Θεωρία Υπολογισμού και Πολυπλοκότητα

Κεφάλαιο 12.
Θεωρία Υπολογισιμότητας

30Μαρτίου, 17 Απριλίου 2007

Δρ. Παπαδοπούλου Βίκη

Θέση Church-Turing

- Τι μπορεί να υπολογιστεί και τι δεν μπορεί να υπολογιστεί?
- **Θέση Church-Turing:**
  - ότι μπορεί να υπολογιστεί (υπάρχει αλγόριθμος) μπορεί να υπολογιστεί από μια μηχανή Turing.
  - Δηλ. Αλγόριθμος είναι μια μηχανή Turing που τερματίζει για όλες τις εισόδους του προβλήματος
  - Τα προβλήματα που δεν μπορούν να επιλυθούν από μια μηχανή Turing δεν μπορούν να επιλυθούν. (είναι **μη επιλύσιμα**).
- Η θέση των Church-Turing είναι η επικρατέστερη άποψη για το θέμα.

Υπάρχουν γλώσσες που δεν είναι αναδρομικές ή αναδρομικά απαριθμίσιμες?

- Οι γλώσσες αναπαραστούνται με συμβολοσειρές (λέξεις).
  - Π.χ. L={w∈ Σ : |w|: περιττός}
  - Π.χ. $L=\{a^n b^n a^n\}$
  - Π.χ. Πεπερασμένα Αυτόματα ή μηχανές Turing
- Οι λέξεις που μπορούν να οριστούν για όλα τα αλφάβητα είναι απείρως αριθμήσιμες.
  - Οι γλώσσες που μπορούν να αναπαρασταθούν με συμβολοσειρές είναι απείρως αριθμίσημες

  ⇒ Υπάρχουν μόνο απείρως αριθμίσιμες γλώσσες

  ⇒ Υπάρχουν μόνο απείρως αριθμίσιμες γλώσσες αναδρομικές ή αναδρομικά αριθμήσιμες
- Όλες οι γλώσσες όμως είναι μη αριθμήσιμες

  ⇒ Υπάρχουν γλώσσες που δεν είναι αναδρομικές ή αναδρομικά αριθμήσιμες

Καθολικές Μηχανές Turing

- Οι μηχανές Turing σχεδιάζονται από την αρχή για μια λειτουργία
- Μπορούμε να προγραμματίσουμε μια μηχανή Turing να κάνει μια λειτουργία?
  - Καθολική Μηχανή Turing
    - Κωδικοποιώ την λειτουργία (μηχανή Turing, M_1) που θέλω να εκτελέσω ως μια λέξη
    - Την εισάγω στην είσοδο της Καθολική Μηχανή Turing
    - Η μηχανή προσδιορίζει τις μεταβάσεις της με βάση την κωδικοποιημένη M_1
-

Το Πρόβλημα του Τερματισμού

- Πρόβλημα Τερματισμού:
- Ας υποθέσουμε ότι έχουμε το εξής καταπληκτικό πρόγραμμα:
- Πρόγραμμα **halts(P, X)**, γραμμένο σε μια γλώσσα προγραμματισμού
  - Είσοδος:
    - ένα πρόγραμμα P στην ίδια γλώσσα
    - μια είσοδος X του προγράμματος P
  - Έξοδος:
    - NAI: αν το P τερματίζει πάντα με είσοδο X
    - OXI: αν το P θα έτρεχε επ' άπειρον

Πρόγραμμα Diagonal(X)

- **Diagonal(X)**

α : If halts(X, X) then goto α else halt

- Τι κάνει το diagonal?
  - Αν το πρόγραμμα halts αποφασίσει ότι το πρόγραμμα X θα τερμάτιζε με είσοδο X (τον εαυτό του), τότε το diagonal πέφτει σε άπειρη ανακύκλωση

    αλλιώς τερματίζει.

- ΕΡΩΤΗΣΗ: **Τερματίζει το diagonal(diagonal)?**
  - Το diagonal(diagonal) τερματίζει εάν και μόνο εάν το halts(diagonal, diagonal) επιστρέφει ΟΧΙ.
  - halts(diagonal, diagonal) επιστρέφει ΟΧΙ εάν και μόνο εάν το πρόγραμμα diagonal με είσοδο X=diagonal (δηλ. το diagonal(diagonal)) δεν τερματίζει.
  - Δηλ. diagonal(diagonal) τερματίζει εάν και μόνο εάν το diagonal(diagonal) δεν τερματίζει!

