



ΗΥ - 280 Θεωρία Υπολογισμού

Διάλεξη 17

Κωνσταντίνος Βάρσος

Τμήμα Επιστήμης Υπολογιστών – Πανεπιστήμιο Κρήτης

Χειμερινό εξάμηνο 2025 - 26

Υπολογισιμότητα

Τι είδαμε στην Διάλεξη 16

- Θεώρημα Rice.
- Αυτογράφος.
- Θεώρημα Αναδρομής.

Αυτοαναφορά

■ Υπάρχει υπολογίσιμη συνάρτηση $q : \Sigma^* \rightarrow \Sigma^*$ που για κάθε λέξη w επιστρέφει ως $q(w)$ την περιγραφή της TM P_w η οποία εκτυπώνει την w και τερματίζει.

$S =$ “Για είσοδο w :

1. Κατασκευάζουμε την εξής TM P_w :

$P_w =$ “Για οποιαδήποτε είσοδο :

1. Σβήνουμε την είσοδο.

2. Γράφουμε στην ταινία την λέξη $\langle w \rangle$.

3. Τερματίζουμε.”

2. Επιστρέφουμε ως έξοδο τη συμβολοσειρά $\langle P_w \rangle$.”

□

Μηχανή ΑΥΤΟΓΡΑΦΟΣ

$$A = P_{\langle B \rangle}.$$

$B =$ “Για είσοδο $\langle M \rangle$, όπου M τυχαία TM :

1. Υπολογίζουμε, από (1) την περιγραφή $S(\langle M \rangle)$, έστω $\langle N \rangle$. [$“S \rightarrow q”$] (2)
2. Συνδυάζουμε το αποτέλεσμα του 1. N με M
 ώστε να ολοκληρώσουμε την κατασκευή της TM.
3. Εκτυπώνουμε την περιγραφή αυτής της TM και τερματίζουμε.”



Μηχανή ΑΥΤΟΓΡΑΦΟΣ

π.χ., Θεωρούμε την TM M_1

$M_1 =$ “Για είσοδο x :

1. Βρες $\langle M_1 \rangle$.
2. Εκτύπωσε $\langle M_1 \rangle$.”



π.χ., Θεωρούμε την TM M_2

$M_2 =$ “Για είσοδο w :

1. Αν $w = \varepsilon$, εκτύπωσε 0.
2. Αλλιώς, βρες $\langle M_2 \rangle$.

Εκτέλεσε M_2 στο $\varphi(w)$.

Αν η M_2 στο $\varphi(w)$ επιστρέφει n εκτύπωσε $n + 1$.”

$\varphi(w)$ υπολογίσιμη συνάρτηση που επιστρέφει το $|w|$.



Θεώρημα Αναδρομής

- Χρησιμοποιείται για να αιτιολογήσει αυτοαναφορικά προγράμματα όπως τα M_1 , M_2 , υποστηρίζοντας ότι έχουν αντίστοιχες (και καθένα ισοδύναμες) TM.
- Κάθε πρόγραμμα μπορεί να 'βρει' την περιγραφή του εαυτού του και να τη χρησιμοποιήσει μέσα στον υπολογισμό του.
- Το θεώρημα αναδρομής μας επιτρέπει να κατασκευάσουμε μια TM Q τ.ω. $Q(\varepsilon)$ να παράγει το ίδιο αποτέλεσμα με $M(\langle Q \rangle)$.
- Με άλλα λόγια, η Q εκτελεί την υπορουτίνα M πάνω στον δικό της πηγαίο κώδικα $\langle Q \rangle$, και όχι απλώς στον κώδικα της M .

Θεώρημα Αναδρομής

Θεώρημα. Έστω αλφάβητο Σ , και έστω M μία αυθαίρετη TM με αλφάβητο εισόδου Σ . Τότε υπάρχει μία TM Q τ.ω. για κάθε συμβολοσειρά $w \in \Sigma^*$, $Q(w) = M(\langle\langle Q \rangle, w \rangle)$.

■ Αν η $M(\langle\langle Q \rangle, w \rangle)$ τερματίζει με κάποια συμβολοσειρά x γραμμένη στην ταινία, τότε και η $Q(w)$ τερματίζει με την ίδια συμβολοσειρά x γραμμένη στην ταινία.

■ Η $Q(w)$ αποδέχεται αν η $M(\langle\langle Q \rangle, w \rangle)$ αποδέχεται, και απορρίπτει αν απορρίπτει.

■ Αντίστοιχα, αν η $M(\langle\langle Q \rangle, w \rangle)$ δεν τερματίζει και τρέχει επί άπειρον, τότε και η $Q(w)$ δεν τερματίζει.

Θεώρημα Αναδρομής

Θεώρημα. Έστω T μια TM που υπολογίζει κάποια συνάρτηση $t : \Sigma^* \times \Sigma^* \rightarrow \Sigma^*$. Τότε υπάρχει TM R που υπολογίζει μια συνάρτηση $r : \Sigma^* \rightarrow \Sigma^*$ τ.ω. $\forall w \ r(w) = t(\langle R \rangle, w)$.

■ Η T είναι μια τυχία TM, της οποίας την λειτουργία μπορεί και να μην γνωρίζουμε, και θα την προσομοιώσουμε με μια διαφορετική TM.

