

Θεωρία Αναδρομής 1η Σειρά Ασκήσεων

Άσκηση 2

Θα δείξουμε ότι το πρόβλημα είναι αποφάνσιμο κατασκευάζοντας μηχανή Turing που το αποφασίζει. Παρατηρούμε το εξής: οι διαμορφώσεις μιας μηχανής turing $M = (Q, \Sigma, \Gamma, q_0, q_{yes}, q_{no})$ που η κεφαλή δεν περνάει από τη θέση n της ταινίας είναι πεπερασμένες. Συγκεκριμένα κάθε διαμόρφωση μπορεί να κωδικοποιηθεί ως (q, i, w) όπου q είναι η κατάσταση της μηχανής, i η θέση της κεφαλής και w η λέξη που περιέχει η ταινία. Αφού μετράμε τις καταστάσεις που η κεφαλή δεν φτάνει στη θέση n το μήκος της w είναι μικρότερο από n . Άρα όλες οι πιθανές διαφορετικές διαμορφώσεις της M είναι το πολύ $C = |Q| \cdot n \cdot |\Sigma|^n$. Αν λοιπόν τρέξουμε τη μηχανή M για περισσότερα από C βήματα έχουμε τρεις περιπτώσεις:

1. Θα περάσει η κεφαλή από τη θέση n
2. Θα τερματίσει η μηχανή χωρίς να περάσει η κεφαλή από τη θέση n
3. Μία διαμόρφωση θα επαναληφθεί.

Αν συμβεί το τρίτο ενδεχόμενο ο υπολογισμός δεν θα τερματίσει ποτέ αφού η Μηχανή θα επαναλαμβάνει τα ίδια configuration (ο υπολογισμός είναι ντετερμινιστικός) και η κεφαλή δεν θα περάσει ποτέ από τη θέση n αν δεν έχει ήδη περάσει.

Θα χρησιμοποιήσουμε την καθολική μηχανή Turing U για να προσομοιώσουμε $C + 1$ βήματα της M . Η μηχανή που θα κάνει την προσομοίωση θα έχει δύο επιπλέον ταινίες από αυτές που χρειάζεται η U . Αρχικά στην μία επιπλέον ταινία εκτελούμε n κινήσεις δεξιά και βάζουμε ένα ειδικό σύμβολο $*$ σε αυτή τη θέση. Στην άλλη κρατάμε έναν δυαδικό μετρητή ξεκινώντας από 0.

Η μηχανή U προσομοιώνει ένα ένα τα βήματα της M . Σε κάθε βήμα που προσομοιώνει κάνει τις εξής επιπλέον δουλειές: Κινεί την 1η επιπλέον κεφαλή ακριβώς όπως κινείται η κεφαλή της M , αυξάνει κατά 1 τον μετρητή, ελέγχει αν διάβασε το ειδικό σύμβολο $*$ και αν ναι πηγαίνει στην κατάσταση αποδοχής και τέλος συγκρίνει τον μετρητή με την ποσότητα C και αν είναι μεγαλύτερος από αυτή πηγαίνει στην κατάσταση απόρριψης.

Άσκηση 4

Θα κατασκευάσουμε την αναδρομική γλώσσα C ως τη γλώσσα που αποφασίζει μια μηχανή Turing (δηλαδή θα κατασκευάσουμε M_C και θα ορίσουμε $C = L(M_C)$). Η μηχανή M_C χρησιμοποιεί τις M_{coA} και M_{coB} που ημιαποφασίζουν τις $\bar{A}, \bar{B} \in RE$. Με είσοδο w η M_C προσομοιώνει "παράλληλα" (ένα βήμα την καθεμία) τις M_{coA}, M_{coB} ξεκινώντας από την M_{coA} . Αν η M_{coA} αποδεχτεί τότε η M_C αποδέχεται. Αν η M_{coB} αποδεχτεί τότε η M_C απορρίπτει. Ισχύουν τα κάτωθι:

- $\bar{A} \cup \bar{B} = \Sigma^*$ γιατί αν δεν ίσχυε τότε για κάποιο x θα είχαμε ότι $x \notin \bar{A} \cup \bar{B} \Rightarrow x \in \overline{\bar{A} \cup \bar{B}} = A \cap B$ που είναι άτοπο από υπόθεση
- $C \in REC$ γιατί αφού η ένωση είναι το Σ^* όπως δείξαμε η παραπάνω μηχανή τερματίζει σε κάθε πιθανή είσοδο είτε σε αποδοχή είτε σε απόρριψη.
- $B \subseteq C$. Γιατί αν $x \in B$ η M_{coB} δεν θα τερματίσει αλλά θα τερματίσει η M_{coA} και θα αποδεχτεί οπότε η M_C αποδέχεται εκ κατασκευής
- $A \subseteq \bar{C}$. Γιατί αν $x \in A$ η M_{coA} δεν θα τερματίσει αλλά θα τερματίσει η M_{coB} και θα αποδεχτεί οπότε η M_C απορρίπτει εκ κατασκευής

Τελικά όλα τα παραπάνω δείχνουν ότι η C διαχωρίζει τις A, B .

