ΕΘΝΙΚΟ ΚΑΙ ΚΑΠΟΔΙΣΤΡΙΑΚΟ ΠΑΝΕΠΙΣΤΗΜΙΟ ΑΘΗΝΩΝ

ΣΧΟΛΗ ΘΕΤΙΚΩΝ ΕΠΙΣΤΗΜΩΝ
ΤΜΗΜΑ ΠΛΗΡΟΦΟΡΙΚΗΣ ΚΑΙ ΤΗΛΕΠΙΚΟΙΝΩΝΙΩΝ

ΠΤΥΧΙΑΚΗ ΕΡΓΑΣΙΑ

Ο Μη Ντετερμινιστικός Χώρος Είναι Κλειστός ως Προς το Συμπλήρωμα

Ευγένιος Ν. Μαζαράκης

Επιβλέπων: Σταύρος Κολλιόπουλος, Αναπληρωτής Καθηγητής

ΑΘΗΝΑ

ΣΕΠΤΕΜΒΡΙΟΣ 2015

ΠΤΥΧΙΑΚΗ ΕΡΓΑΣΙΑ

Ο Μη Ντετερμινιστικός Χώρος Είναι Κλειστός ως Προς το Συμπλήρωμα

Ευγένιος Ν. Μαζαράκης
Α.Μ: 1115200800110

Επιβλέπων: Σταύρος Κολλιόπουλος, Αναπληρωτής Καθηγητής

Περίληψη

Στη πτυχιακή αυτή εργασία εξηγείται η μελέτη του Neil Immerman που δημοσιεύτηκε το 1988 [23]. Για αυτή τη δημοσίευση του απονεμήθηκε το 1995 το βραβείο Gödel διότι απάντησε σε ένα ερώτημα που έμενε πάνω από τρεις δεκαετίες ανοικτό. Κύριος στόχος της εργασίας αυτής είναι να εξηγήσουμε τι απέδειξε ο Neil Immerman αλλά και με ποιο τρόπο τα κατάφερε. Αρχικά αναφέρονται όλοι οι απαραίτητοι ορισμοί γύρω από τις μηχανές Turing, την πολυπλοκότητα, τις κλάσεις πολυπλοκότητας, τις διάφορες σχέσεις που υπάρχουν ανάμεσά τους, τις γραμματικές. Αυτό είναι το απαραίτητο υπόβαθρο που θα χρειαστούμε για να ακολουθήσουμε το συλλογισμό του Neil Immerman. Μετά παρουσιάζονται και εξηγούνται με απλό και κατανοητό τρόπο τα περιεχόμενα της δημοσίευσης του. Τέλος ακολουθούν τα συμπεράσματα που προκύπτουν από την ενασχόλησή μας με τα αποτελέσματα στα οποία οδηγήθηκε ο Immerman.

ΘΕΜΑΤΙΚΗ ΠΕΡΙΟΧΗ: Θεωρία Υπολογισμού

ΛΕΞΕΙΣ ΚΛΕΙΔΙΑ: Πολυπλοκότητα Χώρου, Μηχανή Turing, Μη Ντετερμινισμός, Κλειστότητα ως Προς Συμπλήρωμα, Κλάσεις Πολυπλοκότητας

Αφιέρωση

Η πτυχιακή αυτή εργασία
είναι αφιερωμένη
στο Νίκο Μαζαράκη,
στην Έλλη Μαζαράκη,
στο Θεόφιλο Μαζαράκη,
στο Δημήτρη Ντουμπάρα
και στην αγαπημένη Δανάη Τουρλή.

Ευχαριστίες

Αισθάνομαι την ανάγκη να επισημάνω την απέραντη ευγνωμοσύνη στους γονείς μου Νίκο και Έλλη Μαζαράκη και στον αγαπημένο μου αδελφό Θεόφιλο Μαζαράκη, για όλα όσα μου έχουν προσφέρει στη διάρκεια των μαθητικών και φοιτητικών μου χρόνων αλλά και για την αμέριστη υποστήριξη τους σε κάθε μου επιλογή. Σε αυτό το σημείο, θα ήθελα να ευχαριστήσω για ακόμη μια φορά τον επιβλέποντα καθηγητή μου κ. Σταύρο Κολλιόπουλο για την εμπιστοσύνη και την αφιέρωση πολύτιμου χρόνου ώστε να ολοκληρωθεί η εργασία αυτή.

ΠΕΡΙΕΧΟΜΕΝΑ

Σελίδα

ΚΑΤΑΛΟΓΟΣ ΣΧΗΜΑΤΩΝ

ΠΡΟΛΟΓΟΣ

Στόχος αυτής της εργασίας είναι η εξήγηση της δημοσίευσης του Neil Immerman [23], για την οποία του απονεμήθηκε το βραβείο Gödel το 1995, διότι απάντησε σε ένα ερώτημα που παρέμενε αρκετές δεκαετίες ανοικτό. Η εργασία αποτελείται συνολικά από τρία κεφάλαια. Το πρώτο κεφάλαιο περιέχει όλους τους απαραίτητους ορισμούς γύρω από τις μηχανές Turing, ντετερμινιστικές και μη ντετερμινιστικές, την πολυπλοκότητα, τις κλάσεις πολυπλοκότητας, τις διάφορες σχέσεις που υπάρχουν ανάμεσά τους, τις γραμματικές. Αυτό είναι το απαραίτητο υπόβαθρο που χρειαζόμαστε για να είμαστε σε θέση να κατανοήσουμε το συλλογισμό του Immerman. Στο δεύτερο κεφάλαιο αναφέρονται και εξηγούνται τα περιεχόμενα της δημοσίευσης του με απλό και κατανοητό τρόπο. Στο τρίτο και τελευταίο κεφαλαίο αυτής της εργασίας υπάρχουν τα συμπεράσματα που προέκυψαν από τη μελέτη του γύρω από το μη ντετερμινιστικό χώρο. Επίσης γίνεται αναφορά σε μερικά σημαντικά προβλήματα της θεωρίας υπολογισμού που παραμένουν ακόμα ανοικτά, περιμένοντας κάποιον να τα επιλύσει με απλό και λιτό τρόπο όπως έκανε ο Immerman στη δημοσίευση που εξηγούμε παρακάτω.

1. ΕΙΣΑΓΩΓΗ

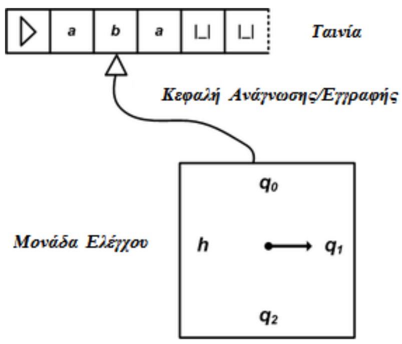
1.1. Μηχανές Turing

Το 1936 ο Άγγλος μαθηματικός Alan Turing πρότεινε και περιέγραψε τυπικά ένα γενικό μοντέλο υπολογισμού το οποίο έμεινε γνωστό ως μηχανή Turing. Είναι το κατάλληλο μαθηματικό μοντέλο για την τυπική περιγραφή του υπολογισμού όπως αυτός εκτελείται από τους σύγχρονους υπολογιστές γενικής χρήσης. Μια μηχανή Turing, για την ακρίβεια μια καθολική μηχανή, μπορεί να υπολογίσει οτιδήποτε μπορεί και ένας υπολογιστής.

Η μηχανή Turing αποτελείται από μία μονάδα ελέγχου πεπερασμένων καταστάσεων και μια ταινία, που είναι απεριόριστη ως προς το ένα άκρο. Η επικοινωνία μεταξύ τους επιτυγχάνεται με μια κεφαλή, η οποία διαβάζει σύμβολα από την ταινία και χρησιμοποιείται επίσης για να μεταβάλλει τα σύμβολα στην ταινία. Η μονάδα ελέγχου λειτουργεί σε διακριτά βήματα, σε κάθε βήμα εκτελεί δύο λειτουργίες με τρόπο που εξαρτάται από την τρέχουσα κατάσταση της και το σύμβολο της ταινίας που μόλις διαβάστηκε από την κεφαλή ανάγνωσης/εγγραφής.

Αρχικά η ταινία περιέχει την είσοδο, μια ακολουθία από σύμβολα που δεν περιέχει το κενό σύμβολο, που μπορεί όμως να βρίσκεται οπουδήποτε αλλού μέσα στη ταινία. Η μηχανή αρχίζει να κάνει υπολογισμούς έως ότου αποφασίσει να σταματήσει παράγωντας μια έξοδο. Φυσικά, σε πολλές περιπτώσεις τίποτα δε μπορεί να μας εγγυηθεί ότι η μηχανή κάποτε θα τερματίσει. Στην περίπτωση αυτή η μηχανή συνεχίζει τον υπολογισμό της για πάντα.

Οι ορισμοί που ακολουθούν παρακάτω βασίζονται στα βιβλία των Παπαδημητρίου, Lewis [20] και Sipser [21].



Σχήμα 1.1: Μηχανή Turing.

1.1.1. Ντετερμινιστική Μηχανή Turing

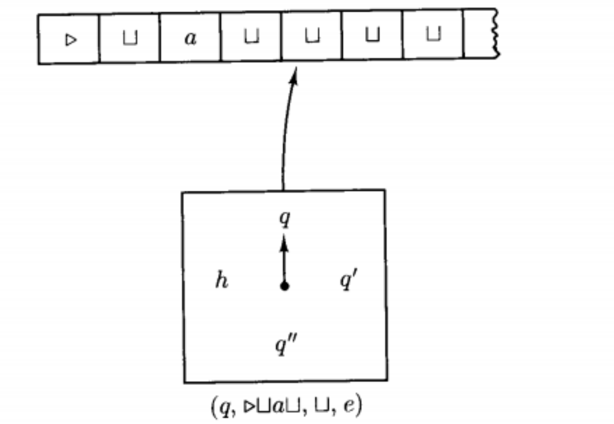
Μπορούμε τώρα να παρουσιάσουμε μια σειρά από ορισμούς που αφορούν τις ντετερμινιστικές μηχανές Turing.

Ορισμός 1.1. Μια πρότυπη μηχανή Turing είναι μια πεντάδα (K, Σ, δ, s, H) όπου K είναι ένα πεπερασμένο σύνολο καταστάσεων, Σ είναι ένα αλφάβητο που περιέχει το κενό σύμβολο $\sqcup$ και το σύμβολο του αριστερού άκρου $\triangleright$, αλλά δεν περιέχει τα σύμβολα αριστερό βέλος $\leftarrow$ και δεξί βέλος $\rightarrow$:

- s $\in$ K είναι η αρχική κατάσταση
- H $\subseteq$ K είναι το σύνολο των καταστάσεων τερματισμού
- δ είναι η συνάρτηση μετάβασης της μηχανής από το $(K-H) \times \Sigma$ στο $K \times (\Sigma \cup \{\leftarrow, \rightarrow\})$ τέτοια ώστε:

  1. $\forall$ q $\in$ (K$-$H), αν δ(q,$\triangleright$) $=$ (p,b) τότε b$=\rightarrow$.
  2. $\forall$ q $\in$ (K$-$H) και α $\in$ Σ, αν δ(q,α) $=$ (p,b) τότε b$\neq \triangleright$.

Ορισμός 1.2. Η συνολική κατάσταση μίας μηχανής Turing M $=$ (K, Σ, δ, s, H) είναι μέλος του $K \times \triangleright\Sigma^* \times (\Sigma^* (\Sigma - \{\sqcup\}) \cup \{e\})$.



Σχήμα 1.2: Συνολική Κατάσταση Μηχανής Turing.

Ορισμός 1.3. Έστω M $=$ (K, Σ, δ, s, H) μια μηχανή Turing και $(q_1, w_1\underline{a_1}u_1)$ και $(q_2, w_2\underline{a_2}u_2)$ δύο συνολικές καταστάσεις της M, όπου a_1, $a_2 \in \Sigma$. Τότε

$$(q_1, w_1\underline{a_1}u_1) \vdash_M (q_2, w_2\underline{a_2}u_2)$$

αν και μόνον αν, για κάποιο b $\in \Sigma \cup \{\leftarrow, \rightarrow\}$, $\delta(q_1, a_1) = (q_2, a_2)$ και είτε:

1. b $\in$ Σ, $w_1 = w_2$, $u_1 = u_2$ και $a_2 = b$ είτε
2. b $=\leftarrow$, $w_1 = w_2 a_2$, και είτε

   (a) $u_2 = a_1u_1$, αν $a_1 \neq \sqcup$ ή $u_1 \neq e$ ή
   (b) $u_2 = e$, αν $a_1 = \sqcup$ και $u_1 = e$ είτε

3. $b = \rightarrow$, $w_2 = w_1a_1$, και είτε

   (a) $u_1 = a_2u_2$ ή
   (b) $u_1 = u_2 = e$ και $a_2 = \sqcup$

Ορισμός 1.4. Για κάθε μηχανή Turing M, έστω $\vdash^*_M$ η ανακλαστική, μεταβατική κλειστότητα της $\vdash_M$, λέμε ότι η συνολική κατάσταση C_1 παράγει τη συνολική κατάσταση C_2 αν ισχύει ότι $C_1 \vdash^*_M C_2$. Ένας υπολογισμός της μηχανής είναι μια ακολουθία από συνολικές καταστάσεις $C_0, C_1, \ldots, C_n$, για κάποιο $n \geq 0$ τέτοιο ώστε

$$C_0 \vdash_M C_1 \vdash_M C_2 \vdash_M \ldots \vdash_M C_n$$

Λέμε ότι ο υπολογισμός είναι μήκους n ή ότι έχει n βήματα, και γράφουμε $C_0 \vdash^n_M C_n$.

