπολογίζει, σε είσοδο $\langle x,y \rangle$ επιλέγει μη ντετερμινιστικά ένα από τα x, y και μία ανάθεση τιμών για την επιλεγμένη λογική έκφραση, και αποδέχεται αν και μόνο αν η ανάθεση που επέλεξε ικανοποιεί την έκφραση. Τα μονοπάτια που καταλήγουν σε κατάσταση αποδοχής δίνουν ως έξοδο την επιλεγμένη λογική έκφραση.

Εφόσον έχουμε υποθέσει ότι κάθε **NPMV** συνάρτηση έχει **NPSV** εκλέπτυνση, ας είναι η $g_{\mathbf{SAT}}$ μία **NPSV** εκλέπτυνση της $f_{\mathbf{SAT}}$. Παρατηρούμε ότι η $g_{\mathbf{SAT}}$ ορίζεται αν και μόνο αν τουλάχιστον ένα από τα ορίσματά της είναι ικανοποιήσιμη λογική έκφραση. Επιπλέον, είναι **NPSV** συνάρτηση και αν τουλάχιστον ένα από τα ορίσματά της είναι ικανοποιήσιμη λογική έκφραση τότε και η τιμή εξόδου της είναι ικανοποιήσιμη λογική έκφραση. Κατά συνέπεια η $g_{\mathbf{SAT}}$ είναι μία **NPSV** συνάρτηση επιλογής για το **SAT**, από το οποίο έπεται ότι $\mathbf{SAT} \in \mathbf{NPSV}\text{-sel}$.

Θα δείξουμε ότι κάθε γλώσσα της κλάσης **NP** έχει **NPSV** συνάρτηση επιλογής. Έστω $L \in \mathbf{NP}$. Επειδή το **SAT** είναι πλήρες για την κλάση **NP** η L ανάγεται πολυωνυμικά στο **SAT**, δηλαδή υπάρχει μία $g \in \mathbf{FP}$ ώστε $(\forall x)[x \in L \Leftrightarrow g(x) \in \mathbf{SAT}]$. Θεωρούμε τη συνάρτηση g_L που ορίζεται ως εξής:

$$\text{set-}g_L(x,y) = \begin{cases} \{x\} & , \quad \text{αν set-}g_{\mathbf{SAT}}(g(x),g(y)) = \{g(x)\} \\ \{y\} & , \quad \text{αν set-}g_{\mathbf{SAT}}(g(x),g(y)) = \{g(y)\} \\ \emptyset & , \quad \text{αλλιώς} \end{cases}$$

Η g_L είναι **NPSV** συνάρτηση επιλογής για την L, άρα $L \in \mathbf{NPSV}\text{-sel}$. Καταλήγουμε λοιπόν στο ζητούμενο,

$$\mathbf{NP} \subseteq \mathbf{NPSV}\text{-sel} \Rightarrow \mathbf{NP} \subseteq \mathbf{NPSV}\text{-sel} \cap \mathbf{NP}$$

■

Το χαρακτηριστικό που εκμεταλλευόμαστε κυρίως για την απόδειξη των θεωρημάτων 4.2.1, 4.2.3 και 4.3.9 είναι ότι ο ορισμός 4.1.1 δεν επιβάλλει κανέναν περιορισμό στην πολυπλοκότητα υπολογισμού της συνάρτησης συμβουλών, αλλά μόνο στο μήκος των τιμών της. Για το λόγο αυτό μας αρκεί, στα γραφοθεωρητικά αποτελέσματα 4.1.13 και 4.1.14, να εξασφαλίσουμε την ύπαρξη του εκάστοτε υποσυνόλου των κόμβων του γράφου χωρίς να μας ενδιαφέρει πόσο δύσκολο είναι στην πραγματικότητα να υπολογιστεί ένα τέτοιο υποσύνολο.

Στα θεωρήματα 4.2.1 και 4.2.3 η εφαρμογή των αποτελεσμάτων για τους γράφους τουρνουά είναι άμεση. Απεικονίζοντας το πρόβλημα σε ένα τουρνουά μέσω της συνάρτησης επιλογής ("κερδίζει όποιος επιλέγεται από τη συνάρτηση επιλογής"), η ύπαρξη του "μικρού" υποσυνόλου των κόμβων παρέχει εύκολα τόσο την ίδια τη συνάρτηση συμβουλών όσο και το άνω φράγμα στο μήκος των τιμών της.

Στο θεώρημα 4.3.9 η κατάσταση είναι λίγο πιο πολύπλοκη. Θέλουμε να δείξουμε ότι ο ερμηνευτής συμβουλών ανήκει στο $\mathbf{NP} \cap \mathbf{coNP}$. Για την απόδειξη ότι ανήκει στο **NP** θα μας αρκούσε η ίδια συνάρτηση συμβουλών που χρησιμοποιήσαμε και στο θεώρημα 4.2.1, δηλαδή $g(n) = H_n$. Ο μη ντετερμινιστικός αλγόριθμος που αποφασίζει τον ερμηνευτή συμβουλών, σε είσοδο $\langle x, \langle a_1, a_2, \ldots, a_z \rangle \rangle$ θα μπορούσε μη ντετερμινιστικά να ελέγξει αν κάθε ένα από τα a_i ανήκει στην L, και να απόρριψει την είσοδο αν έστω και ένα δεν ανήκει. Όμως ο μη ντετερμινιστικός αλγόριθμος για το συμπλήρωμα του ερμηνευτή συμβουλών θα έπρεπε να αποδέχεται την είσοδο αν υπάρχει ένα $a_i \notin L$. Επειδή δεν είναι γνωστό ότι $L \in \mathbf{coNP}$, αυτό δεν μπορεί να γίνει με απευθείας εξομοίωση μη ντετερμινιστικού αλγορίθμου για την $\overline{L}$. Αναγκαζόμαστε έτσι να συμπεριλάβουμε στη συμβουλή και έναν μάρτυρα για κάθε a_i, ώστε ο έλεγχος $a_i \in L$ να μπορεί να γίνεται σε ντετερμινιστικό πολυωνυμικό χρόνο.

Κεφάλαιο 5

Ελάττωση μαρτύρων

Οι κλάσεις συναρτήσεων που έχουμε συναντήσει μέχρι στιγμής είναι οι κλάσεις των μερικών μονότιμων συναρτήσεων που υπολογίζονται σε ντετερμινιστικό πολυωνυμικό χρόνο, των μερικών πλειονότιμων συναρτήσεων που υπολογίζονται σε μη ντετερμινιστικό πολυωνυμικό χρόνο και των μερικών μονότιμων συναρτήσεων που υπολογίζονται σε μη ντετερμινιστικό πολυωνυμικό χρόνο. Και στις τρεις κλάσεις, οι τιμές των συναρτήσεων προκύπτουν από τις εξόδους των θεωρούμενων μηχανών Turing στα μονοπάτια που καταλήγουν σε κατάσταση αποδοχής.

Στο κεφάλαιο αυτό θα θεωρήσουμε συναρτήσεις που υπολογίζονται όχι από τις τιμές των εξόδων, αλλά από το πλήθος των μονοπατιών που καταλήγουν σε κατάσταση αποδοχής. Σκοπός είναι να διερευνήσουμε κατά πόσο οι νέες κλάσεις συναρτήσεων που θα ορίσουμε είναι κλειστές ως προς διάφορες απλές αριθμητικές πράξεις. Στην προσπάθεια απάντησης αυτού του ερωτήματος θα αναπτύξουμε μία θεωρία που έχει ομοιότητα, τουλάχιστον ως προς τη διατύπωση, με τη θεωρία της **NP**-πληρότητας. Η παρουσίαση αυτού του κεφαλαίου βασίζεται σε εργασία των Ogihara, Hemachandra και Gupta [OH93, Gup95].

5.1 Εισαγωγικά

Για το παρόν κεφάλαιο θα χρησιμοποιούμε φυσικούς αριθμούς και συμβολοακολουθίες χωρίς διάκριση. Βασιζόμαστε βέβαια στον ισομορφισμό μεταξύ των δύο συνόλων, που σε κάθε φυσικό αριθμό n αντιστοιχίζει την $(n+1)$-οστή συμβολοακολουθία κατά τη λεξικογραφική διάταξη[1]. Πρόκειται για μία απεικόνιση που είναι υπολογιστή και αντιστρέψιμη σε πολυωνυμικό χρόνο. Όταν λέμε ότι μία μηχανή Turing υπολογίζει μία συνάρτηση από φυσικούς σε φυσικούς, για παράδειγμα την πρόσθεση, θα εννοούμε ότι παράγει στην έξοδο τη συμβολοακολουθία που αντιστοιχεί στο άθροισμα των φυσικών που αντιστοιχούν στις δύο συμβολοακολουθίες εισόδου.

5.1.1 Υπολογισμοί με βάση το πλήθος των μαρτύρων

Όπως έχουμε δει, κάθε μη ντετερμινιστική μηχανή Turing N πολυωνυμικού χρόνου ορίζει μία γλώσσα L, η οποία περιλαμβάνει ακριβώς εκείνες τις συμβολοακολουθίες x για κάθε μία από τις οποίες η N με είσοδο x έχει τουλάχιστον ένα υπολογιστικό μονοπάτι που καταλήγει σε κατάσταση αποδοχής. Οι γλώσσες που ορίζονται με αυτόν τον τρόπο αποτελούν την κλάση **NP**. Ο αντίστοιχος ορισμός για την κλάση **coNP** απαιτεί ότι η είσοδος της μηχανής Turing ανήκει στη γλώσσα αν και μόνο αν όλα τα υπολογιστικά μονοπάτια καταλήγουν σε κατάσταση αποδοχής.

Παρατηρούμε ότι και στις δύο περιπτώσεις ο τρόπος με τον οποίο λειτουργεί η μηχανή Turing είναι ίδιος, αλλά για να αποφασίσουμε αν η είσοδος ανήκει στη γλώσσα ή όχι ελέγχουμε διαφορετική συνθήκη για το πλήθος των μαρτύρων. Συγκεκριμένα, στην πρώτη περίπτωση αποδεχόμαστε αν και μόνο αν υπάρχει τουλάχιστον ένας μάρτυρας. Στη δεύτερη περίπτωση αποδεχόμαστε αν και μόνο αν όλα τα υπολογιστικά μονοπάτια καταλήγουν σε κατάσταση αποδοχής, δηλαδή υπάρχουν ακριβώς $2^{p(|x|)}$ μάρτυρες όπου x η είσοδος και p το πολυώνυμο που καθορίζει το ύψος του δέντρου υπολογισμού.

[1] Μεταξύ συμβολοακολουθιών διαφορετικού μήκους, αυτή με το μικρότερο μήκος είναι και λεξικογραφικά μικρότερη.

Τροποποιώντας τη συνθήκη που ελέγχουμε για το πλήθος των μαρτύρων, καταλήγουμε στον ορισμό των ακόλουθων κλάσεων πολυπλοκότητας.