Άσκηση 5

Έστω L μία άπειρη αναδρομικά απαριθμήσιμη γλώσσα. Θα υπάρχει απαριθμητής E_L που θα απαριθμεί τα στοιχεία της γλώσσας (σε τυχαία σειρά). Θα αντιστοιχίσουμε τις λέξεις που προκύπτουν από τα άρτια βήματα εκτέλεσης στη γλώσσα L_1 και τις λέξεις που προκύπτουν από τα περιττά βήματα στη γλώσσα L_2 , ελέγχοντας τυχόν προηγούμενες εμφανίσεις της λέξης μέχρι εκείνο το βήμα (μπορεί να γίνει με μία μηχανή Turing που γράφει σε μία ταινία τις λέξεις μέχρι το σημείο n που είμαστε και ελέγχει αν υπάρχει εκεί η λέξη). Αν η λέξη έχει ξαναεμφανιστεί την παραλείπουμε και προχωράμε. Έτσι η γλώσσα L_1 θα περιλαμβάνει όλες τις (μοναδικές) λέξεις που προέκυψαν από τον απαριθμητή για άρτια βήματα, η γλώσσα L_2 αντίστοιχα όλες τις (μοναδικές) λέξεις που προέκυψαν από τον απαριθμητή για περιττά βήματα και ισχύει ότι $L_1 \cap L_2 = \emptyset$ και $L_1 \cup L_2 = L$.

Άσκηση 6

(\Rightarrow) Έστω $rank_A$ υπολογίσιμη συνάρτηση, τότε θα υπάρχει μία μηχανή Turing M_{rank_A} που θα την υπολογίζει. Θα φτιάξουμε μία μηχανή Turing M_{L_A} η οποία θα δέχεται ως είσοδο έναν αριθμό n σε δυαδική αναπαράσταση, θα τον μετατρέπει στην αντίστοιχη δεκαδική του αναπαράσταση x και θα υπολογίζει

το $rank_A(x)$ (μέσω της μηχανής M_{rank_A}). Στη συνέχεια θα αυξάνει το x κατά 1 και θα υπολογίζει το $rank_A(x+1)$. Αν το αποτέλεσμα είναι ίδιο με το προηγούμενο σημαίνει ότι το n δεν ανήκει στη γλώσσα L_A , οπότε η M_{L_A} απορρίπτει αλλιώς αποδέχεται. Άρα $L_A \in REC$.

(\Leftarrow) Έστω $L_A \in REC$, τότε θα υπάρχει μηχανή Turing M_{L_A} που την αποφασίζει. Θα φτιάξουμε μηχανή Turing M_{rank_A} η οποία θα δέχεται ως είσοδο έναν φυσικό αριθμό x και χρησιμοποιώντας έναν μετρητή i , ξεκινώντας από το 1 μέχρι το x , θα μετατρέπουμε τον i σε δυαδική αναπαράσταση και αν ανήκει στη γλώσσα (που θα αποφασίζουμε χρησιμοποιώντας την μηχανή M_{L_A}) τότε θα ανήκει και στο σύνολο A , θα αυξάνουμε έναν 2ο μετρητή $rank$ κατά 1 και θα προχωράμε στο επόμενο i . Μόλις το $i = x$ η μηχανή θα σταματήσει και η έξοδος της θα είναι το πλήθος των φυσικών αριθμών που είναι μικρότεροι από x και ανήκουν στο σύνολο A . Άρα η M_{rank_A} είναι μία μηχανή που υπολογίζει την συνάρτηση $rank_A(x) = |\{y \mid y < x \wedge y \in A\}|$, $A \subseteq \mathbb{N}$.