1.1.2. Μη Ντετερμινιστική Μηχανή Turing

Στην παρούσα ενότητα θα ορίσουμε ένα μοντέλο μηχανών Turing που δεν ανταποκρίνεται στην έννοια του πραγματικού υπολογισμού. Πέραν τούτου, το νέο μοντέλο θα είναι ισοδύναμο με το κλασικό μοντέλο της μηχανής, με την έννοια ότι κάθε μηχανή Turing είναι σε θέση να προσομοιώσει το νέο μη ντετερμινιστικό μοντέλο.

Ορισμός 1.5. Μια μη ντετερμινιστική μηχανή Turing είναι μια πεντάδα (K, Σ, δ, s, H) όπου K είναι ένα πεπερασμένο σύνολο καταστάσεων, Σ είναι ένα αλφάβητο, που περιέχει το κενό σύμβολο $\sqcup$ και το σύμβολο του αριστερού άκρου $\triangleright$, αλλά δεν περιέχει τα σύμβολα αριστερό βέλος $\leftarrow$ και δεξί βέλος $\rightarrow$:

- $s \in K$ είναι η αρχική κατάσταση
- $H \subseteq K$ είναι το σύνολο καταστάσεων τερματισμού
- Δ είναι ένα υποσύνολο του $((K-H) \times \Sigma) \times (K \times (\Sigma \cup \{\leftarrow, \rightarrow\}))$

   1. $\forall\, q \in K-H$, αν $(q,\triangleright,p,b) \in \Delta$, τότε $b = \rightarrow$
   2. $\forall\, q \in K-H$ και $\forall\, a \in \Sigma$, αν $(q,a,p,b) \in \Delta$, τότε $b \neq \triangleright$

Από τα παραπάνω, γίνεται σαφές ότι κάθε ντετερμινιστική μηχανή Turing είναι και μη ντετερμινιστική, καθώς οι συναρτήσεις είναι ειδική περίπτωση σχέσεων. Δηλαδή οι ντετερμινιστικές μηχανές Turing είναι στην ουσία μη ντετερμινιστικές μηχανές για τις οποίες το επόμενο βήμα είναι πάντοτε καλά καθορισμένο.

1.1.3. Υπολογισμοί Με Μηχανές Turing

Γενικά οι μηχανές Turing φαίνονται ιδανικές για να λύνουν συγκεκριμένους τύπους προβλημάτων ή να υπολογίζουν συναρτήσεις ως προς διάφορες συμβολοσειρές, δηλαδή φαίνονται ικανές να υπολογίζουν και να αποφασίζουν ή να αποδέχονται διάφορες γλώσσες. Για να πραγματοποιήσουν τα υπολογιστικά τους καθήκοντα, υιοθετούμε την εξής σύμβαση για να αναπαριστούμε τα στοιχεία εισόδου των μηχανών Turing: Η συμβολοσειρά εισόδου, χωρίς κενά σύμβολα, γράφεται στα δεξιά του συμβόλου αριστερού άκρου $\rhd$, με ένα κενό σύμβολο στα αριστερά της και κενά στα δεξιά της, η κεφαλή είναι τοποθετημένη στο τετράγωνο της ταινίας που περιέχει το κενό μεταξύ του $\rhd$ και της εισόδου και η μηχανή αρχίζει να λειτουργεί από την αρχική της κατάσταση. Αν M $=$ (K, Σ, δ, s, H) είναι μια μηχανή και w $\in$ (Σ$-$ $\{\sqcup, \rhd\}$)*, τότε η αρχική συνολική κατάσταση της M με είσοδο w είναι (s,$\rhd\underline{\sqcup}$ w).

Ορισμός 1.6. Έστω M $=$ (K, Σ, δ, s, H) μια μηχανή Turing, τέτοια ώστε το H $=$ $\{$y,n$\}$ να αποτελείται από δύο διακεκριμένες καταστάσεις τερματισμού. Κάθε συνολική κατάσταση τερματισμού της οποίας η κατάσταση τερματισμού είναι η y ονομάζεται συνολική κατάσταση αποδοχής, ενώ μία συνολική κατάσταση τερματισμού της οποίας η κατάσταση τερματισμού είναι η n ονομάζεται συνολική κατάσταση απόρριψης. Λέμε ότι η M δέχεται ως είσοδο μια συμβολοσειρά w $\in$ (Σ $-$ $\{\sqcup, \rhd\}$)* αν η (s,$\rhd\underline{\sqcup}$ w) παράγει μια συνολική κατάσταση αποδοχής και λέμε ότι η M απορρίπτει τη w αν η (s,$\rhd\underline{\sqcup}$ w) παράγει μια συνολική κατάσταση απόρριψης.

Τώρα λέμε ότι η M αποφασίζει μια γλώσσα L $\subseteq$ Σ_0^* αν είναι αληθές το εξής: Αν w $\in$ L, η M δέχεται τη w και αν w $\notin$ L , η M απορρίπτει τη w. Ονομάζουμε μια γλώσσα L αναδρομική αν υπάρχει μια μηχανή Turing που την αποφασίζει.

Ορισμός 1.7. Έστω M $=$ (K, Σ, δ, s, $\{$h$\}$) μια μηχανή Turing, Σ_0 $\subseteq$ (Σ $-$ $\{\sqcup, \rhd\}$) ένα αλφάβητο και έστω w $\in$ Σ_0^*. Υποθέτουμε ότι η M τερματίζει με είσοδο w και ότι (s,$\rhd\underline{\sqcup}$ w) $\vdash_M^*$ (h,$\rhd\underline{\sqcup}$ y) για κάποιο y $\in$ Σ_0^*. Τότε το y ονομάζεται η έξοδος της M με είσοδο w και συμβολίζεται M(w). Παρατηρήστε ότι η M(w) ορίζεται μόνον αν η M τερματίζει με είσοδο w και στην πραγματικότητα αυτό συμβαίνει σε μια συνολική κατάσταση της μορφής με (h,$\rhd\underline{\sqcup}$ y) με y $\in$ Σ_0^*.

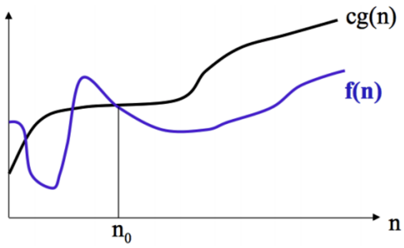
Τώρα έστω f μια οποιαδήποτε συνάρτηση από το Σ_0^* στο Σ_0^*. Λέμε ότι η M υπολογίζει τη συνάρτηση f αν για κάθε w $\in$ Σ_0^*, M(w) $=$ f(w). Δηλαδή για κάθε w $\in$ Σ_0^*, η M κάποια στιγμή τερματίζει με είσοδο w και όταν πράγματι τερματίσει η ταινία της περιέχει τη συμβολοσειρά ($\rhd\sqcup$f(w)). Μια συνάρτηση f ονομάζεται αναδρομική, αν υπάρχει μια μηχανή Turing M που υπολογίζει τη f.

1.2. Πολυπλοκότητα

Ένας τρόπος για να εκτιμήσουμε την επίδοση ενός αλγορίθμου είναι ο θεωρητικός (theoretical) ή ο λεγόμενος εκ των προτέρων (a priori). Για το σκοπό αυτό εισάγεται μια μεταβλητή n, που εκφράζει το μέγεθος (size) του προβλήματος, ώστε η μέτρηση της αποδοτικότητας του αλγορίθμου να ισχύει για οποιοδήποτε σύνολο δεδομένων και ανεξάρτητα από υποκειμενικούς παράγοντες. Ο χρόνος επεξεργασίας και ο απαιτούμενος χώρος μνήμης εκτιμώνται με τη βοήθεια μιας συνάρτησης f που εκφράζει τη χρονική πολυπλοκότητα (Time Complexity) ή τη χωρική πολυπλοκότητα (Space Complexity). Σε πολλές περιπτώσεις όμως δεν ενδιαφέρουν οι επακριβείς τιμές αλλά μόνο η γενική συμπεριφορά του αλγορίθμου, δηλαδή η τάξη του αλγορίθμου. Για το λόγο αυτό εισάγουμε τους παρακάτω ορισμούς που ορίζουν τη συμπεριφορά συναρτήσεων πολυπλοκότητας ασυμπτωτικά.

Ορισμός 1.8. Συμβολισμός O (big-O notation), ορίζει ένα ασυμπτωτικό άνω φράγμα. Για μια δεδομένη συνάρτηση g(n), η έκφραση O(g(n)) δηλώνει το σύνολο των συναρτήσεων:

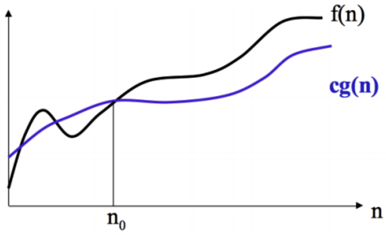
$$O(g(n)) = \{f(n) : \exists\ c, n_0 > 0 \text{ τέτοια ώστε } 0 \leq f(n) \leq cg(n), \forall\ n \geq n_0\}$$



Σχήμα 1.3: $f(n) \in O(g(n))$

Ορισμός 1.9. Συμβολισμός Ω (big-Ω notation), ορίζει ένα ασυμπτωτικό κάτω φράγμα. Για μια δεδομένη συνάρτηση g(n), η έκφραση Ω(g(n)) δηλώνει το σύνολο των συναρτήσεων:

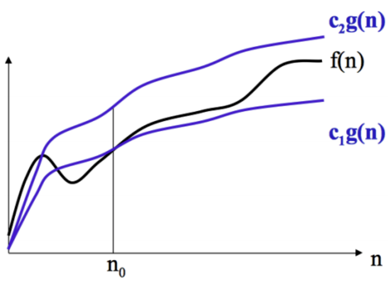
$$\Omega(g(n)) = \{f(n) : \exists\ c, n_0 > 0 \text{ τέτοια ώστε } 0 \leq cg(n) \leq f(n), \forall\ n \geq n_0\}$$



Σχήμα 1.4: $f(n) \in \Omega(g(n))$

Ορισμός 1.10. Συμβολισμός Θ (big-Θ notation), ορίζει ένα ασυμπτωτικό αυστηρό φράγμα. Για μια δεδομένη συνάρτηση g(n), η έκφραση Θ(g(n)) δηλώνει το σύνολο των συναρτήσεων:

$$\Theta(g(n)) = \{f(n) : \exists\ c_1, c_2, n_0 > 0 \text{ τέτοια ώστε } 0 \leq c_1 g(n) \leq f(n) \leq c_2 g(n), \forall\ n \geq n_0\}$$



Σχήμα 1.5: $f(n) \in \Theta(g(n))$

Ορισμός 1.11. Συμβολισμός ο (small-o notation), ορίζει ένα αυστηρό ασυμπτωτικό άνω φράγμα. Για μια δεδομένη συνάρτηση g(n), η έκφραση o(g(n)) δηλώνει το σύνολο των συναρτήσεων:

$$o(g(n)) = \{f(n) : \exists\ c, n_0 > 0 \text{ τέτοια ώστε } 0 \leq f(n) < cg(n), \forall\ n \geq n_0\}$$

Ορισμός 1.12. Συμβολισμός ω (small-ω notation), ορίζει ένα αυστηρό ασυμπτωτικό κάτω φράγμα. Για μια δεδομένη συνάρτηση g(n), η έκφραση ω(g(n)) δηλώνει το σύνολο των συναρτήσεων:

$$\omega(g(n)) = \{f(n) : \exists\ c, n_0 > 0 \text{ τέτοια ώστε } 0 \leq cg(n) < f(n), \forall\ n \geq n_0\}$$

Στις επόμενες ενότητες θα ορίσουμε την έννοια του χρόνου που διαρκεί ένας υπολογισμός αλλά και του χώρου που απαιτεί μια μηχανή Turing για έναν υπολογισμό, ντετερμινιστική και μη ντετερμινιστική.