Ορισμός 5.1.1 *1. [Sim75, Gil77] Η κλάση* **PP** *περιέχει ακριβώς εκείνες τις γλώσσες L, για κάθε μία από τις οποίες υπάρχει ένα πολυώνυμο q και μία γλώσσα $A \in \mathbf{P}$ ώστε για κάθε x να ισχύει:*

$$x \in L \Leftrightarrow |\{y : |y| = q(|x|) \wedge \langle x, y \rangle \in A\}| \geq 2^{q(|x|)-1}$$

2. *[Sim75, Wag86] Η κλάση* $\mathbf{C_=P}$ *περιέχει ακριβώς εκείνες τις γλώσσες L, για κάθε μία από τις οποίες υπάρχει ένα πολυώνυμο q, μία συνάρτηση $f \in \mathbf{FP}$ και μία γλώσσα $A \in \mathbf{P}$ ώστε για κάθε x να ισχύει:*

$$x \in L \Leftrightarrow |\{y : |y| = q(|x|) \wedge \langle x, y \rangle \in A\}| = f(x)$$

3. *[OH93, FFK94] Η κλάση* **SPP** *περιέχει ακριβώς εκείνες τις γλώσσες L, για κάθε μία από τις οποίες υπάρχει ένα πολυώνυμο q, μία συνάρτηση $f \in \mathbf{FP}$ και μία γλώσσα $A \in \mathbf{P}$ ώστε για κάθε x να ισχύουν τα παρακάτω:*

   - $x \in L \Rightarrow |\{y : |y| = q(|x|) \wedge \langle x, y \rangle \in A\}| = f(x)$
   - $x \notin L \Rightarrow |\{y : |y| = q(|x|) \wedge \langle x, y \rangle \in A\}| = f(x) - 1$

4. *[PZ83, GP86] Η κλάση* $\oplus\mathbf{P}$ *περιέχει ακριβώς εκείνες τις γλώσσες L, για κάθε μία από τις οποίες υπάρχει ένα πολυώνυμο q και μία γλώσσα $A \in \mathbf{P}$ ώστε για κάθε x να ισχύει:*

$$x \in L \Leftrightarrow |\{y : |y| = q(|x|) \wedge \langle x, y \rangle \in A\}| \not\equiv 0(\mathrm{mod} 2)$$

5. *[CH90] Για κάθε $k \geq 2$, η κλάση* $\mathbf{Mod_kP}$ *περιέχει ακριβώς εκείνες τις γλώσσες L, για κάθε μία από τις οποίες υπάρχει ένα πολυώνυμο q και μία γλώσσα $A \in \mathbf{P}$ ώστε για κάθε x να ισχύει:*

$$x \in L \Leftrightarrow |\{y : |y| = q(|x|) \wedge \langle x, y \rangle \in A\}| \not\equiv 0(\mathrm{mod} k)$$

Για τις κλάσεις $\mathbf{C_{=}P}$ και $\mathbf{SPP}$ υπάρχουν και οι ακόλουθοι ισοδύναμοι εναλλακτικοί ορισμοί.

Ορισμός 5.1.2 *1. Η κλάση $\mathbf{C_{=}P}$ περιέχει ακριβώς εκείνες τις γλώσσες L, για κάθε μία από τις οποίες υπάρχει ένα πολυώνυμο q και μία γλώσσα $A \in \mathbf{P}$ ώστε για κάθε x να ισχύει:*

$$x \in L \Leftrightarrow |\{y : |y| = q(|x|) \wedge \langle x, y \rangle \in A\}| = 2^{q(|x|)-1}$$

2. Η κλάση $\mathbf{SPP}$ περιέχει ακριβώς εκείνες τις γλώσσες L, για κάθε μία από τις οποίες υπάρχει ένα πολυώνυμο q, ένα πολυώνυμο p και μία γλώσσα $A \in \mathbf{P}$ ώστε για κάθε x να ισχύουν τα παρακάτω:

- $x \in L \Rightarrow |\{y : |y| = q(|x|) \wedge \langle x, y \rangle \in A\}| = 2^{p(|x|)} + 1$
- $x \notin L \Rightarrow |\{y : |y| = q(|x|) \wedge \langle x, y \rangle \in A\}| = 2^{p(|x|)}$

Οι κλάσεις που ορίσαμε συνδέονται μεταξύ τους, καθώς και με άλλες κλάσεις που έχουμε ορίσει νωρίτερα, με τους εξής εγκλεισμούς:

$$\mathbf{P} \subseteq \mathbf{UP} \subseteq \mathbf{SPP} \subseteq \oplus\mathbf{P} \subseteq \mathbf{PP}$$

$$\mathbf{US} \subseteq \mathbf{C_{=}P} \subseteq \mathbf{PP}$$

$$\mathbf{UP} \subseteq \mathbf{SPP} \subseteq \oplus\mathbf{P} \cap \mathbf{C_{=}P}$$

5.1.2 Οι κλάσεις #P, OptP

Ορισμός 5.1.3 *[Val79] Η κλάση συναρτήσεων $\#\mathbf{P}$ περιέχει ακριβώς εκείνες τις συναρτήσεις f, για κάθε μία από τις οποίες υπάρχει μία μη ντετερμινιστική μηχανή Turing N πολυωνυμικού χρόνου ώστε για κάθε x το πλήθος των υπολογιστικών μονοπατιών της N με είσοδο x που καταλήγουν σε κατάσταση αποδοχής να είναι ίσο με $f(x)$.*

Η $\#\mathbf{P}$ είναι λοιπόν η κλάση των συναρτήσεων που μετρούν το πλήθος των μαρτύρων της εισόδου τους. Για παράδειγμα, η συνάρτηση που σε κάθε λογική έκφραση απεικονίζει το πλήθος των αναθέσεων τιμών που την ικανοποιούν ανήκει στην **#P** και μάλιστα είναι **#P**-πλήρης, ως προς πολυωνυμική αναγωγή που διατηρεί το πλήθος των λύσεων (φειδωλή αναγωγή).

Η δεύτερη κλάση συναρτήσεων που θα μας απασχολήσει συλλαμβάνει την έννοια του προβλήματος μεγιστοποίησης. Υποτίθεται ότι κάθε υπολογιστικό μονοπάτι μίας μη ντετερμινιστικής μηχανής Turing παράγει ως έξοδο έναν φυσικό αριθμό. Αν δεν παράγει ρητά μία τέτοια έξοδο, θεωρούμε ότι αυτή είναι ο αριθμός 0. Με κάθε μηχανή Turing αυτού του τύπου συσχετίζουμε τη συνάρτηση που απεικονίζει την είσοδο x στη μέγιστη τιμή που εμφανίζεται ως έξοδος σε κάποιο υπολογιστικό μονοπάτι. Η κλάση των συναρτήσεων που ορίζονται με αυτόν τον τρόπο από μη ντετερμινιστικές μηχανές Turing πολυωνυμικού χρόνου είναι η **OptP**.

Ορισμός 5.1.4 *[Kre88] Η κλάση συναρτήσεων* **OptP** *περιέχει ακριβώς εκείνες τις συναρτήσεις f για κάθε μία από τις οποίες υπάρχει μία μη ντετερμινιστική μηχανή Turing N πολυωνυμικού χρόνου ώστε για κάθε x να ισχύει:*

$$f(x) = \max \left\{ i \in \mathbb{N} : \begin{array}{l} \textit{υπάρχει υπολογιστικό μονοπάτι της } N \\ \textit{με είσοδο } x \textit{ που παράγει έξοδο } i \end{array} \right\}$$

5.2 Κατηγοριοποίηση των ιδιοτήτων κλειστότητας

Οι πράξεις στις οποίες θα αναφερόμαστε είναι απεικονίσεις από ζεύγη φυσικών σε φυσικούς. Λέμε ότι μία κλάση συναρτήσεων είναι κλειστή ως προς μία πράξη, αν η συνάρτηση που προκύπτει από την εφαρμογή της πράξης σε δύο οποιεσδήποτε συναρτήσεις της κλάσης εξακολουθεί να ανήκει στην κλάση.

Ορισμός 5.2.1 *Αν $\sigma : \mathbb{N} \times \mathbb{N} \to \mathbb{N}$ είναι μία πράξη και $\mathcal{F}$ είναι μία κλάση συναρτήσεων που απεικονίζουν φυσικούς σε φυσικούς, λέμε ότι η $\mathcal{F}$ είναι* κλειστή ως προς την πράξη σ *αν για κάθε $f_1, f_2 \in \mathcal{F}$ ισχύει $\sigma(f_1, f_2) \in \mathcal{F}$, όπου $\sigma(f_1, f_2)$ είναι μία συνάρτηση που για κάθε n ορίζεται ως εξής: $\sigma(f_1, f_2)(n) = \sigma(f_1(n), f_2(n))$.*

Για παράδειγμα, η κλάση **#P** είναι κλειστή ως προς πρόσθεση και ως προς πολλαπλασιασμό, όπως φαίνεται από τα δύο επόμενα θεωρήματα.

Θεώρημα 5.2.2 *Η* $\#\mathbf{P}$ *είναι κλειστή ως προς πρόσθεση.*

Απόδειξη

Έστω $f_1, f_2 \in \#\mathbf{P}$, και ας είναι N_1 (αντ. N_2) η μη ντετερμινιστική μηχανή Turing πολυωνυμικού χρόνου που για κάθε είσοδο x έχει $f_1(x)$ (αντ. $f_2(x)$) υπολογιστικά μονοπάτια που καταλήγουν σε κατάσταση αποδοχής. Θεωρούμε τη μη ντετερμινιστική μηχανή Turing N, που για κάθε είσοδο x επιλέγει μη ντετερμινιστικά μία από τις N_1, N_2 και εξομοιώνει τον υπολογισμό της με είσοδο x. Από την εξομοίωση της N_1 έχει $f_1(x)$ μονοπάτια που αποδέχονται, και από την εξομοίωση της N_2 έχει $f_2(x)$ μονοπάτια που αποδέχονται. Άρα $(f_1 + f_2) \in \#\mathbf{P}$.

■

Θεώρημα 5.2.3 *Η* $\#\mathbf{P}$ *είναι κλειστή ως προς πολλαπλασιασμό.*

Απόδειξη

Θεωρούμε και πάλι τις $f_1, f_2 \in \#\mathbf{P}$ και τις αντίστοιχες μη ντετερμινιστικές μηχανές Turing πολυωνυμικού χρόνου, N_1 και N_2. Έστω η μη ντετερμινιστική μηχανή Turing N, που για κάθε είσοδο x εξομοιώνει τη λειτουργία της N_1 με είσοδο x. Σε κάθε υπολογιστικό μονοπάτι, αν η N_1 απορρίπτει την είσοδο, τότε και η N απορρίπτει την είσοδο. Αν η N_1 αποδέχεται, τότε η N εξομοιώνει την N_2. Κάθε ένα από τα $f_1(x)$ μονοπάτια που αποδέχονται στην εξομοίωση της N_1 καταλήγει σε $f_2(x)$ μονοπάτια που αποδέχονται στην εξομοίωση της N_2. Άρα συνολικά έχουμε $f_1(x) \cdot f_2(x)$ μονοπάτια που αποδέχονται, και $(f_1 \cdot f_2) \in \#\mathbf{P}$.