■ Η T υπολογίζει την συνάρτηση t η οποία παίρνει TM στην είσοδο. Το Θεώρημα Αναδρομής λέει ότι υπάρχει R η οποία κατασκευάζει μια τυχαία συνάρτηση r τ.ω. για την δεδομένη TM R , $r(w) = t(\langle R \rangle, w) \ \forall w$.

■ $t : (TM) \times (input) \rightarrow (output)$, $r : (input) \rightarrow (output)$.

Θεώρημα Αναδρομής

■ Για να δημιουργήσουμε μια μηχανή που χρησιμοποιεί τον δικό της κώδικα, πρέπει να ακολουθήσουμε τα εξής βήματα:

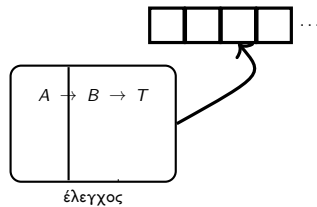
- 1 Η μηχανή πρέπει να καταφέρει να παράγει ένα αντίγραφο της δικής της περιγραφής.
- 2 Πρέπει να ενσωματώσει αυτό το αντίγραφο μέσα στον κώδικα της.
- 3 Πρέπει στο τέλος να καλέσει T πάνω σε αυτήν την περιγραφή.

Αυτό είναι αδύνατο χωρίς:

- A μια μηχανή που παράγει “χαμηλού επιπέδου κομμάτια κώδικα”.
- B μια μηχανή που συναρμολογεί αυτά τα κομμάτια σε ‘πρόγραμμα που περιέχει τον εαυτό του’.
- T μια μηχανή που κάνει το τελικό βήμα εφαρμογής.

Θεώρημα Αναδρομής

$$A = P_{\langle BT \rangle}.$$



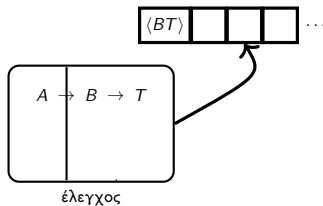
Θεώρημα Αναδρομής

$$A = P_{\langle BT \rangle}.$$

1. Εκτελούμε την A .

με την περιγραφή της B και της T στην ταινία.

Όμως η T έχει είσοδο w .



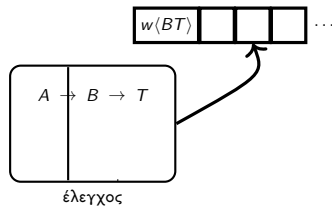
Θεώρημα Αναδρομής

$$A = P_{\langle BT \rangle}.$$

1. Εκτελούμε την A .

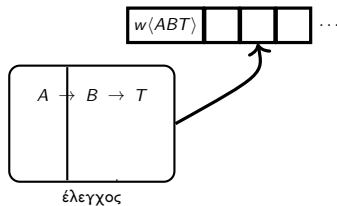
με την περιγραφή της B και της T στην ταινία.

Τροποποιούμε την A ώστε να μην σβήνει w .



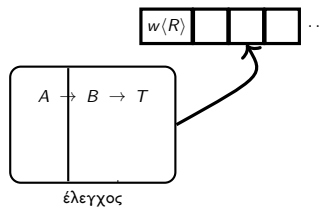
Θεώρημα Αναδρομής

2. Η B τρέχει και εφαρμόζει τη λειτουργία μετατροπής σε ΤΜ με βάση το περιεχόμενο της ταινίας. Ποιο είναι το αποτέλεσμα αυτού; Αυτό είναι η περιγραφή της A .



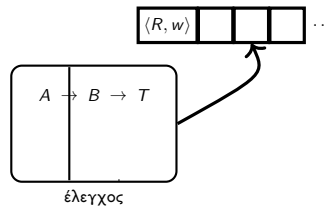
Θεώρημα Αναδρομής

3. Μετατρέπουμε την ABT σε R



Θεώρημα Αναδρομής

4. Ο έλεγχος περνάει στην T . Το w περνάει δεξιά του $\langle R \rangle$ και συγχωνεύεται μαζί του. Τέλος, εκτελούμε την T .



Θεώρημα Αναδρομής

π.χ., Θεωρούμε τη γλώσσα A_{TM} . Ας υποθέσουμε, για λόγους αντίφασης, ότι η D την διαγιγνώσκει. Κατασκευάζουμε την μηχανή R .

$R =$ “Για είσοδο w :

1. Βρες την περιγραφή $\langle R \rangle$.
2. Εκτέλεσε την D στην είσοδο $\langle R, w \rangle$.

Αν η D αποδέχεται $\langle R, w \rangle$, τότε απόρριψε.

Αν η D απορρίπτει $\langle R, w \rangle$, τότε αποδέξου.”

Το Θεώρημα Αναδρομής λέει ότι η R υπάρχει, εφόσον υπάρχει η D .

...

Θεώρημα Αναδρομής

π.χ., (συνέχεια) Θα κατασκευάσουμε την μηχανή T .

$T =$ “Για είσοδο $\langle M \rangle$ και w :

1. Εκτέλεσε την D στην είσοδο $\langle M, w \rangle$..

Αν η D αποδέχεται $\langle M, w \rangle$, τότε απόρριψε.

Αν η D απορρίπτει $\langle M, w \rangle$, τότε αποδέξου.”

Αν η R αποδέχεται την w , τότε η D αποδέχεται $\langle R, w \rangle$. Άρα η R απορρίπτει w από τον ορισμό της.