1.2.1. Πολυπλοκότητα Χρόνου

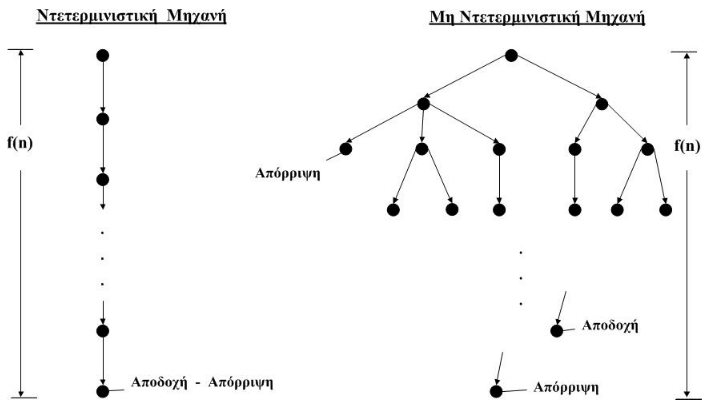
Στην ενότητα αυτή θα ορίσουμε τον τρόπο με τον οποίο μετράται ο χρόνος εκτέλεσης μιας μηχανής Turing, ως συνάρτηση του μήκους μιας συμβολοσειράς που παριστάνει την είσοδο της. Στην ανάλυση της χείριστης περίπτωσης, όπως εδώ, λαμβάνουμε υπόψιν το μεγαλύτερο χρόνο εκτέλεσης από όλες τις εισόδους ενός συγκεκριμένου μήκους.

Ορισμός 1.13. Έστω M μια ντετερμινιστική μηχανή Turing που τερματίζει για κάθε είσοδο. Ο χρόνος εκτέλεσης ή η χρονική πολυπλοκότητα της M είναι η συνάρτηση $f : \mathbb{N} \rightarrow \mathbb{N}$ που για κάθε n επιστρέφει ως f(n) το μέγιστο πλήθος βημάτων που είναι δυνατόν να πραγματοποιήσει η M όταν το μήκος της εισόδου της είναι n. Εάν ο χρόνος εκτέλεσης της M είναι f(n), λέμε ότι η M εκτελείται σε χρόνο f(n) ή ότι είναι μια μηχανή Turing χρόνου f(n). Κατά σύμβαση, το μήκος της εισόδου αναπαρίσταται πάντοτε από τη μεταβλητή n.

Ουσιαστικά για οποιαδήποτε συμβολοσειρά w, με |w|=n, η μηχανή Turing θα κάνει το πολύ f(n) βήματα, πριν φτάσει σε μια συνολική κατάσταση τερματισμού.

Ορισμός 1.14. Έστω M μια μη ντετερμινιστική μηχανή Turing, όπου όλοι οι κλάδοι των υπολογισμών της τερματίζουν για κάθε είσοδο. Ο χρόνος εκτέλεσης ή η χρονική πολυπλοκότητα της M είναι η συνάρτηση $f : \mathbb{N} \rightarrow \mathbb{N}$ η οποία για κάθε n επιστρέφει ως f(n) το μέγιστο πλήθος βημάτων που είναι δυνατόν να πραγματοποιήσει η M σε οποιονδήποτε κλάδο του υπολογισμού της για οποιαδήποτε είσοδο μήκους n. Εάν ο χρόνος εκτέλεσης της M είναι f(n), λέμε ότι η M εκτελείται σε χρόνο f(n) ή ότι είναι μια μη ντετερμινιστική μηχανή Turing χρόνου f(n). Κατά σύμβαση, το μήκος της εισόδου αναπαρίσταται πάντοτε από τη μεταβλητή n.

Ουσιαστικά για οποιαδήποτε συμβολοσειρά w, με |w|=n, η μη ντετερμινιστική μηχανή Turing θα κάνει το πολύ f(n) βήματα πριν κάποιος κλάδος κάποιου υπολογισμού φτάσει σε μια συνολική κατάσταση τερματισμού.



Σχήμα 1.6: Μέτρηση του Ντετερμινιστικού και του Μη Ντετερμινιστικού χρόνου.

1.2.2. Πολυπλοκότητα Χώρου

Στην ενότητα αυτή θα ορίσουμε τον τρόπο με τον οποίο μετράται ο χώρος, δηλαδή η μνήμη, που απαιτεί μια μηχανή Turing κατά τη διάρκεια της εκτέλεσης της, ως συνάρτηση του μήκους μιας συμβολοσειράς που παριστάνει την είσοδο της.

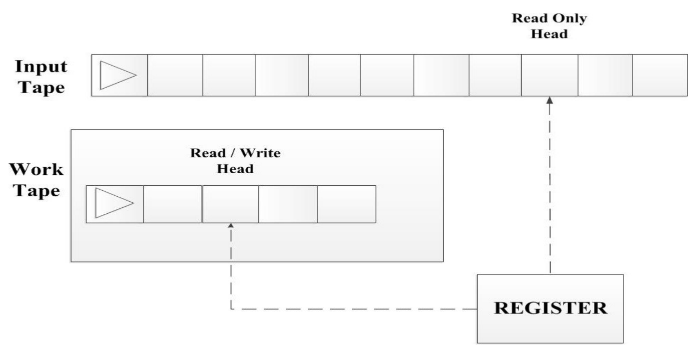
Είδαμε παραπάνω ότι η χρονική πολυπλοκότητα μιας μηχανής θα είναι τουλάχιστον γραμμική, αφού γραμμικό χρόνο χρειάζεται για την πλήρη ανάγνωση της εισόδου. Όμως η χωρική πολυπλοκότητα μιας μηχανής μπορεί να είναι τουλάχιστον υπογραμμική. Δηλαδή μια μηχανή Turing με υπογραμμική χωρική πολυπλοκότητα έχει τη δυνατότητα να διαβάσει ολόκληρη την είσοδο, παρ' ότι δε διαθέτει αρκετό χώρο για να την αποθηκεύσει. Για το λόγο αυτό θα τροποποιήσουμε λίγο το υπολογιστικό μοντέλο σε σχέση με τη χρονική πολυπλοκότητα και θα χρησιμοποιήσουμε μηχανή Turing 2-ταινιών.

Ορισμός 1.15. Έστω M μια ντετερμινιστική μηχανή Turing 2-ταινιών που τερματίζει για όλες τις εισόδους. Η 1η ταινία περιέχει την είσοδο και είναι μόνο αναγνώσιμη. Η 2η ταινία, η ταινία εργασίας είναι ταυτόχρονα εγγράψιμη και αναγνώσιμη. Η χωρική πολυπλοκότητα της M είναι η συνάρτηση $f : \mathbb{N} \to \mathbb{N}$, που για κάθε n επιστρέφει ως f(n) το μέγιστο πλήθος θέσεων της 2ης ταινίας που είναι δυνατόν να διατρέξει η M όταν το μήκος της εισόδου της είναι n. Εάν η χωρική πολυπλοκότητα της M είναι f(n), λέμε επίσης ότι η M λειτουργεί σε χώρο f(n) ή ότι έχει χώρο εκτέλεσης f(n).

Ουσιαστικά για οποιαδήποτε συμβολοσειρά w, με $|w|$=n, η μηχανή Turing θα σαρώσει το πολύ f(n) κελιά της ταινίας εργασίας, πριν φτάσει σε μια συνολική κατάσταση τερματισμού.

Ορισμός 1.16. Έστω M μια μη ντετερμινιστική μηχανή Turing 2-ταινιών τέτοια ώστε για κάθε είσοδο να τερματίζουν όλοι οι κλάδοι του υπολογισμού. Η 1η ταινία περιέχει την είσοδο και είναι μόνο αναγνώσιμη. Η 2η ταινία, η ταινία εργασίας, είναι ταυτόχρονα εγγράψιμη και αναγνώσιμη. Η χωρική πολυπλοκότητα της M είναι η συνάρτηση $f : \mathbb{N} \to \mathbb{N}$, που για κάθε n επιστρέφει ως f(n) το μέγιστο πλήθος θέσεων της 2ης ταινίας που είναι δυνατόν να διατρέξει η M σε οποιονδήποτε κλάδο του υπολογισμού της για οποιαδήποτε είσοδο μήκους n. Εάν η χωρική πολυπλοκότητα της M είναι f(n), λέμε επίσης ότι η M λειτουργεί σε χώρο f(n) ή ότι έχει χώρο εκτέλεσης f(n).

Ουσιαστικά για οποιαδήποτε συμβολοσειρά w, με $|w|$=n, η μηχανή Turing θα σαρώσει το πολύ f(n) κελιά της ταινίας εργασίας, πριν κάποιος κλάδος του υπολογισμού φτάσει σε μια συνολική κατάσταση τερματισμού.



Σχήμα 1.7: Μέτρηση του Ντετερμινιστικού και του Μη Ντετερμινιστικού χώρου.

Λήμμα 1.1. *Έστω μια μηχανή Turing M = (K, Σ, δ, s, H) χώρου s(n). Έστω μια συμβολοσειρά w, με $|w|$=n. Τότε η M(w) έχει στη διάθεσή της $s(|w|) = p$ κελιά για να κάνει τους υπολογισμούς της στην ταινία εργασίας. Το πλήθος των συνολικών καταστάσεων της μηχανής Turing M είναι $|K| * p * |\Sigma|^p$, άρα μπορεί να είναι $L = c^p$ για κάποια σταθερά c.*

1.3. Κλάσεις Πολυπλοκότητας

Μια κλάση πολυπλοκότητας δεν είναι τίποτε άλλο παρά ένα σύνολο γλωσσών με κοινές ιδιότητες. Υπάρχουν διάφορες παράμετροι που προσδιορίζουν μια κλάση πολυπλοκότητας. Κατ' αρχήν το μοντέλο του υπολογισμού, έχουμε συμφωνήσει ότι θα χρησιμοποιήσουμε τη μηχανή Turing. Ακόμα μια κλάση πολυπλοκότητας χαρακτηρίζεται από τον τρόπο του υπολογισμού, δηλαδή αν η μηχανή μας θα είναι ντετερμινιστική ή μη. Μια ακόμα παράμετρος είναι ο υπολογιστικός πόρος που θέλουμε να μετρήσουμε, δηλαδή το χρόνο ή το χώρο που απαιτεί η μηχανή Turing. Τέλος πρέπει να καθορίσουμε ένα φράγμα για τον υπολογισμό, δηλαδή μια συνάρτηση f: $\mathbb{N} \rightarrow \mathbb{N}$.

Ορισμός 1.17. Έστω L μια γλώσσα. Η $\overline{L}$, είναι η γλώσσα που περιλαμβάνει όλες τις συμβολοσειρές του Σ^* που δεν ανήκουν στην L. Επίσης έστω **C** μια κλάση πολυπλοκότητας. Η συμπληρωματική της κλάση, η **coC** ορίζεται σαν:

$$\textbf{coC} = \{ L \mid L \subseteq \Sigma^*, \overline{L} \in \textbf{C} \}$$

1.3.1. Κλάσεις Χρονικής Πολυπλοκότητας

Η χρονική πολυπλοκότητα μας δίνει τη δυνατότητα να ταξινομήσουμε τα διάφορα προβλήματα με βάση το χρόνο που απαιτούν για να επιλυθούν.

Ορισμός 1.18. Έστω t: $\mathbb{N} \rightarrow \mathbb{R}^+$ οποιαδήποτε συνάρτηση. Ορίζουμε ως κλάση χρονικής πολυπλοκότητας **TIME(t(n))** τη συλλογή όλων των γλωσσών που μπορούν να διαγνωστούν από κάποια ντετερμινιστική μηχανή Turing χρόνου O(t(n)).

Η κλάση **TIME(t(n))** περιέχει όλες εκείνες τις γλώσσες για τις οποίες υπάρχει κάποια ντετερμινιστική μηχανή Turing με χρονική πολυπλοκότητα t(n) που μπορεί να τις αποφασίσει.

Ορισμός 1.19. Έστω t: $\mathbb{N} \rightarrow \mathbb{R}^+$ οποιαδήποτε συνάρτηση. Ορίζουμε ως κλάση χρονικής πολυπλοκότητας **NTIME(t(n))** τη συλλογή όλων των γλωσσών που μπορούν να διαγνωστούν από κάποια μη ντετερμινιστική μηχανή Turing χρόνου O(t(n)).

Η κλάση **NTIME(t(n))** περιέχει όλες εκείνες τις γλώσσες για τις οποίες υπάρχει κάποια μη ντετερμινιστική μηχανή Turing με χρονική πολυπλοκότητα t(n) που μπορεί να τις αποφασίσει.

Μερικές σημαντικές κλάσεις χρονικής πολυπλοκότητας είναι οι ακόλουθες:

Ορισμός 1.20. Η κλάση γλωσσών **P** αποτελείται από τις γλώσσες που μπορούν να διαγνωστούν σε πολυωνυμικό χρόνο από κάποια ντετερμινιστική μηχανή Turing. Με άλλα λόγια,

$$\textbf{P} = \bigcup_{\mathbf{k} \in \mathbb{N}} \textbf{TIME}(\mathbf{n^k})$$

Η κλάση **P** περιέχει όλες τις γλώσσες για τις οποίες υπάρχει κάποια ντετερμινιστική μηχανή Turing που τις αποφασίζει ύστερα από πολυωνυμικό αριθμό βημάτων.