■

Στα επόμενα θα μας απασχολήσει το ερώτημα κατά πόσο η κλάση $\#\mathbf{P}$ είναι κλειστή ως προς ορισμένες πολυωνυμικά υπολογιστές πράξεις. Με τον όρο αυτό εννοούμε τις πράξεις σ για τις οποίες υπάρχει ντετερμινιστικός αλγόριθμος πολυωνυμικού χρόνου που για κάθε ζεύγος εισόδων x, y υπολογίζει την τιμή $\sigma(x, y)$. Θα δούμε ότι το ερώτημα αν η $\#\mathbf{P}$ είναι κλειστή ως προς αφαίρεση[2] είναι τόσο σημαντικό για τις ιδιότητες κλειστότητας της $\#\mathbf{P}$, όσο σημαντικό είναι και το ερώτημα αν $\mathbf{SAT} \in \mathbf{P}$ για τη θεωρία της **NP**-πληρότητας. Εκεί, γνωρίζουμε αρκετά προβλήματα που ανήκουν στην κλάση **P**, αλλά για το **SAT** το ερώτημα δεν έχει ακόμη απαντηθεί. Είναι γνωστό όμως ότι το **SAT** είναι **NP**-πλήρες, δηλαδή αν $\mathbf{SAT} \in \mathbf{P}$ (ή αν οποιοδήποτε άλλο **NP**-πλήρες πρόβλημα ανήκει στην **P**) τότε όλα τα προβλήματα της κλάσης **NP** επίσης ανήκουν στην **P**. Εντελώς αντίστοιχα, γνωρίζουμε ότι η $\#\mathbf{P}$ είναι κλειστή ως προς πρόσθεση, πολλαπλασιασμό και άλλες πολυωνυμικά υπολογιστές πράξεις, αλλά δεν γνωρίζουμε αν είναι κλειστή ως προς αφαίρεση. Μάλιστα, όπως θα δούμε στην επόμενη ενότητα, αν η $\#\mathbf{P}$ είναι κλειστή ως προς αφαίρεση τότε είναι κλειστή ως προς οποιαδήποτε πολυωνυμικά υπολογιστή πράξη. Επομένως η αφαίρεση είναι κατά μία έννοια "πλήρης" πράξη. Σε αυτήν την κατηγορία ανήκει και η ακέραια διαίρεση. Για άλλες πράξεις, όπως είναι η εύρεση ελαχίστου και μεγίστου, η μείωση κατά ένα και ο υποδιπλασιασμός, δεν είναι γνωστό αν η $\#\mathbf{P}$ είναι κλειστή ως προς αυτές, αλλά ούτε έχει αποδειχθεί ότι είναι "πλήρεις". Αυτές βρίσκουν ανάλογο στη θεωρία **NP**-πληρότητας σε ενδιάμεσα προβλήματα όπως είναι ο ισομορφισμός γράφων, για τα οποία δεν έχει βρεθεί ντετερμινιστικός πολυωνυμικός αλγόριθμος αλλά ούτε έχει αποδειχθεί ότι είναι **NP**-πλήρη.

Παρατηρούμε λοιπόν ότι ενώ είναι δυνατόν να αυξήσουμε τους μάρτυρες ενός υπολογισμού μη ντετερμινιστικής μηχανής Turing πολυωνυμικού χρόνου, η ελάττωση τους διαισθητικά είναι πιο δύσκολη και πιθανώς να είναι αδύνατη. Μάλιστα, θα δούμε πως η υπόθεση ότι μπορούμε να μειώσουμε έστω και κατά ένα το πλήθος των μαρτύρων έχει σημαντικές συνέπειες για τη θεωρία πολυπλοκότητας, αφού συνεπάγεται την κατάρρευση κλάσεων πολυπλοκότητας που πιστεύεται ευρέως ότι είναι διαφορετικές μεταξύ τους. Στις επόμενες ενότητες θα χαρακτηρίσουμε κάποιες πράξεις επί συναρτήσεων της κλάσης $\#\mathbf{P}$ ως "πλήρεις" ή ενδιάμεσες, διατηρώντας την ορολογία που προκύπτει από την πα-

[2]Επειδή αναφερόμαστε σε αφαίρεση φυσικών, περιοριζόμαστε στην "γνήσια" αφαίρεση $\ominus$, που ορίζεται ως εξής: $a \ominus b = \max\{0, a - b\}$.

ραπάνω αναλογία. Παράλληλα, υποθέτοντας ότι η $\#\mathbf{P}$ είναι κλειστή ως προς την εκάστοτε "πλήρη" ή ενδιάμεση πράξη θα καταλήξουμε σε κατάρρευση ορισμένων κλάσεων πολυπλοκότητας, γεγόνος που υποδεικνύει ότι κατά πάσα πιθανότητα η $\#\mathbf{P}$ δεν είναι κλειστή ως προς τη θεωρούμενη πράξη.

5.3 Κλειστότητα της $\#\mathbf{P}$ ως προς αφαίρεση

Θεώρημα 5.3.1 *Οι ακόλουθες προτάσεις είναι ισοδύναμες:*

1. *Η* $\#\mathbf{P}$ *είναι κλειστή ως προς αφαίρεση.*

2. *Η* $\#\mathbf{P}$ *είναι κλειστή ως προς οποιαδήποτε πολυωνυμικά υπολογιστή πράξη.*

3. $\mathbf{UP} = \mathbf{PP}$.

Απόδειξη

Η συνεπαγωγή (2) $\Rightarrow$ (1) ισχύει, αφού η αφαίρεση είναι πολυωνυμικά υπολογιστή πράξη.

Θα αποδείξουμε τη συνεπαγωγή (1) $\Rightarrow$ (3). Έστω ότι η $\#\mathbf{P}$ είναι κλειστή ως προς αφαίρεση. Αρκεί να δείξουμε ότι $\mathbf{PP} \subseteq \mathbf{coNP}$ και $\mathbf{coNP} \subseteq \mathbf{PP}$, αφού από τις σχέσεις αυτές προκύπτει $\mathbf{PP} \subseteq \mathbf{UP}$, άρα $\mathbf{UP} = \mathbf{PP}$ αφού είναι γνωστό ότι $\mathbf{UP} \subseteq \mathbf{PP}$.

Έστω $L \in \mathbf{PP}$. Από τον ορισμό 5.1.1 έπεται ότι υπάρχει ένα πολυώνυμο q και μία γλώσσα $A \in \mathbf{P}$ ώστε η L να μπορεί να οριστεί ως

$$L = \{x : |\{y : |y| = q(|x|) \wedge \langle x, y\rangle \in A\}| \geq 2^{q(|x|)-1}\}$$

Μάλιστα, μπορούμε χωρίς βλάβη της γενικότητας να θεωρήσουμε ότι για κάθε n, $q(n) \geq 1$. Αυτό συμβαίνει γιατί αν τα q, A ικανοποιούν τον ορισμό της $\mathbf{PP}$ για την L, τότε και τα q', A' τον ικανοποιούν, όπου για κάθε n το q' ορίζεται ως $q'(n) = q(n) + 1$, και $A' = \{\langle x, y0\rangle : \langle x, y\rangle \in A\} \cup \{\langle x, y1\rangle : \langle x, y\rangle \in A\}$. Θεωρούμε τη μη ντετερμινιστική μηχανή Turing N, που για κάθε είσοδο x επιλέγει μη ντετερμινιστικά μία συμβολοακολουθία y με μήκος $q(|x|)$ και αποδέχεται αν και μόνο αν $\langle x, y\rangle \in A$. Η N ορίζει μία συνάρτηση $f \in \#\mathbf{P}$ για την οποία ισχύει, για κάθε x:

$$x \notin L \Rightarrow f(x) < 2^{q(|x|)-1}$$

$$x \in L \Rightarrow f(x) \geq 2^{q(|x|)-1}$$

Επίσης, η σταθερή συνάρτηση 1 ανήκει στην $\#\mathbf{P}$ και η συνάρτηση $2^{q(|x|)-1}$ ανήκει στην $\#\mathbf{P}$ λόγω της μη ντετερμινιστικής μηχανής Turing που επιλέγει μη ντετερμινιστικά μία συμβολοακολουθία μήκους $q(|x|)$ και αποδέχεται αν και μόνο αν το τελευταίο bit είναι 0. Έχουμε υποθέσει ότι η $\#\mathbf{P}$ είναι κλειστή ως προς αφαίρεση, άρα $g \in \#\mathbf{P}$ όπου $g(x) = 2^{q(|x|)-1} - 1$, και $h \in \#\mathbf{P}$ όπου $h(x) = f(x) \ominus g(x)$. Παρατηρούμε ότι για την h ισχύουν, για κάθε x:

$$x \notin L \Rightarrow h(x) = 0$$

$$x \in L \Rightarrow h(x) \geq 1$$

Εφόσον $h \in \#\mathbf{P}$, υπάρχει μία μη ντετερμινιστική μηχανή Turing N' πολυωνυμικού χρόνου, που για κάθε είσοδο x, αν $x \in L$ έχει υπολογιστικό μονοπάτι που καταλήγει σε κατάσταση αποδοχής, ενώ αν $x \notin L$ όλα τα υπολογιστικά μονοπάτια της καταλήγουν σε κατάσταση απόρριψης. Άρα $L \in \mathbf{NP}$. Έχουμε λοιπόν ότι $\mathbf{PP} \subseteq \mathbf{NP}$, και επειδή η $\mathbf{PP}$ είναι κλειστή ως προς συμπλήρωμα [Sim75, Gil77] προκύπτει τελικά το ζητούμενο $\mathbf{PP} \subseteq \mathbf{coNP}$.

Έστω $L \in \mathbf{coNP}$. Αυτό σημαίνει ότι $\overline{L} \in \mathbf{NP}$, και η μη ντετερμινιστική μηχανή Turing πολυωνυμικού χρόνου που αποδέχεται την $\overline{L}$ ορίζει μία συνάρτηση $f \in \#\mathbf{P}$ για την οποία ισχύουν για κάθε x:

$$x \in L \Rightarrow f(x) = 0$$

$$x \notin L \Rightarrow f(x) \geq 1$$

Επίσης $g \in \#\mathbf{P}$, όπου $g(x) = 1$ για κάθε x, και αφού έχουμε υποθέσει ότι η $\#\mathbf{P}$ είναι κλειστή ως προς αφαίρεση τότε και η $h(x) = g(x) \ominus f(x)$ ανήκει στην $\#\mathbf{P}$. Όμως για την h ισχύουν, για κάθε x:

$$x \notin L \Rightarrow h(x) = 0$$

$$x \in L \Rightarrow h(x) = 1$$

Άρα υπάρχει μη ντετερμινιστική μηχανή Turing πολυωνυμικού χρόνου που αποδέχεται την L και για κάθε είσοδο έχει το πολύ ένα υπολογιστικό μονοπάτι που καταλήγει σε κατάσταση αποδοχής. Κατά συνέπεια $L \in \mathbf{UP}$. Το ζητούμενο $\mathbf{coNP} \subseteq \mathbf{UP}$ έπεται.

Ολοκληρώνοντας την απόδειξη του θεωρήματος, θα αποδείξουμε τη συνεπαγωγή (3) $\Rightarrow$ (2). Έστω ότι $\mathbf{UP} = \mathbf{PP}$ και ας είναι f, g αυθαίρετες συναρτήσεις της κλάσης $\#\mathbf{P}$ και $op : \mathbb{N} \times \mathbb{N} \to \mathbb{N}$ αυθαίρετη πολυωνυμικά υπολογιστή πράξη. Θα δείξουμε ότι $h \in \#\mathbf{P}$, όπου για κάθε x η h ορίζεται ως $h(x) = op(f(x), g(x))$.