Αν η R δεν αποδέχεται την w , τότε η D απορρίπτει $\langle R, w \rangle$. Άρα R αποδέχεται w από τον ορισμό της. Αντίφαση.



Θεώρημα Αναδρομής

■ Σταθερό σημείο μιας συνάρτησης f λέγεται το σημείο x για το οποίο ισχύει ότι $f(x) = x$.

■ Για κάθε μετασχηματισμό πάνω σε TM , υπάρχει μια TM η οποία παραμένει αναλλοίωτη από τον μετασχηματισμό.

Θεώρημα. Έστω $t : \Sigma^* \rightarrow \Sigma^*$ υπολογίσιμη συνάρτηση, τότε υπάρχει TM F τ.ω. η συμβολοσειρά $t(\langle F \rangle)$ να περιγράφει μια TM ισοδύναμη της F .

Απόδειξη. Για δεδομένο μετασχηματισμό t , επιλέγουμε ως TM F την ακόλουθη:

$F =$ “Για είσοδο w :

1. Κατασκευάζουμε, μέσω του θεωρήματος αναδρομής, την αυτοπεριγραφή $\langle F \rangle$.
2. Υπολογίζουμε την συμβολοσειρά $t(\langle F \rangle)$. Έστω G η TM η οποία την περιγράφει.
3. Προσομοιώνουμε την G για είσοδο w .”

G, F είναι ισοδύναμες. Επίσης, $\langle G \rangle = t(\langle F \rangle)$. Άρα, $\langle F \rangle$ και $t(\langle F \rangle)$ είναι ισοδύναμες. $\square$

Θεώρημα Αναδρομής

■ Τι θα συμβεί όταν προσπαθήσουμε να χρησιμοποιήσουμε το Θεώρημα Αναδρομής για μια γλώσσα η οποία είναι διαγνώσιμη;

Θεώρημα Αναδρομής

■ Τι θα συμβεί όταν προσπαθήσουμε να χρησιμοποιήσουμε το Θεώρημα Αναδρομής για μια γλώσσα η οποία είναι διαγνώσιμη;

περίπτωση	αντιστροφή	Θ.Αναδρομής	σχόλια
διαγνώσιμη	ναι	R	$\nexists$ αντίφαση
διαγνώσιμη	όχι	R	R μιμείται D
μη-διαγνώσιμη	ναι	R	αντίφαση
μη-διαγνώσιμη	όχι	R	κανένα συμπέρασμα

Θεώρημα Αναδρομής

■ Όταν αποδεικνύεται η μη-διαγνωσιμότητα είναι συνήθως πιο αποτελεσματικό και σαφές να χρησιμοποιείτε απλούστερες μεθόδους όπως η αναγωγή από γνωστά προβλήματα που δεν μπορούν να διαγνωστούν.

■ Η απευθείας χρήση του θεωρήματος Αναδρομής συνήθως δεν προσφέρει αποτελεσματική συλλογιστική διαδικασία. [χωρίς πρόσθετη πληροφορία δεν θεωρείται καλή πρακτική.]

■ Το θεώρημα Αναδρομής τείνει να είναι πιο χρήσιμο για προβλήματα που περιλαμβάνουν αυτοαναφορικές κατασκευές ή πιο προχωρημένα θεωρητικά αποτελέσματα.

■ Το Θεώρημα Rice μας λέει **οι γλώσσες των TM δεν είναι διαγνώσιμες**. Το θεώρημα Αναδρομής μας λέει **το πως**.

Εισαγωγή – Πολυπλοκότητα

■ Μέχρι τώρα μας απασχολούσε αν μια γλώσσα είναι διαγνώσιμη, δηλαδή αν το αντίστοιχο πρόβλημα είναι επιλύσιμο.

■ Τώρα, θα εστιάσουμε σε γλώσσες που είναι διαγνώσιμες, και θα θεωρήσουμε περιορισμούς στις υπολογιστικές δυνατότητες όπως ο χρόνος και ο χώρος.

- Θεωρούμε ότι όλα τα προβλήματα είναι διαγνώσιμα.
- Μελετάμε πόσο αποδοτική είναι η διερεύνηση της διαγνωσιμότητας.
- Εξετάζουμε πως μπορεί να γίνει αποδοτικότερη η διερεύνηση της διαγνωσιμότητας.

■ **Στόχος**, να δώσουμε άνω φράγμα για όλες τις εισόδους μεγέθους n , για $n \in \mathbb{N}$. Αυτό ονομάζεται ανάλυση χειρότερης περίπτωσης [(ή) χειρότερη δυνατή πολυπλοκότητα]

Εισαγωγή – Πολυπλοκότητα

- Το πλήθος των βημάτων που εκτελεί ένας αλγόριθμος εξαρτάται από πολλές παραμέτρους.
- Υπολογίζουμε τον αριθμό εκτέλεσης ενός αλγορίθμου ως συνάρτηση **μόνο** του μήκους της συμβολοσειράς που αναπαριστά την είσοδο.
- Στην ανάλυση χειρότερης περίπτωσης, μας ενδιαφέρει, για κάθε μήκος, μόνο ο μέγιστος από τους χρόνους εκτέλεσης για όλες τις εισόδους του συγκεκριμένου μήκους.
- Υπάρχουν και άλλες περιπτώσεις, όπως η ανάλυση μέσης περίπτωσης.