- ΑΤΟΠΟ.
- Άρα η υπόθεση ότι το halts() υπάρχει είναι λάθος.
  - Δηλ. Είναι αδύνατο να βρεθεί ένα πρόγραμμα ή αλγόριθμος που να αποφασίζει για οποιοδήποτε πρόβλημα αν θα τερματίσει ή θα πέσει σε άπειρη ανακύκλωση.
-

Ορισμοί Δύο Σημαντικών Γλωσσών

- K_0 = {(p(M),x) | η μηχανή Turing M αποδέχεται τη λέξη x}
  - οι λέξεις που κωδικοποιούν μια μηχανή Turing M που αποδέχεται τη λέξη x
  - Γλώσσα Τερματισμού
- K_1 = {x| η λέξη x κωδικοποιεί μια μηχανή Turing M_x που αποδέχεται τη λέξη x}
    - κωδικοποιεί μια μηχανή Turing M_x που αποδέχεται τη λέξη x
    - οι λέξεις που κωδικοποιούν μια μηχανή Turing και γίνονται δεχτές από αυτήν
    - Γλώσσα Αυτοτερματισμού
- Κοινό Χαρακτηριστικό: Αυτοαναφορά
  - K_1 = Η μηχανή M_x δέχεται τη λέξη x, δηλαδή παίρνει τον εαυτό της σαν είσοδο κωδικοποιημένο.

Παρατηρείστε

- Η K_1 είναι αναδρομικά απαριθμίσιμη.

Θεώρημα

- Θεώρημα. Η γλώσσα $\overline{K_1}$ δεν είναι αναδρομικά αριθμήσιμη.

Απόδειξη.

- Από τον ορισμό της γλώσσας $\overline{K_1}$, έχουμε ότι για κάθε λέξη $x \in \Sigma^*$,
  - $x \in \overline{K_1} \Leftrightarrow$
    - η λέξη x κωδικοποιεί μια μηχανή Turing M_x η οποία δεν αποδέχεται τη λέξη x}

    ή

    - η λέξη x δεν κωδικοποιεί μηχανή Turing

  (σχέση 1)

απόδειξη θεωρήματος

- Υποθέτουμε για να φτάσουμε σε αντίφαση ότι η γλώσσα $\overline{K_1}$ είναι αναδρομικά αριθμήσιμη.
- Από τον ορισμό των αναδρομικά αριθμησίμων γλωσσών,

  $\Rightarrow$ υπάρχει λέξη $x_0 \in \Sigma^*$ η οποία κωδικοποιεί μηχανή Turing M_{x_0} η οποία αποδέχεται τη γλώσσα $\overline{K_1}$.

$\Rightarrow$ για κάθε λέξη $x \in \Sigma^*$

- $x \in \overline{K_1} \Leftrightarrow$ η μηχανή Turing M_{x_0} αποδέχεται τη λέξη x .

(σχέση 2)

απόδειξη θεωρήματος

- Από τις σχέσεις 1 και 2, για κάθε λέξη $x \in \Sigma^*$,
  - η λέξη x κωδικοποιεί μηχανή Turing M_x η οποία δεν αποδέχεται τη λέξη x

  ή

  - η λέξη x δεν κωδικοποιεί μηχανή Turing

  $\Leftrightarrow$

  η μηχανή Turing M_{x0} αποδέχεται τη λέξη x .
- Θέτω $x := x_0$ στην τελευταία ισοδυναμία.

  $\Rightarrow$ η μηχανή Turing M_{x0} δεν αποδέχεται τη λέξη x_0

  $\Leftrightarrow$ η μηχανή Turing M_{x0} αποδέχεται τη λέξη x_0.

$\Rightarrow$ **Αντίφαση**.

Πόρισμα

- Αφού η K_1 είναι αναδρομικά απαριθμίσιμη.

$\Rightarrow$

- Πόρισμα. Το σύνολο των αναδρομικά απαριθμίσιμων γλωσσών δεν είναι κλειστό ως προς την συμπλήρωση.
- Πόρισμα 2. Η K_1 δεν είναι αναδρομική.
- Ολα τα στοιχεία προσδιορίζουν την K_0
- Η διαγώνιος προσδιορίζει την K_1
- Άφού η K_1 δεν είναι αναδρομική

$\Rightarrow$ ούτε η K_0 είναι αναδρομική.