Αρχικά, παρατηρούμε ότι τα σύνολα

$$B_f = \{\langle x, n\rangle : f(x) \geq n\}$$

$$B_g = \{\langle x, n\rangle : g(x) \geq n\}$$

ανήκουν στην κλάση $\mathbf{PP}$. Θα δείξουμε ότι $B_f \in \mathbf{PP}$. Η απόδειξη για το B_g είναι όμοια. Εφόσον $f \in \#\mathbf{P}$, υπάρχει μία μη ντετερμινιστική μηχανή Turing N_1 πολυωνυμικού χρόνου που για κάθε είσοδο x έχει ακριβώς $f(x)$ μονοπάτια που καταλήγουν σε κατάσταση αποδοχής, και έστω ότι το μήκος κάθε μονοπατιού είναι $q(|x|)$. Θεωρούμε τη μη ντετερμινιστική μηχανή Turing N που περιγράφεται από τον αλγόριθμο 18. Για κάθε είσοδο $\langle x, n\rangle$ το δέντρο υπολογισμού της N_1 χωρίζεται σε δύο υποδέντρα. Στο πρώτο εξομοιώνεται η λειτουργία της N_1 και δίνει $f(x)$ μονοπάτια που καταλήγουν σε κατάσταση αποδοχής, ενώ στο δεύτερο εξομοιώνεται η μηχανή Turing που αντιστοιχεί στην σταθερή $\#\mathbf{P}$ συνάρτηση n με εναλλαγμένες τις καταστάσεις αποδοχής και απόρριψης. Το ύψος του κάθε υποδέντρου είναι b. Έχουμε λοιπόν $f(x)$ μονοπάτια που αποδέχονται στο πρώτο υποδέντρο, $2^b - n$ μονοπάτια που αποδέχονται στο δεύτερο υποδέντρο, και συνολικά 2^{b+1} μονοπάτια στο δέντρο υπολογισμού της N με είσοδο $\langle x, n\rangle$. Αν $\langle x, n\rangle \in B_f$, τότε $f(x) \geq n$ και το πλήθος των μονοπατιών που αποδέχονται είναι $2^b + f(x) - n \geq 2^b$. Αντίθετα, αν $\langle x, n\rangle \notin B_f$ τότε ισχύει $f(x) < n$ και το πλήθος των μονοπατιών που αποδέχονται είναι μικρότερο από 2^b. Άρα $B_f \in \mathbf{PP}$.

Αλγόριθμος 18 Μη ντετερμινιστικός αλγόριθμος πολυωνυμικού χρόνου που αποδεικνύει ότι $\{\langle x, n\rangle : f(x) \geq n\} \in \mathbf{PP}$ όπου $f \in \#\mathbf{P}$, μέρος της απόδειξης του θεωρήματος 5.3.1

Είσοδος: $\langle x, n\rangle$

$z \leftarrow$ το μικρότερο i για το οποίο ισχύει $2^i \geq n$
$b \leftarrow \max\{q(|x|), i\}$
επέλεξε μη ντετερμινιστικά ένα $y \in \{0, 1\}$
if $y = 0$ **then**
    εξομοίωσε τη λειτουργία της N_1 με είσοδο x, επεκτείνοντας αν χρειάζεται κάθε μονοπάτι ώστε το δέντρο υπολογισμού να είναι πλήρες, δυαδικό, με ύψος b
else if $y = 1$ **then**
    επέλεξε μη ντετερμινιστικά ένα $w : 1 \leq w \leq 2^b$
    if $w \leq n$ **then**
        τερμάτισε την εκτέλεση σε κατάσταση απόρριψης
    else if $w > n$ **then**
        τερμάτισε την εκτέλεση σε κατάσταση αποδοχής
    end if
end if

Θεωρούμε επιπλέον το σύνολο

$$V = \left\{ \langle x, n_1, n_2 \rangle : \begin{array}{l} \langle x, n_1 \rangle \in B_f \wedge \langle x, n_1 + 1 \rangle \notin B_f \wedge \\ \langle x, n_2 \rangle \in B_g \wedge \langle x, n_2 + 1 \rangle \notin B_g \end{array} \right\}$$

που περιέχει τριάδες της μορφής $\langle x, f(x), g(x) \rangle$. Το V ανάγεται με αναγωγή φραγμένου πίνακα αληθείας στο $B_f \oplus B_g$, όπου το σύμβολο $\oplus$ συμβολίζει την ξένη ένωση[3] των συνόλων. Επειδή η κλάση **PP** είναι κλειστή ως προς (ξένη) ένωση και ως προς την αναγωγή φραγμένου πίνακα αληθείας [FR96], προκύπτει ότι $V \in \mathbf{PP}$. Έχουμε υποθέσει ότι $\mathbf{UP} = \mathbf{PP}$, άρα $V \in \mathbf{UP}$. Ας είναι M_1 η μη ντετερμινιστική μηχανή Turing πολυωνυμικού χρόνου που αποδέχεται το V και για κάθε είσοδο έχει το πολύ ένα υπολογιστικό μονοπάτι που καταλήγει σε κατάσταση αποδοχής. Επίσης, αφού οι f, g είναι $\#\mathbf{P}$ συναρτήσεις, υπάρχει ένα πολυώνυμο q ώστε για κάθε x να ισχύει $\max\{f(x), g(x)\} \leq 2^{q(|x|)}$. Η μη ντετερμινιστική μηχανή Turing M που περιγράφεται από τον αλγόριθμο 19 εξομοιώνει αρχικά τη λειτουργία της M_1 με είσοδο $\langle x, i, j \rangle$, για κάθε δυνατό ζεύγος (i, j). Επειδή υπάρχει ακριβώς ένα ζεύγος (i, j) για το οποίο ισχύει $i = f(x)$ και $j = g(x)$, ακριβώς ένας από αυτούς τους υπολογισμούς θα δώσει ακριβώς ένα μονοπάτι που αποδέχεται, ενώ όλα τα μονοπάτια των υπόλοιπων υπολογισμών θα καταλήγουν σε κατάσταση απόρριψης. Αυτό το μονοπάτι γεννά ακριβώς $op(i, j) = op(f(x), g(x)) = h(x)$ μονοπάτια που αποδέχονται, ενώ όλα τα άλλα μονοπάτια απορρίπτουν την είσοδο. Συνεπώς η M έχει, για κάθε είσοδο x, ακριβώς $h(x)$ μονοπάτια που αποδέχονται. Έπεται το ζητούμενο $h \in \#\mathbf{P}$.

■

Αποδείξαμε λοιπόν ότι η αφαίρεση είναι "πλήρης" πράξη για την κλάση $\#\mathbf{P}$, και επιπλέον δώσαμε μία κατάρρευση κλάσεων πολυπλοκότητας που χαρακτηρίζει πλήρως την κλειστότητα της $\#\mathbf{P}$ ως προς την αφαίρεση. Όπως φαίνεται από το ακόλουθο θεώρημα, που δίνουμε χωρίς απόδειξη, η συνθήκη αυτή συνεπάγεται την κατάρρευση πολλών άλλων κλάσεων, μεταξύ των οποίων

[3] Γενικά, αν $X, Y \subseteq \Sigma^\star$ τότε η ξένη ένωση των X,Y είναι το σύνολο $Z = X \oplus Y = \{0x : x \in X\} \cup \{1y : y \in Y\}$.

Αλγόριθμος 19 Μη ντετερμινιστικός αλγόριθμος πολυωνυμικού χρόνου που για κάθε είσοδο x παράγει ακριβώς $op(f(x), g(x))$ μονοπάτια που αποδέχονται ($f, g \in \#\mathbf{P}$ και op είναι αυθαίρετη πολυωνυμικά υπολογιστή πράξη), μέρος της απόδειξης του θεωρήματος 5.3.1

Είσοδος: x

επέλεξε μη ντετερμινιστικά ένα $i : 0 \leq i \leq 2^{q(|x|)}$
επέλεξε μη ντετερμινιστικά ένα $j : 0 \leq i \leq 2^{q(|x|)}$
εξομοίωσε τη λειτουργία της M_1 με είσοδο $\langle x, i, j \rangle$
if η M_1 τερματίζει την εκτέλεσή της σε κατάσταση απόρριψης **then**
    τερμάτισε την εκτέλεση σε κατάσταση απόρριψης
else if η M_1 τερματίζει την εκτέλεσή της σε κατάσταση αποδοχής **then**
    επέλεξε μη ντετερμινιστικά ένα $k : 1 \leq k \leq op(i, j)$
    τερμάτισε την εκτέλεση σε κατάσταση αποδοχής
end if

και της πολυωνυμικής ιεραρχίας. Επομένως το ενδεχόμενο η κλάση $\#\mathbf{P}$ να είναι κλειστή ως προς αφαίρεση θεωρείται εξαιρετικά απίθανο.

Θεώρημα 5.3.2 *Αν* $\mathbf{UP} = \mathbf{PP}$, *τότε:*

$$\mathbf{UP} = \mathbf{NP} = \mathbf{coNP} = \mathbf{PH} = \oplus\mathbf{P} = \mathbf{PP} = \mathbf{PP} \cup \mathbf{PP}^{\mathbf{PP}} \cup \mathbf{PP}^{\mathbf{PP}^{\mathbf{PP}}} \cup \ldots$$

Η αφαίρεση δεν είναι η μόνη πολυωνυμικά υπολογιστή πράξη με την ιδιότητα ότι αν η $\#\mathbf{P}$ είναι κλειστή ως προς την πράξη αυτή, τότε είναι κλειστή ως προς όλες τις πολυωνυμικά υπολογιστές πράξεις. Η ακέραια διαίρεση, που είναι ακόμη μία πράξη που ελαττώνει τους μάρτυρες, έχει επίσης αυτή την ιδιότητα. Την ακέραια διαίρεση μεταξύ δύο αριθμών a και b θα τη συμβολίζουμε με το σύμβολο $\oslash$: $a \oslash b = \lfloor a/b \rfloor$. Ο ορισμός 5.2.1 είναι ελαφρώς προβληματικός γιατί η ακέραια διαίρεση $a \oslash b$ δεν ορίζεται αν $b = 0$. Θα χρησιμοποιήσουμε λοιπόν τον ορισμό 5.3.3 για την κλειστότητα μίας κλάσης συναρτήσεων ως προς ακέραια διαίρεση.

Ορισμός 5.3.3 *Αν* $\mathcal{F}$ *είναι μία κλάση συναρτήσεων που απεικονίζουν φυσικούς σε φυσικούς, λέμε ότι η* $\mathcal{F}$ *είναι κλειστή ως προς ακέραια διαίρεση αν* *για κάθε* $f_1, f_2 \in \mathcal{F}$*, όπου* $(\forall n)[f_2(n) > 0]$*, ισχύει* $f_1 \oslash f_2 \in \mathcal{F}$*, όπου για κάθε* n *η* $f_1 \oslash f_2$ *ορίζεται ως εξής:* $(f_1 \oslash f_2)(n) = f_1(n) \oslash f_2(n)$.

Θεώρημα 5.3.4 *Οι ακόλουθες προτάσεις είναι ισοδύναμες:*

1. *Η* $\#\mathbf{P}$ *είναι κλειστή ως προς ακέραια διαίρεση.*

2. *Η* $\#\mathbf{P}$ *είναι κλειστή ως προς οποιαδήποτε πολυωνυμικά υπολογιστή πράξη.*

3. $\mathbf{UP} = \mathbf{PP}$.

Απόδειξη

Η συνεπαγωγή (2) $\Rightarrow$ (1) είναι προφανής, και το θεώρημα 5.3.1 εξασφαλίζει τη συνεπαγωγή (3) $\Rightarrow$ (2). Επομένως αρκεί να δείξουμε ότι (1) $\Rightarrow$ (3).