Ασυμπτωτική Ανάλυση

■ Μέτρο αποδοτικότητας των αλγορίθμων που δεν εξαρτάται από τις ειδικές σταθερές της μηχανής και δεν απαιτεί την υλοποίηση των αλγορίθμων και τη σύγκριση του χρόνου που απαιτείται από τα προγράμματα. Είναι μαθηματικά εργαλεία για την αναπαράσταση της χρονικής πολυπλοκότητας των αλγορίθμων.

■ Big O (συνοπτικά O) ορίζει ένα ασυμπτωτικό ανώτατο όριο ενός αλγορίθμου. [φράσσει μια συνάρτηση μόνο από πάνω]

π.χ., InsertionSort, παίρνει γραμμικό χρόνο στην καλύτερη περίπτωση και τετραγωνικό χρόνο στη χειρότερη περίπτωση. Άρα, έχει χρονική πολυπλοκότητα $O(n^2)$. Απ την άλλη, η MergeSort έχει $O(n \log n)$ χρονική πολυπλοκότητα.

■ Για μια δεδομένη συνάρτηση $g(n)$, συμβολίζουμε με $O(g(n))$ το σύνολο των συναρτήσεων.

$$O(g(n)) = \{f(n) \mid \text{υπάρχουν θετικές σταθερές } c \text{ και } n_0 \text{ τ.ω. } 0 \leq f(n) \leq c \cdot g(n) \ \forall n \geq n_0\}$$

Ασυμπτωτική Ανάλυση

π.χ., $f(n) = 2n + 3$. Έχουμε, $2n + 3 \leq 10n \ \forall n \geq 1$. Επομένως, για $c = 1$, $n_0 = 1$ και $g(n) = n$, παίρνουμε $f(n) = O(n)$.

■ Big Θ (συνοπτικά Θ) φράσσει τις συναρτήσεις από πάνω και από κάτω, οπότε ορίζει την ακριβή ασυμπτωτική συμπεριφορά.

■ Ένας απλός τρόπος για να βρείτε τη Θ μιας έκφρασης είναι να παραλείψετε τους όρους χαμηλής τάξης και να αγνοήσετε τις σταθερές.

■ Για μια δεδομένη συνάρτηση $g(n)$, συμβολίζουμε με $\Theta(g(n))$ το σύνολο των συναρτήσεων.

$$\Theta(g(n)) = \left\{ f(n) \mid \begin{array}{l} \text{υπάρχουν θετικές σταθερές } c_1, c_2 \text{ και } n_0 \text{ τ.ω.} \\ 0 \leq c_1 \cdot g(n) \leq f(n) \leq c_2 \cdot g(n) \ \forall n \geq n_0 \end{array} \right\}$$

π.χ., $f(n) = 2n^2 + 3n + 4$. Έχουμε, $n^2 \leq 2n^2 + 3n + 4 \leq 2n^2 + 3n^2 + 4n^2 = 9n^2 \ \forall n \geq 1$. Άρα, για $c_1 = 1$, $c_2 = 9$, $n_0 = 1$ και $g(n) = n^2$, παίρνουμε $f(n) = \Theta(n^2)$.

Ασυμπτωτική Ανάλυση

■ Little o (συνοπτικά o), δηλώνει ένα γνήσιο άνω φράγμα στον ρυθμό αύξησης του χρόνου εκτέλεσης ενός αλγορίθμου.

$$o(g(n)) = \left\{ f(n) \mid \begin{array}{l} \text{υπάρχουν θετικές σταθερές } c \text{ και } n_0 \text{ τ.ω.} \\ f(n) < c \cdot g(n) \text{ για κάθε μέγεθος εισόδου } n \geq n_0 \end{array} \right\}$$

■ Σε ορισμένα συγγράμματα το Little o ορίζεται και ως εξής:

$$f(n) \in o(g(n)) \text{ σημαίνει } \lim_{n \rightarrow \infty} \frac{f(n)}{g(n)} = 0.$$

π.χ., $7n + 8 \in o(n^2)$ καθώς

$$\lim_{n \rightarrow \infty} \frac{f(n)}{g(n)} = \lim_{n \rightarrow \infty} \frac{7n + 8}{n^2} = \lim_{n \rightarrow \infty} \frac{7}{n} = 0$$

Ασυμπτωτική Ανάλυση

■ Οι Big O και Little o είναι παρόμοιοι. Η κύρια διαφορά είναι ότι στο $f(n) = O(g(n))$, το όριο $f(n) \leq g(n)$ ισχύει για κάποια σταθερά $c > 0$, αλλά στο $f(n) = o(g(n))$, το όριο $f(n) < c \cdot g(n)$ ισχύει για όλες τις σταθερές $c > 0$.

1 Big O $\rightsquigarrow$ Για τουλάχιστον μία επιλογή μιας σταθεράς $c > 0 \dots$ [είναι το σύνολο όλων των συναρτήσεων για τις οποίες ισχύει αυτή η συνθήκη]

2 Little o $\rightsquigarrow$ Για κάθε επιλογή μιας σταθεράς $c > 0 \dots$

π.χ., Τα παρακάτω είναι αληθή για το Big O, αλλά δεν θα ήταν αληθή αν χρησιμοποιούσατε το Little o

$$x^2 \in O(x^2), \quad x^2 \in O(x^2 + x), \quad x^2 \in O(200 \cdot x^2)$$

■ Σημειώστε ότι αν $f(x) \in o(g(x))$, αυτό συνεπάγεται $f(x) \in O(g(x))$.