	M_{xo}	M_{x1}	M_{x2}	
x_0	X	X	X	
x_1		X		
x_2			X	

Μη Επιλύσημα Προβλήματα με Μηχανές Turing

Συμπέρασμα

- Αφού η K_1 δεν είναι αναδρομικά απαριθμίσιμη
- και αν δεχθούμε την θέση Church-Turing ότι αλγόριθμος=Μηχανή Turing που αποφασίζει

  ⇒ Δεν υπάρχει αλγόριθμος που αποφασίζει δεδομένης μιας οποιασδήποτε μηχανής Turing M και εισόδου w εάν η M αποδέχεται την w ή όχι.

  - Το πρόβλημα είναι μη επιλύσιμο ή μη-αποφασίσιμο

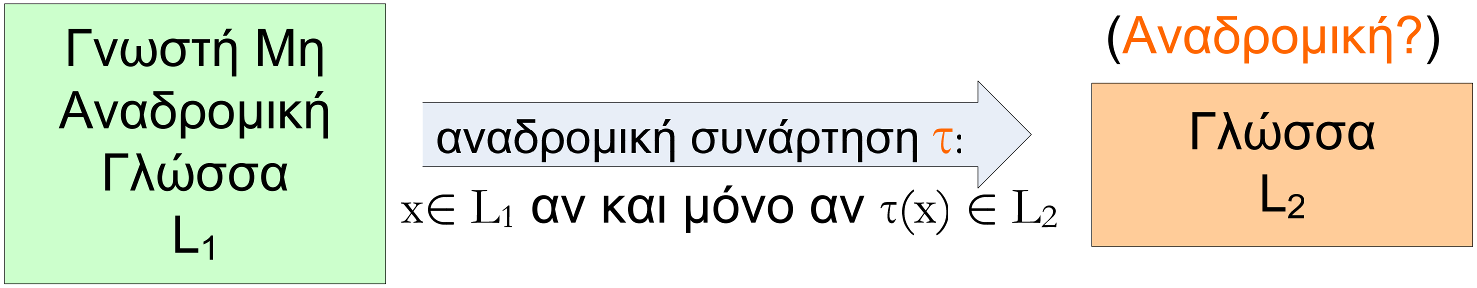
Πιο διάσημο μη-επιλύσιμο πρόβλημα:

- Το πρόβλημα του Τερματισμού: αποφασίζει εάν μια μηχανή Turing τερματίζει για μια δεδομένη είσοδο.

Συνέπεια: Πολλά Μη-επιλύσημα προβλήματα

- Ορισμός. (**Πολλά προς Ένα Αναγωγισιμότητα**) Έστω L_1, $L_2 \subseteq \Sigma^*$ δύο γλώσσες. Μια **αναγωγή από την** L_1 **στην** L_2 , $L_1 \leq L_2$, είναι μια αναδρομική συνάρτηση $\tau: \Sigma^* \to \Sigma^*$ τέτοια ώστε $x \in L_1$ αν και μόνο αν $\tau(x) \in L_2$.
- **Χρήση**: **Για να δείξουμε ότι η L_2 δεν είναι αναδρομική:**
  - Προσδιορίζω μια γλώσσα L_1 που είναι γνωστό ότι είναι μη αναδρομική
  - Ανάγω την L_1 στην L_2.

Χρήση Αναγωγών



- Θεώρημα 1. Αν η L_1 δεν είναι αναδρομική και υπάρχει μια αναγωγή από την L_1 στην L_2, τότε και η L_2 δεν είναι αναδρομική.

- Απόδειξη. Υποθέστε ότι η L_2 είναι αναδρομική.
  - Αποφασίζεται από μια μηχανή M_2.
  - Έστω T η μηχανή που υπολογίζει την αναγωγή τ.
  - ⇒ Η μηχανή TM_2 αποφασίζει για την L_1.
  - ⇒ Η L_1 όμως είναι μη αποφασίσιμη. ΑΤΟΠΟ. ■

Παράδειγμα.

- Αποδείξτε ότι η K_0 δεν είναι αναδρομική.

Απόδειξη.

- Ξέρουμε ότι η K_1 είναι μη αναδρομική. Δείχνουμε ότι $K_1 \leq K_0$
- Ορίζουμε την τ η οποία μετασχηματίζει
  - μια είσοδος της K_1 «M» $\rightarrow$ «M» «M» : είσοδος της K_0
  - η τ είναι αναδρομική συνάρτηση.

  $\Rightarrow$ η «M» είναι αποδεκτή από την K_1 εάν και μόνο αν η «M» «M» είναι αποδεκτή από την K_0

  $\Rightarrow$ η τ είναι συνάρτηση αναγωγής από την K_1 στην K_0

  $\Rightarrow$ αφού η K_1 δεν είναι αναδρομική

  $\Rightarrow$ από Θεώρημα 1 $\Rightarrow$ ούτε η K_0 είναι αναδρομική.