Άσκηση 7

- **ΛΑΘΟΣ** Θα δείξουμε ότι υπάρχει $x \in R$ τέτοιο ώστε $x \notin K$. Έστω ϕ_x η συνάρτηση που δεν ορίζεται πουθενά δηλαδή $(\forall n)\phi_x(n) = \uparrow$. Το $DOM(\phi_x)$ είναι αναδρομικό αφού είναι το κενό το οποίο αναγνωρίζεται από την μηχανή Turing που για κάθε είσοδο απορρίπτει. Εξ' ορισμού $\phi_x(x) = \uparrow$ και $M_x(x) = \uparrow$ δηλαδή $x \notin K$.
- **ΛΑΘΟΣ** Θα δείξουμε ότι υπάρχει x που ανήκει και στις δύο. Με αντίστοιχο με πριν τρόπο επιλέγουμε x τον αριθμό Goedel της σταθερής ολικής συνάρτησης $\phi(x) = 42$. Το πεδίο ορισμού της είναι αναδρομικό (παίρνουμε την Μ.Τ. που αποδέχεται κάθε λέξη) και επίσης $M_x(x) = \downarrow$ αφού η μηχανή για κάθε είσοδο τυπώνει 42. Συνεπώς $x \in M \cap K$
- ?
- **ΛΑΘΟΣ** Υπάρχει ένα προς ένα και επί $\sigma : K \rightarrow \bar{K}$. Έστω M_1, M_2, \dots η αρίθμηση των μηχανών. Έστω $f_1 : \mathbb{N} \rightarrow \mathbb{N}$ που απεικονίζει $n \mapsto i$ όπου i είναι ο αριθμός Goedel της n -οστής Μηχανής που τερματίζει και $f_2 : \mathbb{N} \rightarrow \mathbb{N}$ που απεικονίζει $n \mapsto i$ όπου i είναι ο αριθμός Goedel της n -οστής Μηχανής που δεν τερματίζει. Οι συναρτήσεις είναι καλά ορισμένες αφού και στις δύο κάθε στοιχείο αντιστοιχίζεται μονοσήμαντα στην εικόνα (δεν γίνεται μια θέση να καταλαμβάνεται από δύο μηχανές αφού έχουμε διάταξη) και είναι ολικές αφού για κάθε n υπάρχει σίγουρα η n -οστή μηχανή που (δεν) τερματίζει αλλιώς θα υπήρχε μέγιστο στοιχείο και το (\bar{K}) K θα ήταν πεπερασμένο. Η σ απεικονίζει $x \mapsto y$ τέτοια ώστε αν $f_1(n) = x$ τότε $f_2(n) = y$. Είναι επί γιατί για κάθε $y \in \bar{K}$ υπάρχει προεικόνα της f_2 , έστω n (κάπου είναι στη σειρά των μηχανών που δεν τερματίζουν με είσοδο τον εαυτό τους) και υπάρχει x τέτοιο ώστε $f_1(x) = n$. Είναι 1-1 γιατί αν $\sigma(x) = \sigma(x') = y$ τότε για κάποια n, n' $f_1(n) = x$ και $f_1(n') = x'$ και $y = f_2(n) = f_2(n')$ που συνεπάγεται ότι $n = n'$ και $x = x'$.

Άσκηση 8

Θα κατασκευάσουμε μηχανή Turing που να την υπολογίζει. Με είσοδο i η M θα χρησιμοποιεί ως υπορουτίνα τη μηχανή M_i ως εξής. Θα αρχικοποιεί έναν μετρητή στο 0 και σε κάθε γύρο θα προσομοιώνει κάποια βήματα της M_i για διαφορετικές εισόδους. Θα κρατάει επίσης και έναν άλλον μετρητή c που θα κρατάει το πλήθος των στοιχείων που αποδέχεται η γλώσσα. Στον γύρο 1 θα τρέχει την M_i με είσοδο 1 για 1 βήμα, στο γύρο 2 θα τρέχει τη M_i με εισόδους 1, 2 για δύο βήματα και γενικά στο γύρο j θα τρέχει τη M_i για j βήματα στις εισόδους $1, 2, \dots, j$. Στον j γύρο, αν η μηχανή M_i με είσοδο j τερματίσει ακριβώς στο j -οστό βήμα (και όχι νωρίτερα για να μην διπλομετράμε) τότε αυξάνουμε κατά 1 τον μετρητή c και αν κάποια στιγμή ο c γίνει μεγαλύτερος του i τυπώνει i και τερματίζει. Αν $|L(M_i)| \geq i$ τότε σε πεπερασμένο χρόνο η μηχανή θα τερματίσει με σωστό αποτέλεσμα, ενώ αντίθετα θα κολλήσει όπως πρέπει αφού η ϕ εκεί δεν ορίζεται.