Ορισμός 1.21. Η κλάση γλωσσών **NP** αποτελείται από τις γλώσσες που μπορούν να διαγνωστούν σε πολυωνυμικό χρόνο από κάποια μη ντετερμινιστική μηχανή Turing. Με άλλα λόγια,

$$\mathbf{NP} = \bigcup_{\mathbf{k}\in\mathbb{N}} \mathbf{NTIME(n^k)}$$

Η κλάση **NP** περιέχει όλες τις γλώσσες για τις οποίες υπάρχει κάποια μη ντετερμινιστική μηχανή Turing που τις αποφασίζει ύστερα από πολυωνυμικό αριθμό βημάτων.

Ορισμός 1.22. Η κλάση γλωσσών **EXPTIME** αποτελείται από τις γλώσσες που μπορούν να διαγνωστούν σε εκθετικό χρόνο από κάποια ντετερμινιστική μηχανή Turing. Με άλλα λόγια,

$$\mathbf{EXPTIME} = \bigcup_{\mathbf{k}\in\mathbb{N}} \mathbf{TIME(2^{n^k})}$$

Η κλάση **EXPTIME** περιέχει όλες τις γλώσσες για τις οποίες υπάρχει κάποια ντετερμινιστική μηχανή Turing που τις αποφασίζει ύστερα από εκθετικό αριθμό βημάτων.

Ορισμός 1.23. Η κλάση γλωσσών **NEXPTIME** αποτελείται από τις γλώσσες που μπορούν να διαγνωστούν σε εκθετικό χρόνο από κάποια μη ντετερμινιστική μηχανή Turing. Με άλλα λόγια,

$$\mathbf{NEXPTIME} = \bigcup_{\mathbf{k}\in\mathbb{N}} \mathbf{NTIME(2^{n^k})}$$

Η κλάση **NEXPTIME** περιέχει όλες τις γλώσσες για τις οποίες υπάρχει κάποια μη ντετερμινιστική μηχανή Turing που τις αποφασίζει ύστερα από εκθετικό αριθμό βημάτων.

1.3.2. Κλάσεις Χωρικής Πολυπλοκότητας

Η χωρική πολυπλοκότητα διαθέτει πολλά από τα βασικά χαρακτηριστικά της χρονικής και μας παρέχει έναν ακόμη τρόπο ταξινόμησης των διαφόρων προβλημάτων με βάση το χώρο, τη μνήμη, που απαιτούν για να επιλυθούν.

Ορισμός 1.24. Έστω f: $\mathbb{N} \rightarrow \mathbb{R}^+$ οποιαδήποτε συνάρτηση. Ορίζουμε ως κλάση χωρικής πολυπλοκότητας **SPACE(f(n))** τη συλλογή όλων των γλωσσών που μπορούν να διαγνωστούν από κάποια ντετερμινιστική μηχανή Turing χώρου O(f(n)).

Η κλάση **SPACE(f(n))** περιέχει όλες εκείνες τις γλώσσες για τις οποίες υπάρχει κάποια ντετερμινιστική μηχανή Turing με χωρική πολυπλοκότητα f(n) που μπορεί να τις αποφασίσει.

Ορισμός 1.25. Έστω f: $\mathbb{N} \rightarrow \mathbb{R}^+$ οποιαδήποτε συνάρτηση. Ορίζουμε ως κλάση χωρικής πολυπλοκότητας **NSPACE(f(n))** τη συλλογή όλων των γλωσσών που μπορούν να διαγνωστούν από κάποια μη ντετερμινιστική μηχανή Turing χώρου O(f(n)).

Η κλάση **NSPACE(f(n))** περιέχει όλες εκείνες τις γλώσσες για τις οποίες υπάρχει κάποια μη ντετερμινιστική μηχανή Turing με χωρική πολυπλοκότητα f(n) που μπορεί να τις αποφασίσει.

Μερικές σημαντικές κλάσεις χωρικής πολυπλοκότητας είναι οι ακόλουθες:

Ορισμός 1.26. Η κλάση **PSPACE** αποτελείται από τις γλώσσες που μπορούν να διαγνωστούν σε πολυωνυμικό χώρο από κάποια ντετερμινιστική μηχανή Turing. Με άλλα λόγια,

$$\mathbf{PSPACE} = \bigcup_{\mathbf{k}\in\mathbb{N}} \mathbf{SPACE(n^k)}$$

Η κλάση **PSPACE** περιέχει όλες τις γλώσσες για τις οποίες υπάρχει κάποια ντετερμινιστική μηχανή Turing που τις αποφασίζει ύστερα από τη χρήση πολυωνυμικού αριθμού κελιών της ταινίας εργασίας.

Ορισμός 1.27. Η κλάση **NPSPACE** αποτελείται από τις γλώσσες που μπορούν να διαγνωστούν σε πολυωνυμικό χώρο από κάποια μη ντετερμινιστική μηχανή Turing. Με άλλα λόγια,

$$\mathbf{NPSPACE} = \bigcup_{\mathbf{k} \in \mathbb{N}} \mathbf{NSPACE(n^k)}$$

Η κλάση **NPSPACE** περιέχει όλες τις γλώσσες για τις οποίες υπάρχει κάποια μη ντετερμινιστική μηχανή Turing που τις αποφασίζει ύστερα από τη χρήση πολυωνυμικού αριθμού κελιών της ταινίας εργασίας.

Ορισμός 1.28. Η κλάση **L** αποτελείται από τις γλώσσες που μπορούν να διαγνωστούν από κάποια ντετερμινιστική μηχανή Turing σε λογαριθμικό χώρο. Με άλλα λόγια,

$$\mathbf{L} = \mathbf{SPACE(log(n))}$$

Η κλάση **L** περιέχει όλες τις γλώσσες για τις οποίες υπάρχει κάποια ντετερμινιστική μηχανή Turing που τις αποφασίζει ύστερα από τη χρήση λογαριθμικού αριθμού κελιών της ταινίας εργασίας.

Ορισμός 1.29. Η κλάση **NL** αποτελείται από τις γλώσσες που μπορούν να διαγνωστούν από κάποια μη ντετερμινιστική μηχανή Turing σε λογαριθμικό χώρο. Με άλλα λόγια,

$$\mathbf{NL} = \mathbf{NSPACE(log(n))}$$

Η κλάση **NL** περιέχει όλες τις γλώσσες για τις οποίες υπάρχει κάποια μη ντετερμινιστική μηχανή Turing που τις αποφασίζει ύστερα από τη χρήση λογαριθμικού αριθμού κελιών της ταινίας εργασίας.

Ορισμός 1.30. Η κλάση **EXPSPACE** αποτελείται από τις γλώσσες που μπορούν να διαγνωστούν σε εκθετικό χώρο από κάποια ντετερμινιστική μηχανή Turing. Με άλλα λόγια,

$$\mathbf{EXPSPACE} = \bigcup_{\mathbf{k} \in \mathbb{N}} \mathbf{SPACE(2^{n^k})}$$

Η κλάση **EXPSPACE** περιέχει όλες τις γλώσσες για τις οποίες υπάρχει κάποια ντετερμινιστική μηχανή Turing που τις αποφασίζει ύστερα από τη χρήση εκθετικού αριθμού κελιών της ταινίας εργασίας.

Ορισμός 1.31. Η κλάση **NEXPSPACE** αποτελείται από τις γλώσσες που μπορούν να διαγνωστούν σε εκθετικό χώρο από κάποια μη ντετερμινιστική μηχανή Turing. Με άλλα λόγια,

$$\mathbf{NEXPSPACE} = \bigcup_{\mathbf{k} \in \mathbb{N}} \mathbf{NSPACE(2^{n^k})}$$

Η κλάση **NEXPSPACE** περιέχει όλες τις γλώσσες για τις οποίες υπάρχει κάποια μη ντετερμινιστική μηχανή Turing που τις αποφασίζει ύστερα από τη χρήση εκθετικού αριθμού κελιών της ταινίας εργασίας.

1.4. Σχέσεις Κλάσεων Πολυπλοκότητας

Ανάμεσα στις κλάσεις πολυπλοκότητας του ίδιου είδους υπάρχει κάποια ιεραρχία. Μερικές απλές σχέσεις που συνδέουν τις κλάσεις πολυπλοκότητας που ορίσαμε στην προηγούμενη ενότητα είναι οι παρακάτω:

$$\mathbf{TIME(f(n))} \subseteq \mathbf{NTIME(f(n))}$$

και

$$\mathbf{SPACE(f(n))} \subseteq \mathbf{NSPACE(f(n))}$$

Κάθε ντετερμινιστική μηχανή μπορεί να θεωρηθεί ως μη ντετερμινιστική όπου σε κάθε βήμα της έχει μία μόνο επιλογή, όπως αναφέραμε σε προηγούμενη ενότητα. Οπότε προκύπτει ότι:

$$\mathbf{P} \subseteq \mathbf{NP}$$

Επίσης μια ντετερμινιστική μηχανή Turing μπορεί να αποφασίσει γλώσσα της **NP** σε εκθετικό χρόνο. Άρα προκύπτει ότι:

$$\mathbf{NP} \subseteq \mathbf{EXPTIME}$$

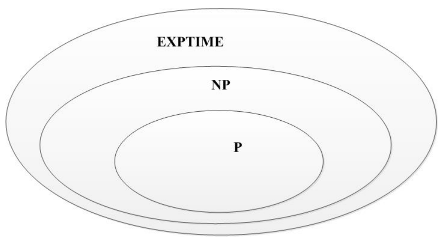
Με βάση τα παραπάνω ισχύει ότι:

$$\mathbf{P} \subseteq \mathbf{EXPTIME}$$

Θεώρημα 1.1. *Έστω t(n) οποιαδήποτε συνάρτηση τέτοια ώστε t(n) $\geq$ n. Για κάθε μη ντετερμινιστική μονοταινιακή μηχανή Turing χρόνου t(n) υπάρχει ισοδύναμη ντετερμινιστική μονοταινιακή μηχανή Turing χρόνου $2^{O(t(n))}$.*

Άρα με βάση το θεώρημα έχουμε ότι:

$$\mathbf{TIME(f(n))} \subseteq \mathbf{TIME(2^{f(n)})}$$



Σχήμα 1.8: Η επικρατούσα άποψη για τις κλάσεις χρονικής πολυπλοκότητας **P, NP, EXPTIME**.

Θεώρημα 1.2 (Θεώρημα του Savitch)**.** *Για οποιαδήποτε χωρικά κατασκευάσιμη*[1] *συνάρτηση f*: $\mathbb{N} \rightarrow \mathbb{N}$ *για την οποία ισχύει ότι f(n) $\geq$ log(n),*

$$\boldsymbol{NSPACE(f(n)) \subseteq SPACE(f^{2}(n))}$$

[1]Ακολουθεί ορισμός στην ενότητα 1.5.

Άρα οποιαδήποτε μη ντετερμινιστική μηχανή Turing η οποία χρησιμοποιεί χώρο f(n) είναι δυνατόν να μετατραπεί σε ισοδύναμη ντετερμινιστική μηχανή Turing η οποία χρησιμοποιεί χώρο μόλις $f^2(n)$.

Με βάση τους ορισμούς των **NPSPACE, PSPACE** και το Θεώρημα 1.2 προκύπτει ότι:

PSPACE = **NPSPACE**

Δηλαδή οι γλώσσες που μπορούν να αποφασιστούν από μη ντετερμινιστική μηχανή Turing σε πολυωνυμικό χώρο μπορούν να αποφασιστούν σε πολυωνυμικό χώρο και από ντετερμινιστική μηχανή Turing.

Υπάρχουν ακόμα κάποιες σχέσεις ανάμεσα σε κλάσεις πολυπλοκότητας διαφορετικού είδους. Αυτές είναι οι εξής:

P $\subseteq$ **PSPACE**

Διότι μια μηχανή με μικρό χρόνο εκτέλεσης δε μπορεί να χρησιμοποιήσει μεγάλη ποσότητα χώρου. Πιο συγκεκριμένα, σε κάθε βήμα του υπολογισμού της μια μηχανή μπορεί να επισκέπτεται το πολύ μια νέα θέση της ταινίας της. Άρα, για κάθε $t(n) \geq n$, οποιαδήποτε μηχανή που λειτουργεί σε χρόνο t(n) χρησιμοποιεί χώρο το πολύ t(n). Για τον ίδιο λόγο έχουμε:

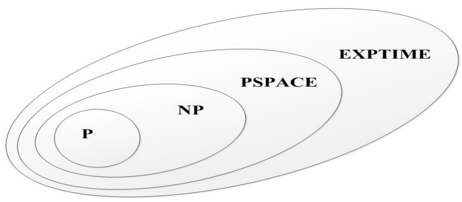
NP $\subseteq$ **NPSPACE**

και

NP $\subseteq$ **PSPACE**

Ακόμα μπορούμε να φράξουμε τη χρονική πολυπλοκότητα μιας μηχανής Turing με βάση τη χωρική της πολυπλοκότητα. Συγκεκριμένα, για κάθε $f(n) \geq n$, μια μηχανή που χρησιμοποιεί χώρο f(n) μπορεί να εμφανίσει στην ταινία $2^{f(n)}$ διαφορετικές συμβολοσειρές, αν υποθέσουμε ότι έχουμε ένα δυαδικό αλφάβητο, δηλαδή $\Sigma = \{0,1\}$ και $|\Sigma| = 2$. Η κεφαλή της μηχανής μπορεί να βρίσκεται σε μία από τις f(n) θέσεις και η ίδια η μηχανή σε μία από τις q διαφορετικές καταστάσεις. Οπότε ο συνολικός αριθμός των συνολικών καταστάσεων της μηχανής Turing είναι $T(n) = qf(n)2^{f(n)}$. Από την άλλη πλευρά, σε κάθε υπολογισμό μιας ντετερμινιστικής μηχανής Turing ο οποίος τερματίζει, όλες οι συνολικές καταστάσεις είναι διαφορετικές. Επομένως ο χρόνος εκτέλεσης μιας μηχανής που χρησιμοποιεί χώρο f(n) δεν μπορεί να υπερβαίνει τα $qf(n)2^{f(n)}$ βήματα. Συνεπώς,

PSPACE $\subseteq$ **EXPTIME**



Σχήμα 1.9: Η επικρατούσα άποψη για τις σχέσεις μεταξύ των κλάσεων **P, NP, PSPACE, EXPTIME**.