Έστω ότι η $\#\mathbf{P}$ είναι κλειστή ως προς ακέραια διαίρεση, και ας είναι L μία αυθαίρετη γλώσσα της κλάσης $\mathbf{PP}$ και N' μία μη ντετερμινιστική μηχανή Turing που αποδεικνύει ότι $L \in \mathbf{PP}$, σύμφωνα με τον ορισμό 5.1.1. Από την N' μπορούμε εύκολα να κατασκευάσουμε μία μηχανή Turing N που ικανοποιεί επίσης τον ορισμό 5.1.1 και έχει επιπλέον την ιδιότητα να υπάρχει ένας φυσικός αριθμός $k > 0$ ώστε να ισχύουν:

- Για κάθε είσοδο x, το δέντρο υπολογισμού της N έχει ακριβώς $2^{|x|^k}$ υπολογιστικά μονοπάτια, κάθε ένα από τα οποία έχει μήκος ακριβώς $|x|^k$.

- Για κάθε είσοδο x, $x \in L$ αν και μόνο αν ο υπολογισμός της N έχει τουλάχιστον $2^{|x|^k-1}$ μονοπάτια που καταλήγουν σε κατάσταση αποδοχής.

- Για κάθε είσοδο x, το δέντρο υπολογισμού της N έχει τουλάχιστον ένα μονοπάτι που καταλήγει σε κατάσταση απόρριψης.

Η N ορίζει ακόμη την $\#\mathbf{P}$ συνάρτηση f. Επιπλέον, η συνάρτηση g που για κάθε x ορίζεται ως $g(x) = 2^{|x|^k-1}$ ανήκει στην $\#\mathbf{P}$, και επειδή έχουμε θεωρήσει ότι η $\#\mathbf{P}$ είναι κλειστή ως προς ακέραια διαίρεση πρέπει να περιέχει και την h, όπου $h(x) = f(x) \oslash g(x)$. Παρατηρούμε ότι αν $x \in L$, $h(x) = f(x) \oslash g(x) = 1$, ενώ αν $x \notin L$ ισχύει $h(x) = f(x) \oslash g(x) = 0$. Δηλαδή η μηχανή Turing που ορίζει την #**P** συνάρτηση h αποδέχεται την L και για κάθε είσοδο έχει το πολύ ένα μονοπάτι που καταλήγει σε κατάσταση αποδοχής. Άρα $L \in$ **UP**.

■

Στην απόδειξη του θεωρήματος 5.3.4 φαίνεται καθαρά η τεχνική της ελάττωσης μαρτύρων που χρησιμοποιούμε για την απόδειξη αποτελεσμάτων σε αυτό το κεφάλαιο. Ο στόχος είναι να δείξουμε ότι αν κάποια κλάση συναρτήσεων, όπως η #**P**, είναι κλειστή ως προς μία πράξη που μειώνει το πλήθος των μαρτύρων τότε κάποιες κλάσεις πολυπλοκότητας καταρρέουν. Κάθε μη ντετερμινιστική μηχανή Turing πολυωνυμικού χρόνου ορίζει ταυτόχρονα μία γλώσσα της κλάσης **NP** (αυτήν που αποδέχεται) και μία συνάρτηση της κλάσης #**P** (τη συνάρτηση που μετράει το πλήθος των μαρτύρων της εισόδου). Εκμεταλλευόμενοι την ιδιότητα κλειστότητας που υποθέτουμε ότι ισχύει, προσπαθούμε να φέρουμε την #**P** συνάρτηση σε μία μορφή που μας βολεύει, έτσι ώστε η μηχανή Turing που ορίζει τη νέα συνάρτηση να αποδέχεται την ίδια γλώσσα αλλά με λιγότερους μάρτυρες.

Το γενικό αυτό αποδεικτικό σχήμα εφαρμόζεται άμεσα στην απόδειξη του θεωρήματος 5.3.4. Το ίδιο σχήμα εφαρμόζεται και στην απόδειξη του θεωρήματος 5.3.1, όμως εκεί χρησιμοποιείται δύο φορές για την απόδειξη δύο διαφορετικών εγκλεισμών κλάσεων πολυπλοκότητας.

5.4 Ενδιάμεσες ιδιότητες κλειστότητας της #P

Είναι ενδιαφέρον να παρατηρήσει κανείς ότι η κλειστότητα της #**P** ως προς κάποια "πλήρη" πράξη (αφαίρεση ή διαίρεση) χαρακτηρίζεται πλήρως και από κάποια κατάρρευση κλάσεων πολυπλοκότητας. Αντίθετα, η κλειστότητα ως προς τις ενδιάμεσες πράξεις δεν χαρακτηρίζεται πλήρως από κάποια τέτοια κατάρρευση. Σε αυτήν την ενότητα θα χρησιμοποιήσουμε και πάλι την ίδια τεχνική της ελάττωσης μαρτύρων για να φτάσουμε σε αναγκαίες συνθήκες για την κλειστότητα της #**P** ως προς διάφορες ενδιάμεσες πράξεις. Φυσικά, λόγω του θεωρήματος 5.3.1, ικανή συνθήκη για την κλειστότητα της $\#\mathbf{P}$ ως προς οποιαδήποτε πολυωνυμικά υπολογιστή πράξη είναι η $\mathbf{UP} = \mathbf{PP}$.

Οι πράξεις με τις οποίες θα ασχοληθούμε είναι η εύρεση ελαχίστου και μεγίστου, η ελάττωση κατά 1 και η ακέραια διαίρεση με το 2. Στη συνέχεια ορίζουμε τι ακριβώς είναι η κλειστότητα της $\#\mathbf{P}$ ως προς αυτές τις πράξεις. Για την περίπτωση του ελαχίστου και του μεγίστου ο ορισμός είναι ειδική περίπτωση του ορισμού 5.2.1.

Ορισμός 5.4.1 *1. Η κλάση $\#\mathbf{P}$ είναι κλειστή ως προς εύρεση ελαχίστου, αν για κάθε $f, g \in \#\mathbf{P}$ ισχύει $h \in \#\mathbf{P}$, όπου h είναι η συνάρτηση που για κάθε x ορίζεται ως $h(x) = \min\{f(x), g(x)\}$.*

2. Η κλάση $\#\mathbf{P}$ είναι κλειστή ως προς εύρεση μεγίστου, αν για κάθε $f, g \in \#\mathbf{P}$ ισχύει $h \in \#\mathbf{P}$, όπου h είναι η συνάρτηση που για κάθε x ορίζεται ως $h(x) = \max\{f(x), g(x)\}$.

3. Η κλάση $\#\mathbf{P}$ είναι κλειστή ως προς ελάττωση κατά 1, αν για κάθε $f \in \#\mathbf{P}$ ισχύει $h \in \#\mathbf{P}$, όπου h είναι η συνάρτηση που για κάθε x ορίζεται ως $h(x) = f(x) \ominus 1$.

4. Η κλάση $\#\mathbf{P}$ είναι κλειστή ως προς ακέραια διαίρεση με το 2, αν για κάθε $f \in \#\mathbf{P}$ ισχύει $h \in \#\mathbf{P}$, όπου h είναι η συνάρτηση που για κάθε x ορίζεται ως $h(x) = f(x) \oslash 2$.

Θεώρημα 5.4.2 *1. Αν η $\#\mathbf{P}$ είναι κλειστή ως προς ελάττωση κατά 1 τότε $\mathbf{coNP} \subseteq \mathbf{SPP}$ (ισοδύναμα, $\mathbf{NP} \subseteq \mathbf{SPP}$).*

2. Αν $\mathbf{UP} = \mathbf{NP}$ τότε η $\#\mathbf{P}$ είναι κλειστή ως προς ελάττωση κατά 1.

Απόδειξη

Για το (1), έστω ότι η $\#\mathbf{P}$ είναι κλειστή ως προς ελάττωση κατά 1, και ας είναι L μία αυθαίρετη γλώσσα της $\mathbf{NP}$ και N μία μη ντετερμινιστική μηχανή Turing πολυωνυμικού χρόνου που αποφασίζει την L. Έστω $f \in \#\mathbf{P}$ η συνάρτηση που ορίζεται από το πλήθος των μονοπατιών της N που καταλήγουν σε κατάσταση αποδοχής. Εφόσον υποθέτουμε ότι η $\#\mathbf{P}$ είναι κλειστή ως προς ελάττωση κατά 1, και η $g(x) = f(x) \ominus 1$ θα ανήκει στην #**P**. Άρα υπάρχει μία μη ντετερμινιστική μηχανή Turing N' πολυωνυμικού χρόνου και ένα πολυώνυμο p, ώστε για κάθε είσοδο x ο υπολογισμός της N' να έχει ακριβώς $2^{p(|x|)}$ υπολογιστικά μονοπάτια μήκους $p(|x|)$ από τα οποία ακριβώς $g(x)$ καταλήγουν σε κατάσταση αποδοχής. Αν εναλλάξουμε τις καταστάσεις αποδοχής και απόρριψης της N', προκύπτει μία μη ντετερμινιστική μηχανή Turing N'' πολυωνυμικού χρόνου, που για κάθε είσοδο x έχει $2^{p(|x|)} - g(x)$ υπολογιστικά μονοπάτια που καταλήγουν σε κατάσταση αποδοχής. Άρα η $2^{p(|x|)} - g(x)$ ανήκει στην κλάση #**P**. Λόγω του θεωρήματος 5.2.2, και η $f(x) + (2^{p(|x|)} - g(x))$ ανήκει στην #**P**. Παρατηρούμε όμως ότι η τελευταία συνάρτηση παίρνει τιμή $2^{p(|x|)}$ αν $x \notin L$, ενώ παίρνει τιμή $2^{p(|x|)} + 1$ αν $x \in L$. Άρα η μηχανή Turing που την ορίζει αποδεικνύει ότι $L \in \mathbf{SPP}$. Επομένως $\mathbf{NP} \subseteq \mathbf{SPP}$, και επειδή η **SPP** είναι κλειστή ως προς συμπλήρωμα αυτό ισοδυναμεί με $\mathbf{coNP} \subseteq \mathbf{SPP}$.

Για το (2), έστω ότι $\mathbf{UP} = \mathbf{NP}$. Αν $f \in \#\mathbf{P}$ είναι μία αυθαίρετη συνάρτηση και N είναι η αντίστοιχη μη ντετερμινιστική μηχανή Turing πολυωνυμικού χρόνου, θεωρούμε το ακόλουθο σύνολο:

$$B = \left\{ \langle x, w \rangle : \begin{array}{c} w \text{ είναι μονοπάτι του υπολογισμού της } N \text{ με είσοδο } x \\ \text{που καταλήγει σε αποδοχή και υπάρχει λεξικογραφικά} \\ \text{μεγαλύτερο μονοπάτι που καταλήγει σε αποδοχή} \end{array} \right\}$$

$B \in \mathbf{NP}$ μέσω της μηχανής Turing που εξομοιώνει την N με είσοδο x και αποδέχεται αν και μόνο αν το w είναι υπολογιστικό μονοπάτι της N με είσοδο x που καταλήγει σε κατάσταση αποδοχής και το τρέχον μονοπάτι της εξομοίωσης καταλήγει σε κατάσταση αποδοχής και είναι λεξικογραφικά μεγαλύτερο από το w. Εφόσον έχουμε υποθέσει ότι $\mathbf{UP} = \mathbf{NP}$, $B \in \mathbf{UP}$. Υπάρχει λοιπόν μία μη ντετερμινιστική μηχανή Turing N' πολυωνυμικού χρόνου που αποφασίζει το B και για κάθε είσοδο έχει το πολύ ένα υπολογιστικό μονοπάτι που καταλήγει σε κατάσταση αποδοχής. Θεωρούμε τώρα τη μη ντετερμινιστική μηχανή Turing που, με είσοδο x, επιλέγει μη ντετερμινιστικά ένα μονοπάτι w του υπολογισμού της N με είσοδο x και στη συνέχεια εξομοιώνει τη λειτουργία της N' με είσοδο $\langle x, w \rangle$. Η μηχανή αυτή λειτουργεί σε πολυωνυμικό χρόνο και για κάθε μονοπάτι της N που αποδέχεται, εκτός από το λεξικογραφικά μεγαλύτερο, προκύπτει μονοπάτι που αποδέχεται ενώ όλα τα υπόλοιπα απορρίπτουν. Δηλαδή για κάθε είσοδο x το πλήθος των μονοπατιών που καταλήγουν σε κατάσταση αποδοχής είναι $f(x) \ominus 1$. Άρα η $\#\mathbf{P}$ είναι κλειστή ως προς ελάττωση κατά 1.