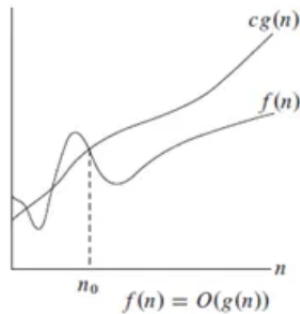
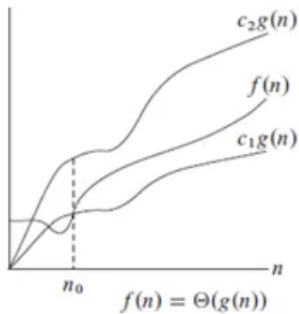
Ασυμπτωτική Ανάλυση

Συγκεντρωτικά, για $f, g : \mathbb{N} \rightarrow \mathbb{N}$

- $f(n) \in O(g(n))$ αν $\exists c > 0$ και $n_0 \geq 0$, τ.ω. $\forall n \geq n_0, f(n) \leq c \cdot g(n)$,
- $f(n) \in \Theta(g(n))$ αν $\exists c_1, c_2 > 0$ και $n_0 \geq 0$, τ.ω. $\forall n \geq n_0, c_1 \cdot g(n) \leq f(n) \leq c_2 \cdot g(n)$,
- $f(n) \in o(g(n))$ αν $\lim_{n \rightarrow \infty} \frac{f(n)}{g(n)} = 0$.

Ασυμπτωτική Ανάλυση

Σχηματικά, για $f, g : \mathbb{N} \rightarrow \mathbb{N}$



Ασυμπτωτική Ανάλυση

■ Το $x = \log_2 n$, η βάση είναι $b = 2$, είναι ισοδύναμο με το $2^x = n$.

■ Για τυχαίο b ισχύει $\log_b n = \frac{\log_2 n}{\log_2 b} = \frac{1}{\log_2 b} \cdot \log_2 n$. Άρα η τιμή του λογαρίθμου μεταβάλλεται μόνο κατά μια πολλαπλασιαστική σταθερά. Συνεπώς, όταν γράφουμε $f(n) \in O(\log n)$ δεν είναι απαραίτητο να διευκρινίζουμε την βάση.

■ Το Big O μπορεί να εμφανιστεί σε αριθμητικές παραστάσεις, π.χ. $f(n) = O(n^2) + O(n)$. Το ίδιο ισχύει και όταν εμφανίζεται σε κάποιον εκθέτη, π.χ. $f(n) = 2^{O(n)}$ [εδώ σημαίνει ένα άνω φράγμα της μορφής 2^{cn} , για κάποια σταθερά c]

■ Στην περίπτωση $f(n) = 2^{O(n \log n)}$, αντιπροσωπεύει ένα άνω φράγμα της μορφής n^c για κάποια σταθερά c .

■ Τα φράγματα της μορφής n^c ονομάζονται **πολυωνυμικά**, ενώ τα φράγματα της μορφής 2^{n^δ} , για $\delta > 0$, **εκθετικά**.

Ασυμπτωτική Ανάλυση

Μερικά παραδείγματα ακόμα,

1 $\sqrt{n} \in o(n)$

2 $3n \log n + 5n \log \log n + 2 \in O(n \log n)$

3 $\frac{1}{2}n^2 - 3n \in \Theta(n^2)$

4 $n^\epsilon \in o((\ln n)^4), \epsilon > 0 (\epsilon \neq e)$

5 $100n + \log n \in O(n + (\log n)^2)$

6 $\log(\log n) \in O(\log^\epsilon n), \forall \epsilon > 0.$

Ανάλυση Αλγορίθμων

■ Έστω γλώσσα $A = \{0^k 1^k \mid k \geq 0\}$. Πόσα βήματα χρειαζόμαστε μέχρι να καταλήξουμε στο αν η A είναι διαγνώσιμη;

Αρχικά, ας περιγράψουμε μια TM M η οποία να διαγιγνώσκει την A .

$M =$ “Για είσοδο w :

1. Διατρέχουμε την ταινία και εάν εντοπίσουμε κάποιο 0 στα δεξιά ενός 1, **απορρίπτουμε**.
2. Όσο η ταινία περιέχει και τα δύο είδη συμβόλων, 0 και 1, επαναλαμβάνουμε τα εξής:
Διατρέχοντας την ταινία, διαγράφουμε ακριβώς ένα 0 και ένα 1.
3. Εάν έχουν διαγραφεί όλα τα 0 αλλά υπάρχουν 1, **απορρίπτουμε**.
Εάν έχουν διαγραφεί όλα τα 1 αλλά υπάρχουν 0, **απορρίπτουμε**.
Διαφορετικά, **αποδεχόμαστε**.”

Ανάλυση Αλγορίθμων

Παρατηρούμε ότι,

- Διατρέχει την είσοδο w και ελέγχει αν $w \in 0^k 1^\ell$ [$O(n)$ βήματα]
- Διατρέχει την ταινία και διαγράφει ένα 0 και ένα 1. Απορρίπτει αν μείνουν στην ταινία είτε μόνο 0 είτε μόνο 1. [$O(n)$ βήματα]
- Πραγματοποιεί το προηγούμενο βήμα μέχρι να διαγράψει τα πάντα από την ταινία. [$O(n)$ επαναλήψεις]
- Αποδέχεται αν έχουν διαγραφεί τα πάντα.