1.5. Θεωρήματα Ιεραρχίας

Η κοινή λογική υπαγορεύει ότι εάν αυξήσουμε το χρόνο ή το χώρο που επιτρέπεται να χρησιμοποιήσει μια μηχανή Turing, θα διευρυνθεί και η κλάση των προβλημάτων που μπορεί να επιλύσει. Παραδείγματος χάριν, θα περίμενε κάνεις ότι οι μηχανές Turing είναι σε θέση να διαγνώσουν περισσότερες γλώσσες σε χρόνο n^3 απ' ότι σε χρόνο n^2. Τα θεωρήματα ιεραρχίας αποδεικνύουν ότι η διαισθητική αυτή πρόβλεψη είναι ορθή, αλλά κάτω από ορισμένες προϋποθέσεις τις οποίες και θα περιγράψουμε παρακάτω. Η ονομασία θεωρήματα ιεραρχίας οφείλεται στο γεγονός ότι τα θεωρήματα αυτά αποδεικνύουν πως οι κλάσεις χρονικής και χωρικής πολυπλοκότητας δεν είναι όλες ίδιες μεταξύ τους, αλλά σχηματίζουν μια ιεραρχία, στην οποία οι κλάσεις που αντιστοιχούν σε μεγαλύτερα φράγματα περιλαμβάνουν περισσότερες γλώσσες απ' ότι οι κλάσεις που αντιστοιχούν σε μικρότερα φράγματα.

1.5.1. Χωρική Ιεραρχία

Μας δίνει τη δυνατότητα να ορίσουμε μια ιεραρχία ανάμεσα στις κλάσεις χωρικής πολυπλοκότητας.

Ορισμός 1.32. Έστω f: $\mathbb{N} \longrightarrow \mathbb{N}$ οποιαδήποτε συνάρτηση τέτοια ώστε η τιμή f(n) να είναι τουλάχιστον O(log(n)). Η f λέγεται χωρικά κατασκευάσιμη εάν η συνάρτηση που απεικονίζει τη λέξη 1^n στη δυαδική αναπαράσταση της τιμής f(n) είναι υπολογίσιμη σε χώρο O(f(n)).[1]

Με άλλα λόγια, η f είναι χωρικά κατασκευάσιμη εάν υπάρχει μηχανή Turing χώρου O(f(n)) η οποία κάθε φορά που ξεκινά με είσοδο 1^n τερματίζει έχοντας στην ταινία της τη δυαδική αναπαράσταση της τιμής f(n).

Θεώρημα 1.3 (Θεώρημα της χωρικής ιεραρχίας). *Για οποιαδήποτε χωρικά κατασκευάσιμη συνάρτηση f: $\mathbb{N} \longrightarrow \mathbb{N}$, υπάρχει γλώσσα που αποφασίζεται σε χώρο O(f(n)) αλλά όχι σε χώρο o(f(n)).*

Πόρισμα 1.1. *Έστω δύο συναρτήσεις f_1, $f_2 : \mathbb{N} \longrightarrow \mathbb{N}$. Εάν η $f_1(n)$ είναι $o(f_2(n))$ και η f_2 είναι χωρικά κατασκευάσιμη, τότε* $\textbf{SPACE}(f_1(n)) \subsetneq \textbf{SPACE}(f_2(n))$.

Πόρισμα 1.2. *Για οποιουσδήποτε δύο πραγματικούς αριθμούς $0 \leq \varepsilon_1 < \varepsilon_2$,*

$$\textbf{SPACE}(n^{\varepsilon_1}) \subsetneq \textbf{SPACE}(n^{\varepsilon_2})$$

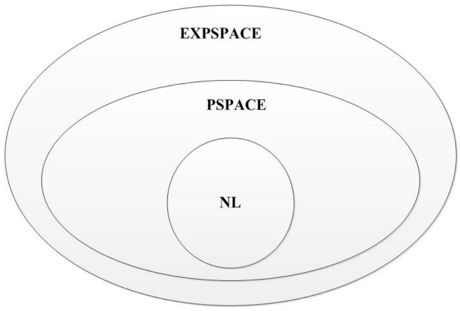
Άμεση συνέπεια των παραπάνω είναι τα παρακάτω πορίσματα.

Πόρισμα 1.3. $\textbf{NL} \subsetneq \textbf{PSPACE}$

Πόρισμα 1.4. $\textbf{PSPACE} \subsetneq \textbf{EXPSPACE}$

Το Πόρισμα 1.4 τεκμηριώνει την ύπαρξη επιλύσιμων προβλημάτων τα οποία είναι δυσεπίλυτα, υπό την έννοια ότι κάθε διαγνωστική διαδικασία για αυτά θα χρησιμοποιεί αναγκαστικά υπερπολυωνυμικό χώρο. Τα προβλήματα αυτά είναι κάπως τεχνητά, δηλαδή παρουσιάζουν ενδιαφέρον μόνο όσον αφορά τον διαχωρισμό των κλάσεων πολυπλοκότητας.

[1] Υπενθυμίζουμε ότι η έκφραση 1^n αναπαριστά μια λέξη από 1 με μήκος n.



Σχήμα 1.10: Η κλάση **NL** είναι γνήσιο υποσύνολο της **PSPACE** που με τη σειρά της είναι γνήσιο υποσύνολο της **EXPSPACE**.

1.5.2. Χρονική Ιεραρχία

Μας δίνει τη δυνατότητα να ορίσουμε μια ιεραρχία ανάμεσα στις κλάσεις χρονικής πολυπλοκότητας.

Ορισμός 1.33. Έστω t: $\mathbb{N} \longrightarrow \mathbb{N}$ οποιαδήποτε συνάρτηση τέτοια ώστε η τιμή t(n) να είναι τουλάχιστον O(nlog(n)). Η t λέγεται χρονικά κατασκευάσιμη εάν η συνάρτηση που απεικονίζει τη λέξη $1^{\mathbf{n}}$ στη δυαδική αναπαράσταση της τιμής t(n) είναι υπολογίσιμη σε χρόνο O(t(n)).

Με άλλα λόγια, η t είναι χρονικά κατασκευάσιμη εάν υπάρχει μηχανή Turing χρόνου O(t(n)) η οποία κάθε φορά που ξεκινά με είσοδο $1^{\mathbf{n}}$ τερματίζει έχοντας στην ταινία της τη δυαδική αναπαράσταση της τιμής t(n).

Θεώρημα 1.4 (Θεώρημα της χρονικής ιεραρχίας)**.** *Για οποιαδήποτε χρονικά κατασκευάσιμη συνάρτηση t*: $\mathbb{N} \longrightarrow \mathbb{N}$*, υπάρχει γλώσσα που αποφασίζεται σε χρόνο O(t(n)) αλλά όχι σε χρόνο o(t(n)/log(t(n))).*

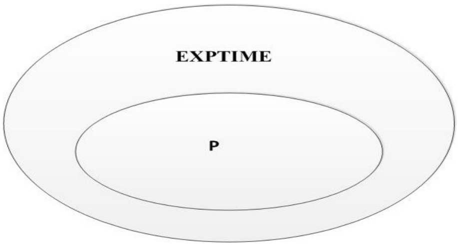
Πόρισμα 1.5. *Έστω δύο συναρτήσεις t_1, t_2* : $\mathbb{N} \longrightarrow \mathbb{N}$*. Εάν η t_1(n) είναι o(t_2(n)/log(t_2(n))) και η t_2 είναι χρονικά κατασκευάσιμη, τότε* ***TIME(t_1(n))*** $\subsetneq$ ***TIME(t_2(n))***.

Πόρισμα 1.6. *Για οποιουσδήποτε δύο πραγματικούς αριθμούς* $1 \leq \varepsilon_1 < \varepsilon_2$,

$$\textit{TIME}(n^{\varepsilon_1}) \subsetneq \textit{TIME}(n^{\varepsilon_2})$$

Άμεση συνέπεια των παραπάνω είναι το παρακάτω πόρισμα.

Πόρισμα 1.7. ***P*** $\subsetneq$ ***EXPTIME***



Σχήμα 1.11: Η κλάση **P** είναι γνήσιο υποσύνολο της κλάσης **EXPTIME**.

1.6. Γραμματικές

Η τυπική θεωρία γλωσσών είναι ένας κλάδος των μαθηματικών ο οποίος χρησιμοποιείται κυρίως στην επιστήμη των υπολογιστών και στη γλωσσολογία με σκοπό να ορίσει τους κανόνες εκείνους με τους οποίους μια συγκεκριμένη (φυσική ή τεχνητή) γλώσσα μπορεί να παραχθεί. Δηλαδή, μια γραμματική περιγράφει τον τρόπο με τον οποίο μπορούν να παραχθούν οι συμβολοσειρές που ανήκουν σε μια γλώσσα. Μια γραμματική δεν περιγράφει τη σημασιολογία, δηλαδή το τι σημαίνουν αυτές οι λέξεις. Ο πρώτος τυπικός ορισμός για τις γραμματικές δόθηκε στη δεκαετία του 1950 από το γλωσσολόγο Νόαμ Τσόμσκι.

Ορισμός 1.34. Μια τυπική γραμματική G είναι μια τετράδα (V, Σ, S, R) όπου:

- V σύνολο μη τερματικών συμβόλων.
- Σ σύνολο τερματικών συμβόλων, τέτοιο ώστε $\Sigma \cap V = \emptyset$
- S μοναδικό αρχικό σύμβολο.
- R σύνολο κανόνων παραγωγής της μορφής $a \longrightarrow b$, όπου a, b συμβολοσειρές που περιέχουν τερματικά και μη τερματικά σύμβολα.

Η λειτουργία μιας γραμματικής μπορεί να οριστεί με βάση τις παρακάτω σχέσεις πάνω στις συμβολοσειρές.

Ορισμός 1.35. Έστω μια γραμματική $G = (V, \Sigma, S, R)$ τότε:

- η δυαδική σχέση $\Rightarrow_{\mathbf{G}}$ ορίζεται πάνω σε συμβολοσειρές από το σύνολο $(V \cup \Sigma)^*$ ως εξής:
  - $x \Rightarrow_{\mathbf{G}} y$ αν και μόνο αν $\exists$ u, v, w $\in \Sigma^*$ και $X \in V$:
    έτσι ώστε $x = uXv$, $y = uwv$, και $X \longrightarrow w \in R$.
- η σχέση $\Rightarrow^*_{\mathbf{G}}$ είναι η ανακλαστική μεταβατική κλειστότητα της $\Rightarrow_{\mathbf{G}}$.

1.6.1. Ιεραρχία Τσόμσκι

Ο Νόαμ Τσόμσκι όρισε τις τυπικές γραμματικές το 1956 και τις ταξινόμησε σύμφωνα με τον τύπο των κανόνων παραγωγής τους. Οι γραμματικές κατατάσσονται με αυξανόμενους περιορισμούς, δηλαδή ισχύει ότι:

Γραμ.Τύπου 3 $\subset$ Γραμ.Τύπου 2 $\subset$ Γραμ.Τύπου 1 $\subset$ Γραμ.Τύπου 0

Γραμματικές Τύπου 0 ή Γραμματικές Χωρίς Περιορισμούς: αυτές περιέχουν όλες τις τυπικές γραμματικές. Παράγουν όλες τις γλώσσες που μπορούν να αναγνωριστούν από μια μηχανή Turing. Οι γλώσσες αυτές είναι επίσης γνωστές ως αναδρομικά απαριθμήσιμες γλώσσες. Αυτές είναι διαφορετικές από τις αναδρομικές γλώσσες που αποφασίζονται από μηχανές Turing που πάντοτε τερματίζουν τη λειτουργία τους.