■

Με αυτό το θεώρημα δεν χαρακτηρίσαμε πλήρως την κλειστότητα της $\#\mathbf{P}$ ως προς ελάττωση κατά 1 με κάποια κατάρρευση κλάσεων πολυπλοκότητας, όμως δώσαμε μία ικανή και μία αναγκαία συνθήκη για αυτήν. Ο πλήρης χαρακτηρισμός με μία ικανή και αναγκαία συνθήκη παραμένει ανοικτό πρόβλημα. Παρατηρούμε όμως ότι η ικανή συνθήκη ($\mathbf{NP} \subseteq \mathbf{UP}$) και η αναγκαία συνθήκη ($\mathbf{NP} \subseteq \mathbf{SPP}$) μοιάζουν πολύ μεταξύ τους, αφού η κλάση $\mathbf{SPP}$ μπορεί να θεωρήθει ως μία γενίκευση της $\mathbf{UP}$.

Θεώρημα 5.4.3 *Αν η $\#\mathbf{P}$ είναι κλειστή ως προς ακέραια διαίρεση με το 2 τότε $\oplus\mathbf{P} = \mathbf{SPP}$.*

Απόδειξη

Έστω ότι η $\#\mathbf{P}$ είναι κλειστή ως προς ακέραια διαίρεση με το 2, και έστω μία γλώσσα $L \in \oplus\mathbf{P}$. Υπάρχει λοιπόν μία μη ντετερμινιστική μηχανή Turing N πολυωνυμικού χρόνου και ένα πολυώνυμο p, ώστε για κάθε είσοδο x η N να έχει ακριβώς $2^{p(|x|)}$ υπολογιστικά μονοπάτια μήκους $p(|x|)$ και $x \in L$ αν και μόνο αν το πλήθος των μονοπατιών που οδηγούν σε κατάσταση αποδοχής είναι περιττός αριθμός. Έστω ότι για κάθε είσοδο x, ο υπολογισμός της N έχει $f(x)$ μονοπάτια που αποδέχονται. Προφανώς $f \in \#\mathbf{P}$. Έχουμε υποθέσει ότι η $\#\mathbf{P}$ είναι κλειστή ως προς ακέραια διαίρεση με το 2 και γνωρίζουμε από το θεώρημα 5.2.3 ότι είναι κλειστή ως προς πολλαπλασιασμό. Άρα και η συνάρτηση $g(x) = 2(f(x) \oslash 2)$ ανήκει στην $\#\mathbf{P}$.

Ακόμη, με εναλλαγή των καταστάσεων αποδοχής και απόρριψης της N προκύπτει ότι και η συνάρτηση $f'(x) = 2^{p(|x|)} - f(x)$ ανήκει στην $\#\mathbf{P}$. Επειδή η $\#\mathbf{P}$ είναι κλειστή ως προς πρόσθεση (θεώρημα 5.2.2), και η συνάρτηση $f'(x) + g(x) = 2^{p(|x|)} - f(x) + 2(f(x) \oslash 2)$ ανήκει στην $\#\mathbf{P}$. Όμως η συνάρτηση αυτή έχει τιμή $2^{p(|x|)} - 1$ αν $x \in L$, ενώ έχει τιμή $2^{p(|x|)}$ αν $x \notin L$. Συνεπώς $\overline{L} \in$ **SPP**, λόγω της μη ντετερμινιστικής μηχανής Turing πολυωνυμικού χρόνου που αντιστοιχεί στην #**P** συνάρτηση $f' + g$. Επειδή η **SPP** είναι κλειστή ως προς συμπλήρωμα, έχουμε $L \in$ **SPP**, άρα $\oplus$**P** $\subseteq$ **SPP**.

Αρκεί να δείξουμε ότι **SPP** $\subseteq$ $\oplus$**P**. Αυτό ισχύει χωρίς προϋποθέσεις. Έστω $L \in$ **SPP**. Από τον ορισμό 5.1.1, υπάρχει $A \in$ **P**, $f \in$ **FP** και πολυώνυμο q ώστε για κάθε $x \in L$ να ισχύει $|\{y : |y| = q(|x|) \wedge \langle x, y\rangle \in A\}| = f(x)$ και για κάθε $x \notin L$ να ισχύει $|\{y : |y| = q(|x|) \wedge \langle x, y\rangle \in A\}| = f(x) - 1$. Ο αλγόριθμος 20 περιγράφει μία μη ντετερμινστική μηχανή Turing πολυωνυμικού χρόνου που για κάθε είσοδο x έχει περιττό αριθμό μονοπατιών που την αποδέχονται αν και μόνο αν $x \in L$. Άρα $L \in \oplus$**P**.

■

Θεώρημα 5.4.4 *Μία γλώσσα L ανήκει στην κλάση $\mathbf{C_=P}$ αν και μόνο αν υπάρχει μία γλώσσα $B \in$ **P** και ένα πολυώνυμο r ώστε για κάθε x να ισχύουν:*

- $x \in L \Rightarrow |\{y : |y| = r(|x|) \wedge \langle x, y\rangle \in B\}| = 2^{r(|x|)-2}$
- $x \notin L \Rightarrow |\{y : |y| = r(|x|) \wedge \langle x, y\rangle \in B\}| < 2^{r(|x|)-2}$

Απόδειξη

Έστω $L \in \mathbf{C_=P}$. Από τον ορισμό 5.1.2, υπάρχει μία γλώσσα $A \in$ **P** και ένα πολυώνυμο q ώστε για κάθε x να ισχύει:

$$x \in L \Leftrightarrow |\{y : |y| = q(|x|) \wedge \langle x, y\rangle \in A\}| = 2^{q(|x|)-1}$$

Θεωρούμε το πολυώνυμο $r(n) = 2q(n)$ και τη γλώσσα B που ορίζεται ως εξής:

$$B = \{\langle x, z\rangle : (\exists w_1, w_2 : |w_1| = |w_2| = q(|x|)) \\ [w_1 w_2 = z \wedge \langle x, w_1\rangle \in A \wedge \langle x, w_2\rangle \notin A]\}$$

Φαίνεται εύκολα ότι $B \in$ **P**. Συμβολίζουμε με $f(x)$ το πλήθος των μαρτύρων του x ως προς A,q και με $h(x)$ το πλήθος των μαρτύρων του x ως προς B,r:

$$f(x) = |\{y : |y| = q(|x|) \wedge \langle x, y\rangle \in A\}|$$

Αλγόριθμος 20 Μη ντετερμινιστικός αλγόριθμος πολυωνυμικού χρόνου που αποδεικνύει ότι $\mathbf{SPP} \subseteq \oplus\mathbf{P}$, μέρος της απόδειξης του θεωρήματος 5.4.3

Είσοδος: x

```
z ← f(x)
επέλεξε μη ντετερμινιστικά μία συμβολοακολουθία w με μήκος ακριβώς
    q(|x|)
if ο z είναι περιττός ή w ≻ 0^{q(|x|)} then
    if ⟨x, w⟩ ∈ A then
        τερμάτισε την εκτέλεση σε κατάσταση αποδοχής
    else
        τερμάτισε την εκτέλεση σε κατάσταση απόρριψης
    end if
else
    if ⟨x, w⟩ ∈ A then
        τερμάτισε την εκτέλεση σε κατάσταση απόρριψης
    else
        τερμάτισε την εκτέλεση σε κατάσταση αποδοχής
    end if
end if
```

$$h(x) = |\{y : |y| = r(|x|) \wedge \langle x, y \rangle \in B\}|$$

Από τον ορισμό του συνόλου B και του πολυωνύμου p προκύπτει ότι $h(x) = f(x)(2^{q(|x|)} - f(x))$. Αν $x \in L$, τότε ισχύει $f(x) = 2^{q(|x|)-1}$ και $h(x) = 2^{2q(|x|)-2} = 2^{r(|x|)-2}$. Επειδή η $h(x)$ μεγιστοποιείται όταν $f(x) = 2^{q(|x|)-1}$, και $f(x) = 2^{q(|x|)-1}$ αν και μόνο αν $x \in L$, προκύπτει ότι για κάθε $x \notin L$ ισχύει $h(x) < 2^{r(|x|)-2}$. Επομένως τα B,r έχουν τις ιδιότητες που ζητάμε.

Αντίστροφα, αν για μία γλώσσα L υπάρχει $B \in \mathbf{P}$ και πολυώνυμο r που ικανοποιούν τις συνθήκες του θεωρήματος τότε, επιλέγοντας απλά $q(n) = r(n) - 1$, το πολυώνυμο q και η γλώσσα A ικανοποιούν τον ορισμό 5.1.2 για την κλάση $\mathbf{C_{=}P}$. Άρα $L \in \mathbf{C_{=}P}$.

■

Θεώρημα 5.4.5 *1. Αν η #**P** είναι κλειστή ως προς εύρεση ελαχίστου τότε* $\mathbf{NP} = \mathbf{UP}$.

*2. Αν η #**P** είναι κλειστή ως προς εύρεση ελαχίστου τότε* $\mathbf{C_{=}P} = \mathbf{SPP}$.

*3. Αν η #**P** είναι κλειστή ως προς εύρεση μεγίστου τότε* $\mathbf{C_{=}P} = \mathbf{SPP}$.

Απόδειξη

Για το (1), έστω $L \in \mathbf{NP}$ και $f \in \#\mathbf{P}$ η συνάρτηση που μετράει το πλήθος των μαρτύρων της L. Εφόσον υποθέτουμε ότι η #**P** είναι κλειστή ως προς εύρεση ελαχίστου, η συνάρτηση $h(x) = \min\{f(x), 1\}$ ανήκει και αυτή στην #**P**. Ο υπολογισμός της μη ντετερμινιστικής μηχανής Turing πολυωνυμικού χρόνου που αντιστοιχεί στην h έχει ακριβώς ένα μονοπάτι που καταλήγει σε κατάσταση αποδοχής αν $x \in L$, ενώ όλα τα μονοπάτια καταλήγουν σε κατάσταση απόρριψης αν $x \notin L$. Κατά συνέπεια $L \in \mathbf{UP}$. Επειδή ως γνωστόν ισχύει $\mathbf{UP} \subseteq \mathbf{NP}$, προκύπτει τελικά $\mathbf{NP} = \mathbf{UP}$.