■ Συνολικά, $O(n^2) + O(n)$ βήματα, δηλαδή $O(n^2)$ βήματα.

■ Άρα έχουμε το ακόλουθο αποτέλεσμα: μια TM M 1-ταινίας μπορεί να διαγνώσει την γλώσσα A σε $O(n^2)$ βήματα.

Ανάλυση Αλγορίθμων

Ορισμός. Έστω TM M , **χρόνος υπολογισμού** ή **χρονική πολυπλοκότητα** της M είναι μια συνάρτηση $f : \mathbb{N} \rightarrow \mathbb{N}$ τ.ω. για κάθε είσοδο x με $|x| = n$, η M τερματίζει μέσα σε $f(n)$ αριθμό βημάτων. [συνοπτικά θα την αναφέρουμε ως TM χρόνου $f(n)$]

■ Γενικά, επιλέγουμε την καλύτερη δυνατή συνάρτηση $f(n)$ όταν περιγράφουμε την χρονική πολυπλοκότητα μιας TM.

Ορισμός. Έστω $t : \mathbb{N} \rightarrow \mathbb{R}_{\geq 0}$ οποιαδήποτε συνάρτηση. Ορίζουμε την **κλάση χρονικής πολυπλοκότητας** $\text{TIME}(t(n))$ τη συλλογή όλων των γλωσσών που μπορούν να διαγνωστούν από μια TM χρόνου $O(t(n))$.

■ Για $t : \mathbb{N} \rightarrow \mathbb{R}_{\geq 0}$:

$$\text{TIME}(t(n)) = \{L \mid L = L(M) \text{ για TM } M \text{ με } O(t(n)) \text{ χρονική πολυπλοκότητα}\}$$

■ Στο προηγούμενο παράδειγμα, $A \in \text{TIME}(n^2)$.

Ανάλυση Αλγορίθμων

■ Έστω γλώσσα $A = \{0^k 1^k \mid k \geq 0\}$. Μπορούμε να κάνουμε κάτι καλύτερο από το $\text{TIME}(n^2)$;

Αρχικά, ας περιγράψουμε μια εναλλακτική ΤΜ M_1 η οποία να διαγιγνώσκει την A .

$M_1 =$ “Για είσοδο w :

1. Διατρέχουμε την ταινία μια φορά και εάν εντοπίσουμε κάποιο 0 στα δεξιά ενός 1, **απορρίπτουμε**.
2. Όσο η ταινία περιέχει και τα δύο είδη συμβόλων, 0 και 1, επαναλαμβάνουμε τα εξής:
 - i. Διατρέχοντας την ταινία, ελέγχουμε αν το συνολικό πλήθος των 0 και 1 είναι άρτιο ή περιττό. Εάν είναι περιττό, **απορρίπτουμε**.
 - ii. Διατρέχοντας την ταινία άλλη μια φορά, διαγράφουμε κάθε 2ο 0 και στη συνέχεια κάθε 2ο 1 (ξεκινώντας από τα πρώτα).
3. Εάν έχουν διαγραφεί όλα τα 0 και 1, **αποδεχόμαστε**.
Διαφορετικά, **απορρίπτουμε**.”

Ανάλυση Αλγορίθμων

■ Στο βήμα 2ii. μειώνει κάθε φορά το συνολικό πλήθος των 0 (1) στο μισό, $[13 \rightarrow 6 \rightarrow 3 \rightarrow 1 \rightarrow 0]$.

■ Στο βήμα 2i. Ελέγχει την ισοτιμία μεταξύ του πλήθους των 0 και του πλήθους των 1.

$|0| : 13 \rightarrow 6 \rightarrow 3 \rightarrow 1 \rightarrow 0, \quad \{\text{περιττός, άρτιος, περιττός, περιττός}\}$

$|1| : 13 \rightarrow 6 \rightarrow 3 \rightarrow 1 \rightarrow 0, \quad \{\text{περιττός, άρτιος, περιττός, περιττός}\}$

■ Αν τώρα πάρουμε την ακολουθία $\{\text{περιττός, άρτιος, περιττός, περιττός}\}$, αντικαταστήσουμε το περιττός με 1 και το άρτιος με 0, παίρνουμε 1011. Αν αντιστρέψουμε την τελευταία ακολουθία καταλήγουμε στην 1101 η οποία είναι η δυαδική αναπαράσταση του 13!

■ Εάν ισχύουν τα 2i. και 2ii. οι δυαδικές αναπαραστάσεις των 0 και των 1 συμπίπτουν. Άρα η M_1 διαγιγνώσκει την A.

Ανάλυση Αλγορίθμων

Παρατηρούμε ότι,

- Διατρέχει την είσοδο w και ελέγχει αν $w \in 0^k 1^\ell$ [$O(n)$ βήματα]
 - Διατρέχει την ταινία και διαγράφει κάθε 2ο 0 και 1 και υπολογίζει ισοτιμίες. [το πολύ $1 + \log_2 n$ βήματα]
 - Πραγματοποιεί το προηγούμενο βήμα μέχρι να διαγράψει τα πάντα από την ταινία. [$O(n)$ επαναλήψεις]
 - Αποδέχεται αν έχουν διαγραφεί τα πάντα.
- Συνολικά, $O(n) + (1 + \log_2 n)O(n) = O(n) + O(n \log n) = O(n \log n)$ βήματα.
- Άρα βελτιώσαμε την χρονική πολυπλοκότητα της διάγνωσης της A από $\text{TIME}(n^2)$ σε $\text{TIME}(n \log n)$.