Ιδιότητες Γλωσσών

- Μεταβατικότητα. Για $L_1, L_{2,} L_3 \subseteq \Sigma^*$, αν $L_1 \leq L_2$ και $L_2 \leq L_3$ τότε $L_1 \leq L_3$.
- Αναδρομικότητα (αναδρομική απαριθμισιμότητα) Κληρονομεί προς τα Κάτω. Για $L_1, L_2 \subseteq \Sigma^*$, αν $L_1 \leq L_2$ και η L_2 είναι αναδρομική (αναδρομικά απαριθμίσιμη) τότε και η L_1 είναι αναδρομική (αναδρομικά απαριθμίσιμη).
- **Μη** Αναδρομικότητα (αναδρομική απαριθμισιμότητα) Κληρονομεί προς τα Πάνω. Για $L_1, L_2 \subseteq \Sigma^*$, αν $L_1 \leq L_2$ και η L_1 είναι μη αναδρομική (αναδρομικά απαριθμίσιμη) τότε και η L_2 είναι μη αναδρομική (αναδρομικά απαριθμίσιμη).
- Συμμετρίας ως προς την συμπλήρωση. Για $L_1, L_2 \subseteq \Sigma^*$, $L_1 \leq L_2$ αν και μόνο αν $\text{NOT}(L_1) \leq \text{NOT}(L_2)$.

Ιδιότητες

- Συγκρισιμότητα. Δύο γλώσσες $L_1, L_2 \subseteq \Sigma^*$ είναι συγκρίσιμες όταν ή $L_1 \leq L_2$ ή $L_2 \leq L_1$.
- Μη συγκρισιμότητα. Για $L \subseteq \Sigma^*$ αν η L είναι αναδρομικά αριθμίσιμη αλλά όχι αναδρομική, τότε ούτε $L \leq \text{NOT}(L)$ ούτε $\text{NOT}(L) \leq L$.

Ισχύει:

- Οποιαδήποτε συναναδρομικά αριθμίσιμη γλώσσα L είναι $L \leq \text{NOT}(K_0)$.

Παραδείγματα.

1. Θεώρημα. Για οποιαδήποτε αναδρομικά αριθμήσιμη γλώσσα L, $L \leq K_0$.

Απόδειξη.

- Η γλώσσα L είναι αναδρομικά αριθμήσιμη $\Rightarrow$ υπάρχει μηχανή Turing M τέτοια ώστε $L(M) = L$.
  - Για οποιαδήποτε λέξη $x \in \Sigma^*$, $x \in L$ αν και μόνο αν η μηχανή Turing M αποδέχεται τη λέξη x.
- Ορίζουμε τη συνάρτηση f ώστε για $x \in \Sigma^*$, $f(x) = \langle \rho(M), x \rangle$.
  - Η συνάρτηση f είναι αναδρομική συνάρτηση.
- Για οποιαδήποτε λέξη $x \in \Sigma^*$,
  - $x \in L$
  - αν και μόνο αν
  - η μηχανή Turing M αποδέχεται τη λέξη x (αφού $L = L(M)$)
  - **f:** αν και μόνο αν $\langle \rho(M), x \rangle$ (από τον ορισμό της γλώσσας K_0).

$\Rightarrow$ Η f είναι συνάρτηση αναγωγής από την L στην K_0. $\Rightarrow$ $L \leq_m K_0$. ■

Παραδείγματα. (φροντιστήριο)

- Αναδρομικότητα Κληρονομεί προς τα Κάτω. Για $L_1, L_2 \subseteq \Sigma^*$, αν $L_1 \leq L_2$ και η L_2 είναι αναδρομική τότε και η L_1 είναι αναδρομική.

Απόδειξη.

Παραδείγματα. (φροντιστήριο)

-

Παραδείγματα. (φροντιστήριο)

- Αποδείξτε ότι το παρακάτω γλώσσα είναι μη-επιλύσιμη (δηλ. μη αναδρομική):
  - $L_2=\{p(M) \mid$ η M τερματίζει για τουλάχιστον μια συμβολοσειρά e$\}$.

γνωρίζοντας ότι η παρακάτω συνάρτηση είναι μη-επιλύσημη:

  - $L_1=\{p(M) \mid$ η M τερματίζει με συμβολοσειρά εισόδου την e$\}$.

.. Δείχνω ότι $L_1 \leq L_2$