Για το (2), έστω ότι $L \in \mathbf{C_{=}P}$, και ας είναι $B \in \mathbf{P}$ η γλώσσα και r το πολυώνυμο που προκύπτουν από το θεώρημα 5.4.4. Επειδή $\overline{B} \in \mathbf{P}$, υπάρχει μία μη ντετερμινιστική μηχανή Turing πολυωνυμικού χρόνου που για κάθε είσοδο x επιλέγει μη ντετερμινιστικά μία συμβολοακολουθία w μήκους ακριβώς $r(|x|)$

και αποδέχεται αν και μόνο αν $\langle x, w\rangle \in \overline{B}$. Για τη συνάρτηση $f \in \#\mathbf{P}$ που αντιστοιχεί σε αυτή τη μηχανή, ισχύουν, για κάθε x:

- $x \in L \Rightarrow f(x) = \frac{3}{4}2^{r(|x|)}$
- $x \notin L \Rightarrow f(x) > \frac{3}{4}2^{r(|x|)}$

Επιπλέον, οι συναρτήσεις $g(x) = 1 + \frac{3}{4}2^{r(|x|)}$ και $h(x) = \min\{f(x), g(x)\}$ ανήκουν στην κλάση $\#\mathbf{P}$ αφού έχουμε υποθέσει ότι η $\#\mathbf{P}$ είναι κλειστή ως προς εύρεση ελαχίστου και ξέρουμε ότι είναι κλειστή ως προς πρόσθεση και πολλαπλασιασμό. Για τη συνάρτηση h ισχύουν, για κάθε x:

- $x \in L \Rightarrow h(x) = \frac{3}{4}2^{r(|x|)}$
- $x \notin L \Rightarrow h(x) = 1 + \frac{3}{4}2^{r(|x|)}$

Επομένως $\overline{L} \in \mathbf{SPP}$, και επειδή η $\mathbf{SPP}$ είναι κλειστή ως προς συμπλήρωμα προκύπτει τελικά $L \in \mathbf{SPP}$. Άρα $\mathbf{C_=P} \subseteq \mathbf{SPP}$, από το οποίο προκύπτει ότι $\mathbf{C_=P} = \mathbf{SPP}$.

Για το (3), έστω $L \in \mathbf{C_=P}$. Από το θεώρημα 5.4.4, υπάρχει ένα πολυώνυμο r και μία $f \in \#\mathbf{P}$ για την οποία ισχύουν, για καθε x:

- $x \in L \Rightarrow f(x) = 2^{r(|x|)-2}$
- $x \notin L \Rightarrow f(x) < 2^{r(|x|)-2}$

Επιπλέον, η συνάρτηση $g(x) = 2^{r(|x|)-2} - 1$ ανήκει στην κλάση $\#\mathbf{P}$. Αφού υποθέτουμε ότι η $\#\mathbf{P}$ είναι κλειστή ως προς εύρεση μεγίστου, και η $h(x) = \max\{f(x), g(x)\}$ ανήκει στην $\#\mathbf{P}$. Όμως αν $x \in L$ έχουμε $h(x) = 2^{r(|x|)-2}$, ενώ αν $x \notin L$ έχουμε $h(x) = 2^{r(|x|)-2} - 1$. Άρα $L \in \mathbf{SPP}$, και $\mathbf{C_=P} = \mathbf{SPP}$.

■

5.5 Κλειστότητα της OptP ως προς αφαίρεση

Για την κλάση **OptP** τίθεται, όπως και για την $\#\mathbf{P}$, το ερώτημα κατά πόσο είναι κλειστή ως προς διάφορες πολυωνυμικά υπολογιστές πράξεις. Φαίνεται εύκολα ότι είναι κλειστή ως προς πρόσθεση και πολλαπλασιασμό, όμως δεν ισχύει το ίδιο και για την αφαίρεση. Μάλιστα ισχύει για την κλάση **OptP** ένα θεώρημα αντίστοιχο του θεωρήματος 5.3.1, δηλαδή η αφαίρεση είναι "πλήρης" πράξη και για την **OptP**.

Θεώρημα 5.5.1 *Οι ακόλουθες προτάσεις είναι ισοδύναμες:*

1. *Η* **OptP** *είναι κλειστή ως προς αφαίρεση.*
2. *Η* **OptP** *είναι κλειστή ως προς οποιαδήποτε πολυωνυμικά υπολογιστή πράξη.*
3. **NP** = **coNP**.

Απόδειξη

Η συνεπαγωγή (2) $\Rightarrow$ (1) ισχύει, αφού η αφαίρεση είναι πολυωνυμικά υπολογιστή πράξη.

Για τη συνεπαγωγή (3) $\Rightarrow$ (2), υποθέτουμε ότι $\mathbf{NP} = \mathbf{coNP}$. Έστω $f, g \in \mathbf{OptP}$ και N_f, N_g οι αντίστοιχες μηχανές Turing πολυωνυμικού χρόνου. Δηλαδή για κάθε είσοδο x η μέγιστη τιμή που προκύπτει ως έξοδος από την N_f σε κάποιο υπολογιστικό μονοπάτι είναι $f(x)$, και η μέγιστη τιμή που προκύπτει ως έξοδος από την N_g σε κάποιο υπολογιστικό μονοπάτι είναι $g(x)$. Θεωρούμε ακόμη τα σύνολα:

$$L_f = \{\langle x, i \rangle : f(x) > i\}$$

$$L_g = \{\langle x, i \rangle : g(x) > i\}$$

Μία μη ντετερμινιστική μηχανή Turing που αποφασίζει το L_f, σε είσοδο $\langle x, i \rangle$ εξομοιώνει την N_f με είσοδο x και σε κάθε μονοπάτι της εξομοίωσης αποδέχεται αν και μόνο αν η έξοδος της N_f είναι μεγαλύτερη από i. Επομένως $L_f \in \mathbf{NP}$ και, ομοίως, $L_g \in \mathbf{NP}$. Έχουμε υποθέσει ότι $\mathbf{NP} = \mathbf{coNP}$, άρα $\overline{L_f}, \overline{L_g} \in \mathbf{NP}$ και έστω N_1, N_2 οι μη ντετερμινιστικές μηχανές Turing πολυωνυμικού χρόνου που αποφασίζουν τις γλώσσες $\overline{L_f}$ και $\overline{L_g}$, αντίστοιχα.

Αν $op : \mathbb{N} \times \mathbb{N} \rightarrow \mathbb{N}$ είναι μία πολυωνυμικά υπολογιστή πράξη, ο αλγόριθμος 21 περιγράφει μία μη ντετερμινιστική μηχανή Turing πολυωνυμικού χρόνου

Αλγόριθμος 21 Μη ντετερμινιστικός αλγόριθμος πολυωνυμικού χρόνου που για κάθε είσοδο x παράγει μονοπάτια των οποίων η μέγιστη τιμή εξόδου είναι ακριβώς $op(f(x), g(x))$ ($f, g \in$ **OptP** και op είναι αυθαίρετη πολυωνυμικά υπολογιστή πράξη), μέρος της απόδειξης του θεωρήματος 5.5.1

Είσοδος: x

1: εξομοίωσε τη λειτουργία της N_f με είσοδο x
2: $z_1 \leftarrow$ η έξοδος του τρέχοντος υπολογιστικού μονοπατιού της N_f
3: εξομοίωσε τη λειτουργία της N_g με είσοδο x
4: $z_2 \leftarrow$ η έξοδος του τρέχοντος υπολογιστικού μονοπατιού της N_g
5: εξομοίωσε τη λειτουργία της N_1 με είσοδο $\langle x, z_1 \rangle$
6: εξομοίωσε τη λειτουργία της N_2 με είσοδο $\langle x, z_2 \rangle$
7: **if** η N_1 και η N_2 τερμάτισαν την εκτέλεσή τους σε κατάσταση αποδοχής **then**
8:     **output** $op(z_1, z_2)$
9: **else**
10:     **output** 0
11: **end if**

που αποδεικνύει ότι η συνάρτηση $op(f(x), g(x))$ ανήκει στην κλάση **OptP**. Στη γραμμή 21.7 εξασφαλίζεται ότι $f(x) \leq z_1$ και $g(x) \leq z_2$, όμως επειδή η $f(x)$ είναι η μέγιστη τιμή των εξόδων των υπολογιστικών μονοπατιών της N_f και αντίστοιχα η $g(x)$ είναι η μέγιστη τιμή των υπολογιστικών μονοπατιών της N_g οι συνθήκες αυτές ισοδυναμούν με $z_1 = f(x)$ και $z_2 = g(x)$. Άρα στα μονοπάτια αυτά η έξοδος της μηχανής είναι $op(f(x), g(x))$. Σε όλα τα άλλα μονοπάτια η έξοδος είναι 0. Συνεπώς για κάθε είσοδο x η μέγιστη τιμή εξόδου της μηχανής είναι $op(f(x), g(x))$, αφού πάντα θα υπάρχει τουλάχιστον ένα ζεύγος μονοπατιών που ικανοποιούν τη συνθήκη της γραμμής 21.7.

Τέλος, αποδεικνύουμε τη συνεπαγωγή (1) $\Rightarrow$ (3). Έστω ότι η **OptP** είναι κλειστή ως προς αφαίρεση και ας είναι L μία αυθαίρετη γλώσσα της κλασης **NP** και N η μη ντετερμινιστική μηχανή Turing που την αποφασίζει. Παρατηρούμε ότι μία μη ντετερμινιστική μηχανή Turing που εξομοιώνει την N και δίνει έξοδο 1 αν το μονοπάτι καταλήγει σε κατάσταση αποδοχής ή έξοδο 0 αν το μονοπάτι καταλήγει σε κατάσταση απόρριψης αποδεικνύει ότι η χαρακτηριστική συνάρτηση της L ανήκει στην κλάση **OptP**. Συνεπώς υπάρχει $f \in$ **OptP** ώστε για κάθε $x \in L$, $f(x) = 1$ και για κάθε $x \notin L$, $f(x) = 0$. Η σταθερή συνάρτηση $g(x) = 1$ ανήκει επίσης στην **OptP**. Θεωρούμε τη συνάρτηση $h(x) = g(x) \ominus f(x)$. Εφόσον έχουμε υποθέσει ότι η **OptP** είναι κλειστή ως προς αφαίρεση, $h \in$ **OptP**. Η h παίρνει τιμή $h(x) = 0$ αν $x \in L$, ενώ παίρνει τιμή $h(x) = 1$ αν $x \notin L$. Άρα υπάρχει μία μη ντετερμινιστική μηχανή Turing N' πολυωνυμικού χρόνου που αν η είσοδός της ανήκει στην L όλα τα μονοπάτια έχουν έξοδο 0, ενώ αν η είσοδος δεν ανήκει στην L υπάρχει τουλάχιστον ένα μονοπάτι που έχει έξοδο 1 και όλα τα υπόλοιπα έχουν έξοδο 0. Θεωρούμε τη μη ντετερμινιστική μηχανή Turing που για είσοδο x εξομοιώνει τη λειτουργία της N' με είσοδο x και κάθε μονοπάτι αποδέχεται αν και μόνο αν η έξοδος της N' στο μονοπάτι αυτό είναι 0. Αν $x \in L$ όλα τα μονοπάτια αποδέχονται, ενώ αν $x \notin L$ υπάρχει μονοπάτι που απορρίπτει. Συνεπώς $L \in$ **coNP**. Έπεται ότι **NP** $\subseteq$ **coNP**, άρα **NP** $=$ **coNP**.

$\blacksquare$

Βιβλιογραφία

[Ζάχ03a] Σ. Ζάχος. Αλγόριθμοι και Πολυπλοκότητα: Σημειώσεις. 2003.

[Ζάχ03b] Σ. Ζάχος. Στοιχεία Θεωρίας Αριθμών και Εφαρμογές στην Κρυπτογραφία: Σημειώσεις. 2003.