Ανάλυση Αλγορίθμων

■ Μέχρι τώρα, στο παράδειγμα, είχαμε TM 1-ταινίας, η οποία μπορεί να διαγνώσει για την γλώσσα A σε $\text{TIME}(n \log n)$. Αυτό είναι το καλύτερο που μπορούμε να κάνουμε.

■ Τι θα γίνει όμως αν χρησιμοποιήσουμε μια TM ℓ -ταινιών; [$\ell > 1$]

Ας περιγράψουμε μια TM M_2 2-ταινιών η οποία να διαγιγνώσκει την A .

$M_2 =$ “Για είσοδο w :

1. Διατρέχουμε την ταινία μια φορά. Εάν εντοπίσουμε 0 στα δεξιά ενός 1, **απορρίπτουμε**.
2. Διατρέχουμε τα 0 της 1ης ταινία μέχρι να βρούμε το 1ο 1.
Ταυτόχρονα, αντιγράφουμε τα 0 στην 2η ταινία.
3. Διατρέχουμε τα 1 της 1ης ταινία μέχρι το τέλος της εισόδου. Για κάθε ένα 1 που διαβάζουμε στην 1η ταινία διαγράφουμε ένα 0 στην 2η ταινία..
Εάν διαγραφούν όλα τα 0 πριν τα 1, **απορρίπτουμε**.
4. Εάν έχουν διαγραφεί όλα τα 0, **αποδεχόμαστε**. Διαφορετικά, **απορρίπτουμε**.”

Ανάλυση Αλγορίθμων

Παρατηρούμε ότι,

- Διατρέχει την είσοδο w και ελέγχει αν $w \in 0^k 1^\ell$ [$O(n)$ βήματα]
 - Διατρέχει τα 0 της 1ης ταινίας και τα αντιγράφει στην 2η [$O(n)$ βήματα]
 - Διατρέχει τα 1 της 1ης ταινίας και διαγράφει 0 στην 2η [$O(n)$ βήματα]
 - Αποδέχεται αν έχουν διαγραφεί τα πάντα.
- Συνολικά, $O(n) + O(n) + O(n) = O(n)$ βήματα. Σε αυτή την περίπτωση λέμε ότι έχουμε **γραμμικό χρόνο**.
- Άρα η χρονική πολυπλοκότητα της διάγνωσης της A με μια TM 2-ταινιών είναι $\text{TIME}(n)$.

Σχετικότητα Μοντέλων

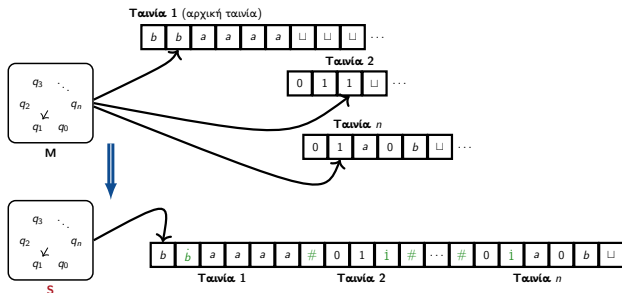
- Ο αριθμός των βημάτων για να διαγνώσουμε για την $A = \{0^k 1^k \mid k \geq 0\}$ εξαρτάται από το μοντέλο
 - $O(n \log n)$ για TM 1-ταινίας.
 - $O(n)$ για TM ℓ -ταινιών, με $\ell = 2$.
- Στην θεωρία Υπολογισιμότητας το αποτέλεσμα δεν εξαρτάται από το μοντέλο [θέση Church-Turing.]
- Στην θεωρία Πολυπλοκότητας το αποτέλεσμα εξαρτάται από το μοντέλο.
- Ποιο μοντέλο εντέλει θα χρησιμοποιήσουμε; [η εξάρτηση είναι συνήθως ασθενής (πολυωνυμική για αιτιοκρατικά ντετερμινιστικά μοντέλα).]

Σχέσεις Πολυπλοκότητας μεταξύ Μοντέλων

■ Μοντέλα: TM 1-ταινίας, TM ℓ -ταινιών, μη-ντετερμινιστική TM.

Θεώρημα. Έστω $t(n) \geq n$ και γλώσσα A . $\forall$ TM ℓ -ταινιών διαγιγνώσκει για την A σε χρόνο $t(n)$, $\exists$ ισοδύναμη TM 1-ταινίας τ.ω. $A \in TIME(t^2(n))$. [Ίσως θα ήταν πιο ορθό μαθηματικώς το $TIME(t(n)^2)$]

Απόδειξη. Θα αναλύσουμε την μετατροπή μιας TM ℓ -ταινιών M σε μια TM 1-ταινίας S .



Σχέσεις Πολυπλοκότητας μεταξύ Μοντέλων

Απόδειξη. Για κάθε βήμα της M η S σαρώνει το ενεργό τμήμα της ταινίας της 2 φορές

- μια για να προσδιορίσει την επόμενη κίνηση.
- μια για να κάνει την επόμενη κίνηση.