Γραμματικές Τύπου 1 ή Γραμματικές Με Συμφραζόμενα: οι γραμματικές αυτές παράγουν τις γλώσσες με συμφραζόμενα. Έχουν κανόνες της μορφής $\alpha A \beta \longrightarrow \alpha\gamma\beta$ όπου A ένα μη τερματικό σύμβολο της γραμματικής, ενώ α,β,γ είναι συμβολοσειρές τερματικών και μη τερματικών συμβόλων με τις α,β να μπορεί να είναι κενές αλλά η γ να είναι απαραίτητα μη κενή. Ο κανόνας S $\longrightarrow$ e επιτρέπεται αν το S δεν εμφανίζεται στο δεξί μέλος κανενός κανόνα του συνόλου κανόνων R. Οι γλώσσες που περιγράφονται από αυτές τις γραμματικές είναι ακριβώς οι ίδιες γλώσσες που αναγνωρίζονται από ένα γραμμικά φραγμένο αυτόματο, δηλαδή από μια μη ντετερμινιστική μηχανή Turing της οποίας η ταινία φράσσεται από το μέγεθος της συμβολοσειράς εισόδου.

Γραμματικές Τύπου 2 ή Γραμματικές Χωρίς Συμφραζόμενα: οι γραμματικές αυτές παράγουν τις γλώσσες χωρίς συμφραζόμενα. Οι κανόνες που περιέχουν οι γραμματικές αυτές είναι της μορφής A $\longrightarrow$ γ όπου A ένα μη τερματικό σύμβολο ενώ γ μια συμβολοσειρά τερματικών και μη τερματικών συμβόλων, ενδεχομένως κενή. Αυτές οι γλώσσες είναι ακριβώς οι γλώσσες που μπορούν να αναγνωριστούν από ένα μη ντετερμινιστικό αυτόματο εφοδιασμένο με μια στοίβα. Οι γλώσσες χωρίς συμφραζόμενα είναι το θεωρητικό υπόβαθρο για το συντακτικό σχεδόν όλων των σύγχρονων γλωσσών προγραμματισμού κυρίως επειδή υπάρχουν πολλοί γνωστοί και σχετικά γρήγοροι αλγόριθμοι για τη συντακτική ανάλυση ενός προγράμματος γραμμένου με κανόνες γραμματικών χωρίς συμφραζόμενα.

Γραμματικές Τύπου 3 ή Κανονικές Γραμματικές: οι γραμματικές αυτές παράγουν τις κανονικές γλώσσες. Τέτοιες γραμματικές περιορίζουν τους κανόνες στο να περιέχουν ένα απλό μη τερματικό σύμβολο στο αριστερό μέλος ενός κανόνα που αντικαθίσταται από ένα απλό τερματικό σύμβολο ενδεχομένως ακολουθούμενο από ένα απλό μη τερματικό σύμβολο. Ο κανόνας S $\longrightarrow$ e επιτρέπεται σε αυτές τις γραμματικές αν το S δεν εμφανίζεται στο δεξί μέλος κανενός κανόνα, τέτοιες γραμματικές δηλαδή μπορούν να δημιουργήσουν την κενή συμβολοσειρά. Αυτές οι γλώσσες, δηλαδή οι κανονικές, αποφασίζονται (και όχι απλά αναγνωρίζονται όπως πριν) από πεπερασμένα αυτόματα. Επιπλέον, αυτή η οικογένεια τυπικών γλωσσών μπορεί να αναπαρασταθεί και από κανονικές εκφράσεις. Οι κανονικές γλώσσες γενικά χρησιμοποιούνται για να οριστούν λεκτικά πρότυπα στη διαδικασία της λεκτικής ανάλυσης γλωσσών προγραμματισμού.

1.6.2. Γραμματικές και Γλώσσες με Συμφραζόμενα

Μια γραμματική με συμφραζόμενα (Context Sensitive Grammar) είναι μια τυπική γραμματική όπου όλοι οι κανόνες παραγωγής είναι της μορφής aAb $\longrightarrow$ acb, το οποίο διαισθητικά σημαίνει ότι όποτε έχουμε το σύμβολο A με συμφραζόμενα a, b τότε μπορούμε να το αντικαταστήσουμε με τη (μη κενή) συμβολοσειρά τερματικών και μη τερματικών συμβόλων c.

Η έννοια της γραμματικής με συμφραζόμενα ως παραγωγού γλωσσών, παρουσιάστηκε για πρώτη φορά από τον Νόαμ Τσόμσκι το 1956 ως ένας τρόπος περιγραφής του συντακτικού φυσικών γλωσσών στις οποίες πράγματι είναι συχνή η περίπτωση όπου η παρουσία ή όχι μιας λέξης σε συγκεκριμένο σημείο εντός μιας πρότασης, εξαρτάται από τις γειτονικές λέξεις, τα συμφραζόμενα. Μια γλώσσα που περιγράφεται από μια γραμματική με συμφραζόμενα, είναι γλώσσα με συμφραζόμενα.

Ορισμός 1.36. Μια γραμματική G με συμφραζόμενα είναι μια τετράδα (V, Σ, S, R) όπου:

- V το σύνολο των μη τερματικών συμβόλων
- Σ το σύνολο των τερματικών συμβόλων (το αλφάβητο της γλώσσας που παράγει η δεδομένη γραμματική)
- S το αρχικό σύμβολο της γραμματικής, S $\in$ V

- R το σύνολο των κανόνων παραγωγής: $R \subseteq (V \cup \Sigma)^* V (V \cup \Sigma)^* \times (V \cup \Sigma)^*$, όπου ο κάθε κανόνας παραγωγής του R είναι της μορφής $\alpha A \beta \longrightarrow \alpha \gamma \beta$ με $\alpha, \beta \in (V \cup \Sigma)^*$, $A \in V$ και $\gamma \in (V \cup \Sigma)^+$, δηλαδή το A είναι ένα απλό μη τερματικό σύμβολο, τα α και β είναι συμβολοσειρές (ενδεχομένως κενές) τερματικών και μη τερματικών συμβόλων ενώ γ είναι μια μη κενή συμβολοσειρά τερματικών και μη τερματικών συμβόλων.

Επιπλέον επιτρέπεται και κανόνας της μορφής $S \longrightarrow e$, όπου e είναι η κενή συμβολοσειρά, δεδομένου ότι το S δεν εμφανίζεται στο δεξί μέλος κανενός κανόνα παραγωγής.

Τέλος η γλώσσα της γραμματικής G, η οποία συμβολίζεται ως L(G), ορίζεται ως όλες εκείνες οι συμβολοσειρές από το αλφάβητο Σ που μπορούν να παραχθούν ξεκινώντας από το αρχικό σύμβολο S της γραμματικής και εφαρμόζοντας τους κανόνες παραγωγής του R μέχρι να καταλήξουμε σε μία συμβολοσειρά η οποία να μην περιέχει καθόλου μη τερματικά σύμβολα. Πιο τυπικά ισχύει ότι $L(G) = \{w \in \Sigma^* \mid S \Rightarrow_G^* w\}$ και η L(G) ονομάζεται γλώσσα με συμφραζόμενα (Context Sensitive Language).

Παρακάτω θα αποδείξουμε ότι η κλάση των γλωσσών με συμφραζόμενα είναι κλειστή ως προς την πράξη του συμπληρώματος. Δηλαδή αν η L είναι γλώσσα με συμφραζόμενα τότε και η $\overline{L}$ (το συμπλήρωμα της L) είναι γλώσσα με συμφραζόμενα.

2. ΕΞΗΓΗΣΗ ΑΠΟΔΕΙΞΗΣ

2.1. Ο Μη Ντετερμινιστικός Χώρος Είναι Κλειστός ως Προς το Συμπλήρωμα

O Neil Immerman στη δημοσίευση του [23], που θα εξηγήσουμε παρακάτω, μας δείχνει ότι ο μη ντετερμινιστικός χώρος s(n) είναι κλειστός ως προς το συμπλήρωμα για κάποια συνάρτηση s(n) $\geq$ log(n). Με άλλα λόγια αυτό που λέει το θεώρημα που απέδειξε είναι ότι αν μια μη ντετερμινιστική μηχανή Turing μπορεί να λύσει ένα πρόβλημα, τότε μπορεί να λύσει και το συμπληρωματικό του πρόβλημα χρησιμοποιώντας το ίδιο ασυμπτωτικό ποσό μνήμης. Το ίδιο αποτέλεσμα με το [23] απέδειξε ανεξάρτητα ο Róbert Szelepcsényi [24].

Από αυτό προκύπτει άμεσα ότι το σύνολο των γλωσσών με συμφραζόμενα (**CSL**) είναι κλειστό ως προς την πράξη του συμπληρώματος, έτσι απαντήθηκε ένα ερώτημα που είχε θέσει ο Sige-Yuki Kuroda το 1964 [9]. Μπορείτε να δείτε Hartmanis και Hunt [4] για μια συζήτηση γύρω από την ιστορία και τη σημασία του προβλήματος που έθεσε ο Kuroda και Hopcroft και Ullman [5] για το σχετικό υπόβαθρο και τους απαραίτητους ορισμούς.

Η ιστορία πίσω από την απόδειξη είναι η ακόλουθη. Ο Immerman έχει ασχοληθεί αρκετά με τη λογική πρώτης τάξης (First-Order Logic), που είναι μια λογική πιο εκφραστική από την προτασιακή λογική και καλύπτει ένα μεγάλο μέρος των απαιτήσεων που έχουμε από μία γλώσσα αναπαράστασης γνώσης [22]. Το 1981 έδειξε ότι το σύνολο των επαγωγικών ορισμών πρώτης τάξης επί πεπερασμένης δομής είναι κλειστό ως προς το συμπλήρωμα [6]. Αυτό ισχύει με ή χωρίς σχέση διάταξης πάνω στη δομή. Αν η διάταξη είναι παρούσα τότε η επιστρεφόμενη κλάση είναι η **P**. Αρκετοί ανέμεναν ότι το αποτέλεσμα ήταν λάθος λόγω απουσίας της διάταξης.

Το 1983 μελέτησε τη λογική πρώτης τάξης, με διάταξη και με τελεστή μεταβατικής κλειστότητας (Transitive Closure). Ένας ορισμός για τον TC είναι ο εξής: Κάθε σύνολο A είναι μέλος μεταβατικού συνόλου M, και μάλιστα υπάρχει ελάχιστο μεταβατικό σύνολο M $=$ TC(A) τέτοιο ώστε A $\in$ TC(A). Το TC(A) είναι η μεταβατική κλειστότητα του A. O Immerman έδειξε ότι η **NSPACE[log(n)]** είναι ίση με (FO $+$ pos TC), δηλαδή λογική πρώτης τάξης με διάταξη και πράξη μεταβατικής κλειστότητας στην οποία ο τελεστής μεταβατικής κλειστότητας δεν εμφανίζει τυχόν αρνητικά σύμβολα [7].

Δηλαδή το (FO $+$ pos TC) είναι ένα σύνολο ιδιοτήτων, εκφρασμένες μέσω της λογικής πρώτης τάξης και του τελεστή μεταβατικής κλειστότητας (TC) που δεν περιέχει κανένα αρνητικό σύμβολο [7].

O Immerman έχει δείξει ότι η κλάση (FO $+$ pos TC) είναι κλειστή ως προς το συμπλήρωμα. Τα κύρια αποτελέσματα της δημοσίευσης ακολουθούν. Δίνει την απόδειξη από την άποψη των μηχανών και μετά ακολουθούν τα αποτελέσματα για τη μεταβατική κλειστότητα στο πόρισμα 2.3. Το ερώτημα αν η κλάση (FO $+$ pos TC) χωρίς σχέση διάταξης είναι κλειστή ως προς το συμπλήρωμα παραμένει ανοικτό.

Η εργασία του στην εκφρασιμότητα πρώτης τάξης οδήγησε στην απόδειξη ότι ο μη ντετερμινιστικός χώρος είναι κλειστός ως προς το συμπλήρωμα. Ωστόσο, επειδή οι εκφραστικές κλάσεις πρώτης τάξης δεν είναι άμεσα σχετικές με τις αποδείξεις σε αυτή τη δημοσίευση παραλείπονται αυτοί οι ορισμοί. Για κάποιον που ενδιαφέρεται μπορεί να ανατρέξει στο [7] για όλους αυτούς τους ορισμούς.

Σημείωση ότι η απόδειξη του θεωρήματος 3.3 στο [7] είναι πιο πολύπλοκη από την απόδειξη του θεωρήματος 2.1 που ακολουθεί, αλλά αρκετά παρόμοια με αυτό. Το ίδιο ισχύει για την απόδειξη στο [6], ότι οι επαγωγικοί τύποι πρώτης τάξης είναι κλειστοί ως προς το συμπλήρωμα.

Το 1995, η επιτροπή απονομής του βραβείου Gödel επέλεξε δύο αποδέκτες. Το βραβείο απονεμήθηκε για δύο δημοσιεύσεις του 1988, του Neil Immerman [23] της οποίας τα αποτελέσματα παρουσιάζουμε παρακάτω και του Róbert Szelepcsényi [24]. Έτσι απαντήθηκε ένα ερώτημα που παρέμενε ανοικτό για πάνω από 3 δεκαετίες. Κάποτε, ο Neil Immerman, δήλωσε πως η βασική ιδέα για την απόδειξη του ήρθε ενώ είχε βγάλει βόλτα το σκύλο του.