[Ζάχ04a] Σ. Ζάχος. Υπολογισιμότητα και Πολυπλοκότητα: Σημειώσεις. 2004.

[Τζά04b] Α. Τζάννες. Ο Ισομορφισμός Γράφων και κλάσεις πολυπλοκότητας. Διπλωματική Εργασία DT2004-0143, Εθνικό Μετσόβιο Πολυτεχνείο, 2004.

[AFK89] M. Abadi, J. Feigenbaum, and J. Kilian. On hiding information from an oracle. *Journal of Computer and System Sciences*, 39(1):21–50, 1989.

[AR88] E. Allender and R. Rubinstein. P-printable sets. *SIAM Journal on Computing*, 17(6):1193–1202, 1988.

[BC94] D. P. Bovet and P. Crescenzi. *Introduction to the Theory of Complexity*. Prentice Hall International (UK) Limited, Campus 400, Maylands Avenue, Hemel Hempstead, Hertfordshire, HP2 7EZ, 1994.

[BDG88] J. L. Balcázar, J. Díaz, and J. Gabarró. *Structural Complexity I*, volume 11 of *Monographs on Theoretical Computer Science (EATCS)*. Springer-Verlag Berlin Heidelberg New York, 1988.

[BDG90] J. L. Balcázar, J. Díaz, and J. Gabarró. *Structural Complexity II*, volume 22 of *Monographs on Theoretical Computer Science (EATCS)*. Springer-Verlag Berlin Heidelberg New York, 1990.

[Bei89] R. Beigel. On the relativized power of additional accepting paths. In *Proceedings of the 4th Structure in Complexity Theory Conference*, pages 216–224. IEEE Computer Society Press, June 1989.

[Ber77] L. Berman. *Polynomial Reducibilities and Complete Sets*. PhD thesis, Cornell University, Ithaca, NY, 1977.

[Ber78] P. Berman. Relationship between density and deterministic complexity of NP-complete languages. In *5th International Colloquium on Automata, Languages, and Programming*, volume 62 of *Lecture Notes in Computer Science*, pages 63–71. Springer-Verlag, July 1978.

[BG82] A. Blass and Y. Gurevich. On the unique satisfiability problem. *Information and Control*, 55(1–3):80–88, 1982.

[BLS84] R. Book, T. Long, and A. Selman. Quantitative relativizations of complexity classes. *SIAM Journal on Computing*, 13(3):461–487, 1984.

[BLS85] R. Book, T. Long, and A. Selman. Quantitative relativizations of complexity classes. *Journal of Computer and System Sciences*, 30(3):395–413, 1985.

[CH90] J. Cai and L. Hemachandra. On the power of parity polynomial time. *Mathematical Systems Theory*, 23(2):95–106, 1990.

[Coo71] S. A. Cook. The complexity of theorem-proving procedures. *Proceedings of the 3rd IEEE Symp. on the Foundations of Computer Science*, pages 151–158, 1971.

[DSW94] M. D. Davis, R. Sigal, and E. J. Weyuker. *Computability, Complexity and Languages*. Morgan Kaufmann, 340 Pine Street, Sixth Floor, San Francisco, CA 94104-3205, 2nd edition, 1994.

[FFK94] S. Fenner, L. Fortnow, and S. Kurtz. Gap-definable counting classes. *Journal of Computer and System Sciences*, 48(1):116–148, 1994.

[For79] S. Fortune. A note on sparse complete sets. *SIAM Journal on Computing*, 8(3):431–433, 1979.

[FR96] L. Fortnow and N. Reingold. PP is closed under truth-table reductions. *Information and Computation*, 124(1):1–6, 1996.

[Gil77] J. Gill. Computational complexity of probabilistic Turing machines. *SIAM Journal on Computing*, 6(4):675–695, 1977.

[GJ79] M. R. Garey and D. S. Johnson. *Computers and Intractability: A Guide to the Theory of NP-Completeness*. W. H. Freeman and Company, San Francisco, 1979.

[GP86] L. Goldschlager and I. Parberry. On the construction of parallel computers from various bases of boolean functions. *Theoretical Computer Science*, 43(1):43–58, 1986.

[GS88] J. Grollmann and A. Selman. Complexity measures for public-key cryptosystems. *SIAM Journal on Computing*, 17(2):309–335, 1988.

[Gup95] S. Gupta. Closure properties and witness reduction. *Journal of Computer and System Sciences*, 50(3):412–432, 1995.

[HH88] J. Hartmanis and L. Hemachandra. Complexity classes without machines: On complete languages for UP. *Theoretical Computer Science*, 58(1–3):129–142, 1988.

[HHN+95] L. Hemaspaandra, A. Hoene, A. Naik, M. Ogiwara, A. Selman, T. Thierauf, and J. Wang. Nondeterministically selective sets. *International Journal of Foundations of Computer Science*, 6(4):403–416, 1995.

[HNOS96] L. Hemaspaandra, A. Naik, M. Ogihara, and A. Selman. Computing solutions uniquely collapses the polynomial hierarchy. *SIAM Journal on Computing*, 25(4):697–708, 1996.

[HNP98] L. Hemaspaandra, C. Nasipak, and K. Parkins. A note on linear-nondeterminism, linear-sized, Karp-Lipton advice for the P-selective sets. *Journal of Universal Computer Science*, 4(8):670–674, 1998.

[HO02] L. A. Hemaspaandra and M. Ogihara. *The Complexity Theory Companion*. Texts in Theoretical Computer Science (EATCS). Springer-Verlag Berlin Heidelberg New York, 2002.

[HPR01] L. Hemaspaandra, K. Pasanen, and J. Rothe. If P != NP then some strongly noninvertible functions are invertible. In *Proceedings of the 13th International Symposium on Fundamentals of Computation Theory*, Lecture Notes in Computer Science, pages 162–171. Springer-Verlag, August 2001.

[HR99] L. Hemaspaandra and J. Rothe. Creating strong, total, commutative, associative one-way functions from any one-way function in complexity theory. *Journal of Computer and System Sciences*, 58(3):648–659, 1999.

[HT96] L. Hemaspaandra and L. Torenvliet. Optimal advice. *Theoretical Computer Science*, 154(2):367–377, 1996.

[HZ93] L. Hemaspaandra and M. Zimmand. Strong forms of balanced immunity. Technical Report TR-480, Department of Computer

Science, University of Rochester, Rochester, NY, December 1993. Revised, May 1994.

[Joc68] C. Jockusch. Semirecursive sets and positive reducibility. *Transactions of the AMS*, 131(2):420–436, 1968.

[Käm91] J. Kämper. Non-uniform proof systems: A new framework to describe non-uniform and probabilistic complexity classes. *Theoretical Computer Science*, 85(2):305–331, 1991.

[Kar72] R. M. Karp. Reducibility among combinatorial problems. In R. Miller and J. Thatcher, editors, *Complexity of Computer Computations*, pages 85–104. Plenum Press, New York, 1972.

[KL80] R. Karp and R. Lipton. Some connections between nonuniform and uniform complexity classes. In *Proceedings of the 12th ACM Symposium on Theory of Computing*, pages 302–309. ACM, ACM Press, April 1980.

[Ko83] K. Ko. On self-reducibility and weak P-selectivity. *Journal of Computer and System Sciences*, 26(2):209–221, 1983.

[Ko85] K. Ko. On some natural complete operators. *Theoretical Computer Science*, 37(1):1–30, 1985.

[Kre88] M. Krentel. The complexity of optimization problems. *Journal of Computer and System Sciences*, 36(3):490–509, 1988.

[Mah82] S. Mahaney. Sparse complete sets for NP: Solution of a conjecture of Berman and Hartmanis. *Journal of Computer and System Sciences*, 25(2):130–143, 1982.

[MS72] A. Meyer and L. Stockmeyer. The equivalence problem for regular expressions with squaring requires exponential space. In *Proceedings of the 13th IEEE Symposium on Switching and Automata Theory*, pages 125–129, October 1972.

[OH93] M. Ogiwara and L. Hemachandra. A complexity theory for feasible closure properties. *Journal of Computer and System Sciences*, 46(3):295–325, 1993.

[OW91] M. Ogiwara and O. Watanabe. On polynomial-time bounded truth-table reducibility of NP sets to sparse sets. *SIAM Journal on Computing*, 20(3):471–483, June 1991.

[Pap94] C. H. Papadimitriou. *Computational Complexity*. Addison-Wesley Publishing Company, Inc., 1994.

[PZ83] C. Papadimitriou and S. Zachos. Two remarks on the power of counting. In *Proceedings 6th GI Conference on Theoretical Computer Science*, volume 145 of *Lecture Notes in Computer Science*, pages 269–276. Springer-Verlag, January 1983.

[RS93] M. Rabi and A. Sherman. Associative one-way functions: A new paradigm for secret-key agreement and digital signatures. Technical Report CS-TR-3183/UMIACS-TR-93-124, Deparment of Computer Science, University of Maryland, College Park, Maryland, 1993.

[RS97] M. Rabi and A. Sherman. An observation on associative one-way functions in complexity theory. *Information Processing Letters*, 64(5):239–244, 1997.

[Sav70] W. J. Savitch. Relationship between nondeterministic and deterministic tape classes. *Journal of Computer and System Sciences*, 4:177–192, 1970.

[Sch76] C. P. Schnorr. Optimal algorithms for self-reducible problems. In R. Milner, editor, *3rd Int. Coll. on Automata, Languages, and Programming*, pages 322–337. Edinburgh University Press, 1976.

[Sel79] A. Selman. P-selective sets, tally languages, and the behavior of polynomial time reducibilities on NP. *Mathematical Systems Theory*, 13(1):55–65, 1979.

[Sel82a] A. Selman. Analogues of semirecursive sets and effective reducibilities to the study of NP complexity. *Information and Control*, 52(1):36–51, 1982.

[Sel82b] A. Selman. Reductions on NP and P-selective sets. *Theoretical Computer Science*, 19(3):287–304, 1982.

[Sel94] A. Selman. A taxonomy of complexity classes of functions. *Journal of Computer and System Sciences*, 48(2):357–381, 1994.

[She86] A. Sherman. *Cryptology and VLSI.* PhD thesis, Massachusettes Institute of Technology, Cambridge, MA, 1986. Available as Technical Report MIT/LCS/TR-381.

[Sim75] J. Simon. *On Some Central Problems in Computational Complexity.* PhD thesis, Cornell University, Ithaca, NY, January 1975. Available as Cornell Department of Computer Science Technical Report TR75-224.

[Sip97] M. Sipser. *Introduction to the Theory of Computation.* PWS Publishing Company, 1997.

[SM73] L. Stockmeyer and A. Meyer. Word problems requiring exponential time. In *Proc. 5th ACM Symp. on Theory of Computing*, pages 1–9, 1973.

[Sto76] L. Stockmeyer. The polynomial-time hierarchy. *Theoretical Computer Science*, 3(1):1–22, 1976.

[Val76] L. Valiant. The relative complexity of checking and evaluating. *Information Processing Letters*, 5(1):20–23, 1976.

[Val79] L. Valiant. The complexity of computing the permanent. *Theoretical Computer Science*, 8(2):189–201, 1979.

[Wag86] K. Wagner. The complexity of combinatorial problems with succinct input representations. *Acta Informatica*, 23(3):325–356, 1986.

[Wan] R. Wang. Lecture notes on NP-Completeness.

[Wat88] O. Watanabe. On hardness of one-way functions. *Information Processing Letters*, 27(3):151–157, 1988.