Ο χρόνος που περνάει η S στο ενεργό τμήμα της εξαρτάται από το μέγεθος της.

Ένα άνω φράγμα για το τελευταίο είναι το άθροισμα των μηκών των ενεργών τμημάτων των ℓ ταινιών της M . Καθένα από αυτά έχει μήκος το πολύ $t(n)$.

Συνεπώς, η σάρωση του ενεργού τμήματος της S απαιτεί $O(t(n))$ βήματα.

...

Σχέσεις Πολυπλοκότητας μεταξύ Μοντέλων

...

Απόδειξη. (συνέχεια) Για κάθε βήμα της M η S σαρώνει το ενεργό τμήμα της ταινίας της 2 φορές και κάνει το πολύ ℓ μετακινήσεις περιεχομένων προς τα δεξιά.

Κάθε μια από αυτές απαιτεί χρόνο $O(t(n))$.

Άρα ο συνολικός χρόνος που απαιτείται για την προσομοίωση οποιουδήποτε βήματος της M είναι $O(t(n))$.

Στο αρχικό στάδιο, η S μορφοποιεί την ταινία της, $O(n)$. Έπειτα, προσομοιώνει τα $t(n)$ βήματα της M , αναλώνοντας για καθένα από αυτά $O(t(n))$ βήματα.

Άρα, το στάδιο της προσομοίωσης απαιτεί $t(n) \times O(t(n)) = O(t^2(n))$ βήματα. Άρα η πλήρης προσομοίωση είναι $O(n) + O(t^2(n)) = O(t^2(n))$. Έχοντας υποθέσει $t(n) \geq n$ καταλήγουμε στο ζητούμενο. □

Σχέσεις Πολυπλοκότητας μεταξύ Μοντέλων

- Πριν χρησιμοποιήσαμε την ισοδυναμία μεταξύ TM 1-ταινίας και TM ℓ -ταινιών.
- Τι θα γίνει αν χρησιμοποιήσουμε την ισοδυναμία μεταξύ ντετερμινιστικής TM 1-ταινίας και μη-ντετερμινιστικής TM;

Ορισμός. Έστω N μια NTM που αποτελεί διαγνώστη. Ο χρόνος εκτέλεσης της N είναι μια συνάρτηση $f : \mathbb{N} \rightarrow \mathbb{N}$ η οποία $\forall n$ επιστρέφει ως $f(n)$ το μέγιστο πλήθος βημάτων που μπορεί να πραγματοποιήσει η N σε οποιονδήποτε κλάδο του υπολογισμού της για οποιαδήποτε είσοδο μήκους n .

Σχέσεις Πολυπλοκότητας μεταξύ Μοντέλων

Θεώρημα. Έστω $t(n) \geq n$. Τότε κάθε NTM N χρόνου $t(n)$ έχει μια ισοδύναμη ντετερμινιστική TM 1-ταινίας D χρόνου $2^{O(t(n))}$.

Απόδειξη. Κατασκευάζουμε μια TM D [χρησιμοποιεί 3 ταινίες!] που προσομοιώνει την N διερευνώντας το δέντρο του μη-ντετερμινιστικού υπολογισμού της.

$\forall$ είσοδο μήκους n , κάθε κλάδος του δέντρου της N έχει μήκος το πολύ $t(n)$.

Κάθε κόμβος στο δέντρο μπορεί να έχει το πολύ b παιδιά. Άρα το συνολικό πλήθος των φύλλων του δέντρου του N είναι $b^{t(n)}$.

Η προσομοίωση διερευνά το δέντρο οριζοντίως, άρα πριν φτάσει σε βάθος $d + 1$ έχει επισκεφθεί όλους τους κόμβους που είναι σε βάθος d .

...

Σχέσεις Πολυπλοκότητας μεταξύ Μοντέλων

Απόδειξη. (συνέχεια) Το συνολικό πλήθος των κόμβων του δέντρου είναι μικρότερο από το διπλάσιο του μέγιστου πλήθους των φύλλων,

$$\sum_{i=0}^{in} 2^i = 2^{n+1} - 1,$$

άρα φράσσεται από πάνω από το $O(b^{t(n)})$.

Ο χρόνος που χρειαζόμαστε ώστε να μεταβούμε από την ρίζα σε οποιονδήποτε κόμβο είναι $O(t(n))$.

Άρα ο συνολικός χρόνος εκτέλεσης της D είναι $O(t(n)b^{t(n)}) = 2^{O(t(n))}$.

[Από τις 3 στην 1 ταινία] Εφαρμόζουμε το προηγούμενο θεώρημα,
 $(2^{O(t(n))})^2 = 2^{O(2t(n))} = 2^{O(t(n))}$.

□

Σχέσεις Πολυπλοκότητας μεταξύ Μοντέλων

Σχηματικά, [από το βιβλίο του M.Sipser]

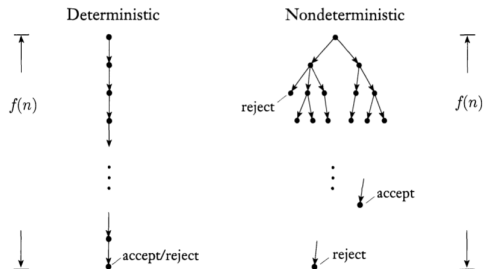


FIGURE 7.10

Measuring deterministic and nondeterministic time