2.1.1. Αποτελέσματα

Θεώρημα 2.1. *Για οποιαδήποτε συνάρτηση s(n) $\geq$ log(n),*

$$\textbf{\textit{NSPACE[s(n)]}} = \textbf{\textit{coNSPACE[s(n)]}}$$

Απόδειξη. Το αποδεικνύουμε αυτό με δύο λήμματα. Θα δείξουμε ότι ο υπολογισμός του ακριβή αριθμού συνολικών καταστάσεων μιας **NSPACE[s(n)]** μηχανής μπορεί να γίνει σε **NSPACE[s(n)]** (Λήμμα 2.2). Το Λήμμα 2.1 λέει ότι άπαξ και έχει υπολογιστεί αυτός ο αριθμός μπορούμε να διακρίνουμε την απόρριψη όπως επίσης και την αποδοχή μιας μηχανής Turing.

■

Λήμμα 2.1. *Ας υποθέσουμε ότι μας δίνεται μια **NSPACE[s(n)]** μηχανή M, μια αρχική συνολική κατάσταση μεγέθους s(n), η START, και ο ακριβής αριθμός N των συνολικών καταστάσεων μεγέθους s(n) στις οποίες μπορεί να φτάσει η μηχανή M ξεκινώντας από τη START. Τότε εμείς μπορούμε να εξετάσουμε σε **NSPACE[s(n)]** αν η M απορρίπτει.*

Απόδειξη. Έστω M μια μη ντετερμινιστική Μηχανή Turing (M.N.M.T) χώρου s(n). Θα φτιάξουμε μια M.N.M.T Tester τέτοια ώστε για όλες τις εισόδους w, η Tester(w) δέχεται αν και μόνο αν η M(w) απορρίπτει.

Σύμφωνα με το Λήμμα 1.1 αν w μια είσοδος της M και s(|w|) $=$ s τότε ο συνολικός αριθμός των συνολικών καταστάσεων της M(w) μπορεί να είναι το πολύ c^s για κάποια σταθερά c. Έστω τώρα ότι t $=$ c^s οπότε ο χρόνος εκτέλεσης της M(w) μπορεί να είναι το πολύ t, δηλαδή μπορούμε να τον φράξουμε. Αν έχουμε υπολογιστικό μονοπάτι μήκους μεγαλύτερο από t, αυτό θα σήμαινε ότι κάποια συνολική κατάσταση επαναλαμβάνεται, συνεπώς θα μπορούσαμε να το μικρύνουμε.

Έστω START η αρχική συνολική κατάσταση της M(w) μεγέθους s. Έστω N ο αριθμός των συνολικών καταστάσεων που είναι προσβάσιμες από τη START. Υποθέτουμε ότι ξέρουμε το N. Θα δείξουμε πως η Tester(w) μπορεί να καθορίσει σωστά
ότι η M(w) δεν δέχεται.
Η M.N.M.T Tester κάνει τα εξής:

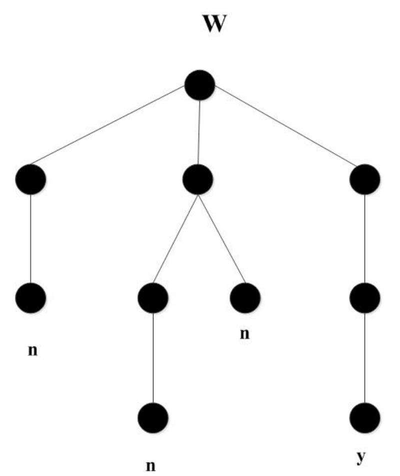
```
Let counter = 0.
For all nonaccepting configurations C of M(w):
      Try to guess a computation path from START to C at most length L.
      IF (found) then:
           Let counter = counter + 1.
IF (counter = N) then:
      Accept.
Otherwise:
      Reject.
```

Η Tester παίρνει μια μια τις συνολικές καταστάσεις μη αποδοχής C με λεξικογραφική σειρά. Για κάθε τέτοια συνολική κατάσταση προσπαθεί να μαντέψει ένα μονοπάτι από τη START προς τη C μήκους το πολύ L. Σύμφωνα με όσα αναφέρουμε παραπάνω το $L = c^s$, δηλαδή αποτελεί ένα άνω φράγμα στο πλήθος των συνολικών καταστάσεων της μηχανής M. Αν λοιπόν βρει μονοπάτι αυξάνει το μετρητή και προχωρά στην επόμενη κατάσταση. Αν βρει μονοπάτι για όλες τις συνολικές καταστάσεις μη αποδοχής - τις N - τότε δέχεται την είσοδο. Η Tester απορρίπτει σε περίπτωση που υπάρχει έστω και μια συνολική κατάσταση αποδοχής ή σε περίπτωση που σε κάποιο υπολογισμό δε βρει μονοπάτι για κάποια από τις N συνολικές καταστάσεις μη αποδοχής.

Αν η M(w) δέχεται τότε υπάρχει μια συνολική κατάσταση αποδοχής προσβάσιμη από τη START, οπότε πρέπει να υπάρχουν λιγότερες από N συνολικές καταστάσεις μη αποδοχής προσβάσιμες από τη START άρα η Tester(w) απορρίπτει. Από την άλλη μεριά, αν η M(w) απορρίπτει τότε δεν υπάρχει καμία συνολική κατάσταση αποδοχής προσβάσιμη από τη START. Άρα η Tester(w) θα βρει N μονοπάτια μη αποδοχής και θα δεχτεί.

Ο συνολικός χώρος που θα χρησιμοποιήσει η Tester είναι το πολύ O(s(n)) αφού ψάχνουμε μόνο μια συνολική κατάσταση τη φορά.

■



Σχήμα 2.1: Η Μ.Ν.Μ.Τ M αποδέχεται τη w. Τότε η Tester του Λήμματος 2.1 την απορρίπτει.

Λήμμα 2.2. *Δοθείσης της αρχικής συνολικής κατάστασης START, όπως και στο Λήμμα 2.1, μπορούμε να υπολογίσουμε το N - τον αριθμό των συνολικών καταστάσεων μεγέθους s(n) όπου μπορεί να φτάσει η μηχανή M ξεκινώντας από την κατάσταση START - σε* ***NSPACE[s(n)]****.*

Απόδειξη. Έστω N_d ο αριθμός των συνολικών καταστάσεων που είναι προσβάσιμες από τη START σε το πολύ d βήματα. Θα δείξουμε πως μπορούμε να υπολογίσουμε το N_{d+1} από το N_d. Θα ξεκινήσουμε με το N_0, θα υπολογίσουμε το N_1, μετά το N_2 και ούτω καθεξής. Με επαγωγή στο d, δηλαδή στο πλήθος των βημάτων, θα δείξουμε ότι κάθε N_d μπορεί να υπολογιστεί σε **NSPACE[s(n)]**.

Βάση της επαγωγής. Για $d = 0$ είναι προφανές ότι $N_0 = 1$, δηλαδή σε 0 βήματα από τη START η μηχανή μπορεί να προσεγγίσει τη START.

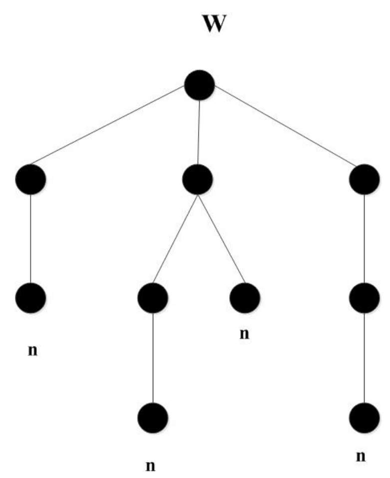
Επαγωγική υπόθεση. Υποθέτουμε ότι γνωρίζουμε το N_d, δηλαδή το πλήθος των συνολικών καταστάσεων που είναι προσβάσιμες από τη START σε το πολύ d βήματα.

Επαγωγικό βήμα. Θα δείξουμε ότι από το N_d μπορούμε να υπολογίσουμε το N_{d+1}. Ακολουθεί ο αλγόριθμος που μη ντετερμινιστικά υπολογίζει το N_{d+1} από το N_d :

```
Let Nd+1 = 0.
For all configurations C in lexicographical order:
      Let flag = 0.
      Let counter = 0.
      For all configurations D in lexicographical order:
           Guess a path from START to D in at most d steps.
           IF (found) then:
                Let counter = counter + 1.
                IF (D = C or D goes to C in 1 step) then:
                     flag = 1.
      IF (counter < Nd) then:
           Halt and Reject.
      Let Nd+1 = Nd+1 + flag.
```

Όπως και στο Λήμμα 2.1 , έτσι και εδώ έχουμε ένα μετρητή που μετρά το πλήθος των συνολικών καταστάσεων που μπορούμε να φτάσουμε σε το πολύ d+1 βήματα, τον N_{d+1}. Μετά ο αλγόριθμος παίρνει μια μια τις συνολικές καταστάσεις C με λεξικογραφική σειρά. Για κάθε τέτοια συνολική κατάσταση κάνει τα εξής, ξαναπαίρνει πάλι μια μια τις συνολικές καταστάσεις D με λεξικογραφική σειρά και προσπαθεί να βρει για κάθε τέτοια μονοπάτι μήκους το πολύ d ξεκινώντας από τη START. Σε ένα μετρητή, τον counter, κρατάει το πλήθος των μονοπατιών μήκους το πολύ d που βρήκε. Για κάθε μονοπάτι που βρίσκει μήκους το πολύ d ξεκινώντας από τη START, ελέγχει αν η συνολική κατάσταση D είναι ίση με τη συνολική κατάσταση C, για την οποία ψάχνουμε να βρούμε μονοπάτι μήκους το πολύ d+1 από τη START ή αν μπορούμε σε ένα βήμα από τη D να φτάσουμε τη C. Αν ένα από τα δύο είναι αληθές αυτό σημαίνει ότι βρήκαμε μονοπάτι μήκους το πολύ d+1 από τη START προς τη C. Η συνθήκη counter $< N_d$ μας εγγυάται ότι έχουμε εξετάσει όλες τις συνολικές καταστάσεις D που είναι προσβάσιμες από τη START σε το πολύ d βήματα. Αν η συνθήκη είναι ψευδής κάθε φορά, τότε εμείς έχουμε υπολογίσει σωστά το flag που θα είναι ίσο με 1 σε περίπτωση που η C είναι προσβάσιμη από τη START σε το πολύ d+1 βήματα αλλιώς θα είναι ίσο με το 0. Αφού το N (ο αριθμός των συνολικών καταστάσεων μεγέθους s(n)) φράσσεται από το $c^{s(n)}$ για κάποια σταθερά c, όλος ο αλγόριθμος μπορεί να τρέξει σε χώρο O(s(n)).

■



Σχήμα 2.2: Η Μ.Ν.Μ.Τ Μ απορρίπτει τη w. Τότε η Tester του Λήμματος 2.1 την αποδέχεται.

Παρατήρηση. Στην αρχική δήλωση του Θεωρήματος 2.1 ο Neil Immerman έκανε την παραδοχή ότι η συνάρτηση s(n) είναι χωρικά κατασκευάσιμη (Space Constructible). Ωστόσο, ο καθιερωμένος ορισμός της μη ντετερμινιστικής μηχανής Turing που έχει πολυπλοκότητα χώρου s(n) λέει σε ένα σημείο το εξής, ”... κανένας υπολογισμός της μηχανής δε μπορεί να χρησιμοποιήσει περισσότερα από s(n) κελιά ...,” [5]. Έτσι, η παραπάνω απόδειξη δουλεύει ακόμη και αν η s(n) δεν είναι χωρικά κατασκευάσιμη [23]. Απλά αφήνουμε την s(n) να αυξάνει όπου χρειάζεται.

Πόρισμα 2.1. *Η κλάση των γλωσσών με συμφραζόμενα (**CSL**) είναι κλειστή ως προς το συμπλήρωμα.*

Απόδειξη. O Sige-Yuki Kuroda έδειξε το 1964 ότι **CSL** = **NSPACE[n]** [9].

■

Άρα αν μια γλώσσα $L \in$ **CSL** τότε με βάση το Πόρισμα 2.1 και το Θεώρημα 2.1 προκύπτει ότι και η γλώσσα $\overline{L} \in$ **CSL**.

Το k^{th} επίπεδο ιεραρχίας εναλλασσόμενου λογαριθμικού χώρου (Σ_k**ALOG**) έχει οριστεί να είναι το σύνολο των προβλημάτων που γίνονται αποδεκτά από μια εναλλασσόμενη (Alternating) [19] μηχανή Turing λογαριθμικού χώρου που κάνει το πολύ $k-1$ εναλλαγές και ξεκινά από μια υπαρξιακή κατάσταση. Οι Lange, Jenner και Kirsig [10] έδειξαν ότι αυτή η ιεραρχία κατέρρευσε στο δεύτερο επίπεδο, Σ_2**ALOG**. Το αποτέλεσμα αυτό επεκτάθηκε στη συνέχεια από αρκετούς συγγραφείς [2, 14] που έδειξαν ότι η χρησμοληπτική (Oracle) ιεραρχία λογαριθμικού χώρου καταρρέει στο $\mathbf{L}^{\mathbf{NL}}$. Εδώ $\mathbf{L} = \mathbf{SPACE[log(n)]}$ και $\mathbf{NL} = \mathbf{NSPACE[log(n)]}$. Η χρησμοληπτική ιεραρχία λογαριθμικού χώρου δίνεται από $\Sigma_1\mathbf{OLOG} = \mathbf{NL}$ και $\Sigma_{k+1}\mathbf{OLOG} = \mathbf{NL}^{\Sigma_k\mathbf{OLOG}}$. Στην περίπτωση της ιεραρχίας πολυωνυμικού χρόνου, η χρησμοληπτική και η εναλλασσόμενη ιεραρχία είναι ταυτόσημες, αλλά εμφανίστηκαν να είναι διαφορετικές στην περίπτωση του λογαριθμικού χώρου. Γνωρίζουμε ότι η χρησμοληπτική ιεραρχία λογαριθμικού χώρου είναι ίση με (FO + TC). Αυτό μαζί με τα παραπάνω αποτελέσματα, μας οδηγεί στο Θεώρημα 2.1. Άμεσα προκύπτει το παρακάτω πόρισμα.

Πόρισμα 2.2. *Η εναλλασσόμενη ιεραρχία λογαριθμικού χώρου και η χρησμοληπτική ιεραρχία λογαριθμικού χώρου και οι δύο καταρρέουν στο **NSPACE[log(n)]**.*

O Neil Immerman στο [7] έδειξε ότι η κλάση **NL** είναι ίση με (FO + pos TC). Στο θεώρημα 3.3 του [7] έδειξε ότι οποιοδήποτε πρόβλημα της **NL** μπορεί να εκφραστεί με την μορφή

$TC[\varphi](\overline{0}, \overline{max})$ όπου φ είναι τύπος πρώτης τάξης χωρίς ποσοδείκτες, με 0 και max να είναι σταθερά σύμβολα. Προκύπτει ότι το ίδιο ισχύει και για την κλάση $(FO + TC)$.

Πόρισμα 2.3. *1.* ***NSPACE[log(n)]*** *$= (FO + pos\ TC) = (FO + TC)$.*

2. Οποιοσδήποτε τύπος της $(FO + TC)$ μπορεί να εκφραστεί με την μορφή $TC[\varphi](\overline{0}, \overline{max})$ όπου φ είναι τύπος πρώτης τάξης χωρίς ποσοδείκτες.

Ο Michael Fischer έχει παρατηρήσει ότι μιας και τώρα μπορούμε να κάνουμε διαγωνιοποίηση στο μη ντετερμινιστικό χώρο είναι εύκολο να αποδείξουμε ένα θεώρημα σφικτής ιεραρχίας για το μη ντετερμινιστικό χώρο. Αν και το Πόρισμα 2.4 δεν είναι καινούριο, η τεχνική του Neil Immerman δίνει μια πιο απλή και κομψή απόδειξη από αυτή που ήταν προηγουμένως γνωστή. (Δες το κεφάλαιο 12 του [5] για την παλιά απόδειξη.)

Πόρισμα 2.4. *Για οποιαδήποτε χωρικά κατασκευάσιμη συνάρτηση $s(n) \geq log(n)$, με*

$$\lim_{n \to \infty} \frac{t(n)}{s(n)} = 0$$

συνεπάγεται ότι :

$$\textbf{NSPACE[t(n)]} \neq \textbf{NSPACE[s(n)]}.$$

Αν η $t(n) \in o(s(n))$ και η $s(n) \geq log(n)$ είναι χωρικά κατασκευάσιμη τότε ισχύει ότι **NSPACE[t(n)]** $\subset$ **NSPACE[s(n)]**, δηλαδή το **NSPACE[t(n)]** είναι γνήσιο υποσύνολο του **NSPACE[s(n)]**.

3. ΣΥΜΠΕΡΑΣΜΑΤΑ

3.1. Συμπεράσματα

Τα πιο ενδιαφέροντα ερωτήματα σχετικά με τη δύναμη του μη ντετερμινισμού παραμένουν ανοικτά. Ακόμα δε γνωρίζουμε αν ο μη ντετερμινιστικός χώρος είναι ίσος με το ντετερμινιστικό χώρο, δηλαδή αν **NSPACE[f(n)]** $=$ **SPACE[f(n)]**. Επίσης είναι ενδιαφέρον εάν η αποδεικτική μέθοδος που χρησιμοποίησε ο Neil Immerman μπορεί να επεκταθεί για να απαντήσει το παραπάνω ερώτημα ή αν μπορεί να μας πει οτιδήποτε νέο για το μη ντετερμινιστικό χρόνο.

Ο Neil Immerman στη δημοσίευση του 1988 που εξηγήσαμε παραπάνω είχε θέσει τέσσερα ανοικτά προβλήματα, ένα εξ αυτών ήταν το εξής: Είναι ο συμμετρικός λογαριθμικός χώρος (Symmetric LogSpace) κλειστός ως προς το συμπλήρωμα; Δηλαδή ισχύει η σχέση, **SL** $=$ **coSL**; Η κλάση **SL** ορίστηκε από τους Lewis και Papadimitriou το 1982 [11]. Ένας ορισμός που μπορούμε να δώσουμε για αυτή την κλάση είναι ο εξής:

- Περιέχει γλώσσες που μπορούν να αποφασιστούν από συμμετρική μη ντετερμινιστική μηχανή Turing λογαριθμικού χώρου.

Επτά χρόνια μετά, το 1995, έρχονται οι Noam Nisan και Amnon Ta-Shma να αποδείξουν ότι **SL** $=$ **coSL**, δηλαδή ο συμμετρικός λογαριθμικός χώρος είναι κλειστός ως προς το συμπλήρωμα [18].

Τα υπόλοιπα τρία προβλήματα που ανέφερε ήταν τα εξής:

1. Είναι η κλάση **NL** ίση με τη κλάση **L** ή με την **SPACE[log^2(n)]** ;
2. Είναι η κλάση (FO without $\leq +$ pos TC) κλειστή ως προς το συμπλήρωμα;
3. Στην απόδειξη του Θεωρήματος 2.1, ο Immerman έκανε χρήση του θεωρήματος σύμπτυξης χώρου, Θεώρημα 12.1 στο [5]. Η κατασκευή του Neil Immerman πολλαπλασιάζει το χωρικό φράγμα με οκτώ. Είναι ενδιαφέρον να ρωτήσουμε πόσο μπορεί αυτό να μειωθεί.

3.2. Μελλοντική Εργασία

Υπάρχει μια τεράστια γκάμα από άλυτα προβλήματα στον τομέα της θεωρίας υπολογισμού και πολυπλοκότητας. Κάποια από αυτά παραμένουν ανοικτά εδώ και αρκετές δεκαετίες. Παρακάτω γίνεται μια αναφορά σε μερικά από αυτά.

3.2.1. Προβλήματα Προς Επίλυση

Μερικά ανοικτά προβλήματα που προτείνουμε, προς κάθε ενδιαφερόμενο, είναι τα εξής:

1. Είναι η κλάση **NL** ίση με τη κλάση **L** ή με την **SPACE[log^2(n)]**;
2. Είναι η κλάση **NP** ίση με τη κλάση **coNP**;
3. Είναι η κλάση **SPACE** ίση με τη κλάση **NSPACE**;
4. Είναι η κλάση **P** ίση με τη κλάση **NP**;

ΠΙΝΑΚΑΣ ΟΡΟΛΟΓΙΑΣ

Ξενόγλωσσος Όρος	**Ελληνικός Όρος**
Alternating	Εναλλασσόμενη
Context Sensitive Grammar	Γραμματική Με Συμφραζόμενα
Context Sensitive Language	Γλώσσα Με Συμφραζόμενα
First-Order Logic	Λογική Πρώτης Τάξης
Oracle	Χρησμοληπτική
Space Complexity	Χωρική Πολυπλοκότητα
Space Constructible	Χωρικά Κατασκευάσιμη
Symmetric LogSpace	Συμμετρικός Λογαριθμικός Χώρος
Time Complexity	Χρονική Πολυπλοκότητα
Transitive Closure	Μεταβατική Κλειστότητα

ΣΥΝΤΜΗΣΕΙΣ - ΑΡΚΤΙΚΟΛΕΞΑ - ΑΚΡΩΝΥΜΙΑ

ALOG	Alternating Logarithmic
CSL	Context Sensitive Language
EXPSPACE	Deterministic Exponential Space
EXPTIME	Deterministic Exponential Time
FO	First-Order Logic
L	Deterministic Logarithmic Space
NEXPSPACE	Non Deterministic Exponential Space
NEXPTIME	Non Deterministic Exponential Time
NL	Non Deterministic Logarithmic Space
NP	Non Deterministic Polynomial Time
NPSPACE	Non Deterministic Polynomial Space
NTIME	Non Deterministic Time
OLOG	Oracle Logarithmic
P	Deterministic Polynomial Time
PSPACE	Deterministic Polynomial Space
SL	Symmetric Logarithmic Space
TC	Transitive Closure
TIME	Deterministic Time
Γραμ.Τύπου	Γραμματική Τύπου
M.N.M.T	Μη Ντετερμινιστική Μηχανή Turing

ΑΝΑΦΟΡΕΣ

[1] A.Borodin, S.A. Cook, P.W. Dymond ,W.L. Ruzzo and M.Tompa, ”Two Applications of Complementation via Inductive Counting, ” this volume.

[2] S.R.Buss, S.A Cook, P.Dymond and L.Hay, ”The Log Space Oracle Hierarchy Collapses,” in preparation.

[3] S.A Cook, ”A Taxonomy of Problems with Fast Parallel Algorithms,” *Information and Control* **64** (1985), 2-22.

[4] J.Hartmanis and H.B Hunt, III, ”The LBA Problem,” *Complexity of Computation*, (ed. R.Karp), *SIAM-AMS Proc.* 7(1974), 1-26.

[5] John E.Hopcroft and Jeffry D.Ullman, *Introduction to Automata Theory, Languages, and Computation*, Addison-Wesley (1979).

[6] N.Immerman, ”Relational Queries Computable in Polynomial Time,” *Information and Control*, 68(1986), 86-104. A preliminary version of this paper appeared in *14th ACM STOC Symp.*, (1982), 147-152.

[7] N.Immerman, ”Languages That Capture Complexity Classes,” *SIAM J. Comput.* **16**, No 4 (1987), 760-778. A preliminary version of this paper appeared in *15th ACM STOC Symp.*, (1983), 347-354.

[8] N.Immerman, ”Expressibility as a Complexity Measure: Results and Directions,” *Second Structure in Complexity Theory Conf.* (1987), 194-202.

[9] S.Y. Kuroda, ”Classes of Languages and Linear-Bounded Automata,” *Information and Control* **7** (1964), 207-233.

[10] K.J. Lange, B. Jenner and B. Kirsig, ”The Logarithmic Hierarchy Collapses: $\mathrm{A}\Sigma_2^L = \mathrm{A}\Pi_2^L$,” *14th International Colloquium on Automata Languages and Programming* (1987).

[11] H.Lewis and C.H.Papadimitriou,” Symmetric Space Bounded Computation,” ICALP(1980). Revised version appeared in *Theoret. Comput. Sci.* **19** (1982), 161-187.

[12] S.R.Mahaney, ”Sparse Complete Sets for NP: Solution of a Conjecture of Berman and hartmanis,” *J. Comput. Systems Sci.* **25** (1982), 130-143.

[13] J. Reif, ”Symmetric Complementation,” *JACM* **31**, No. 2, April(1984), 401-421.

[14] U.Schoning and K.W Wagner, ”Collapsing Oracles, Census Functions, and Logarithmically Many Queries,” Report No. 140 (1987), Mathematics Institute, Univ. Augsburg.

[15] W.J Savitch, ”Relationships Between Nondeterministic and Deterministic Tape Complexities,” *J. Comput. System Sci.* **4** (1970), 177-192.

[16] Robert Szelepcsenyi, ”The Method of Forcing for Nondeterministic Automata,” *Bull. European Association Theor. Comp. Sci* (Oct. 1987), 96-100.

[17] Seinosuke Toda, ”Σ_2SPACE(n) is Closed Under Complement,” *JCSS* **35**, No.2 (1987), 145-152.

[18] Noam Nisan and Amnon Ta-Shma, ”Symmetric logspace is closed under complement”, *Chicago J. Theor. Comput. Sci.*, 1995.

[19] Ashok K. Chandra, Dexter C. Kozen, Larry J. Stockmeyer, ”Alternation”, *JACM* **28**, (1981).

[20] Lewis, Harry R. and Papadimitriou, Christos H., *Elements of the theory of computation*, Prentice Hall, 1998.

[21] Sipser Michael, *Introduction to the Theory of Computation*, Thomson Course Technology, 2006.

[22] H.B. Enderton, *A Mathematical Introduction to Logic*, Academic Press, 1972.

[23] N. Immerman, ”Nondeterministic Space is Closed Under Complementation,” *SIAM Journal on Computing* **17** (1988), 935-938.

[24] Róbert Szelepcsényi, ”The method of forced enumeration for nondeterministic automata,” *Acta Informatica* **26** (1988), 279-284.