Άσκηση 9

Έστω ότι έχουμε κάποιο x και θέλουμε να δείξουμε ότι $Im(\phi_{f(x)}) \in REC$. Το σημαντικό είναι ότι η το δοθέν σύνολο είναι πεπερασμένο. Δηλαδή για κάθε $y > x$ ισχύει ότι $y \notin Im(\phi_{f(x)})$ και συνεπώς $|Im(\phi_{f(x)})| \leq x$. Όμως κάθε πεπερασμένη γλώσσα είναι αναδρομική άρα παίρνουμε το ζητούμενο.

Άσκηση 10

Θα κατασκευάσουμε μηχανή Turing M_f που να την υπολογίζει. Θεωρούμε ότι M_i υπολογίζει την ϕ_i . Με είσοδο n η μηχανή δουλεύει ως εξής: Σε μία ταινία κρατάει το μερικό άθροισμα ξεκινώντας από 0 και σε μία άλλη ταινία αρχικοποιεί έναν μετρητή σε 0 που μετράει επαναλήψεις. Στην i -οστή επανάληψη προσομοιώνει τη λειτουργία της M_i με είσοδο n και το αποτέλεσμα που παίρνει (αν φυσικά τερματίζει) το προσθέτει στο άθροισμα και πηγαίνει στον επόμενο γύρο. Το αποτέλεσμα είναι σωστό αφού αν $\phi_1(n) \downarrow, \dots, \phi_n(n) \downarrow$ η μηχανή θα τερματίσει με το άθροισμα, αλλιώς θα κολλήσει όπως θα έπρεπε αφού σε αυτή τη περίπτωση η f δεν ορίζεται.

Άσκηση 11

(\Rightarrow) Έστω L μη κενή αναδρομικά απαριθμήσιμη γλώσσα. Τότε υπάρχει απαριθμητής E που απαριθμεί τα στοιχεία της. Ορίζουμε f ως εξής: $f(n) =$ το i -οστό στοιχείο που τυπώνει ο E .

- Η f είναι υπολογίσιμη. Θα κατασκευάσουμε μηχανή Turing M_f που υπολογίζει την f . Η M_f λειτουργεί ως εξής: Η μηχανή παίρνει σαν είσοδο n . Ξεκινάει και τρέχει τον E και κάθε φορά που αυτός πηγαίνει

στην κατάσταση εκτύπωσης η M_f αυξάνει έναν μετρητή και τον συγκρίνει με το n . Αν οι δύο τιμές είναι ίσες σταματάει τυπώνοντας το στοιχείο που έδωσε ο E .

- Η f είναι επί. Από τον ορισμό ισχύει ότι κάθε στοιχείο $x \in L$ κάποια στιγμή θα τυπωθεί από τον απαριθμητή και συνεπώς κάθε τέτοιο στοιχείο έχει προεικόνα.

(\Leftarrow) Έστω ότι υπάρχει $f : \mathbb{N} \rightarrow L$ που να είναι υπολογίσιμη και επί. Θα κατασκευάσουμε μηχανή Turing M_L που να ημιαποφασίζει την L . Έστω M_f η μηχανή που υπολογίζει την f . Η M_L δουλεύει ως εξής. Με είσοδο w προσομοιώνει ένα βήμα της εκτέλεσης $M_f(1)$. Στη συνέχεια προσομοιώνει 2 βήματα της εκτέλεσης $M_f(1)$ και $M_f(2)$ κ.ο.κ. (δηλαδή σε κάθε "γύρο" t προσομοιώνει t βήματα των εκτελέσεων $M_f(i)$ για $i \in \{1, 2, \dots, t\}$). Όταν κάποια από τις εκτελέσεις $M_f(i)$ τελειώσει ελέγχει αν η έξοδός της εκτέλεσης είναι ίση με w . Αν είναι αποδέχεται αλλιώς συνεχίζει να εκτελείται κανονικά. Επειδή η f είναι επί κάθε στοιχείο της $x \in L$ έχει προεικόνα δηλαδή για κάποιο $n \in \mathbb{N}$ θα ισχύει ότι $f(n) = x$ και συνεπώς η μηχανή για κάθε στοιχείο της γλώσσας σταματάει αποδεχόμενη.

Άσκηση 12

Διακρίνουμε δύο περιπτώσεις. Αν $Dom(f) = \emptyset$ τότε $\forall n \in \mathbb{N} \ f(n) = \uparrow$ και συνεπώς έχουμε ότι $Im(f) = \emptyset$ το οποίο σύνολο είναι αναδρομικό και αναγνωρίζεται από την μηχανή Turing με $q_{αρχική} = q_{no}$.

Αν $Dom(f) \neq \emptyset$ τότε ως μη κενό υποσύνολο των φυσικών το $Dom(f)$ έχει ελάχιστο στοιχείο, έστω n και έστω $f(n) = m$. Η f λοιπόν δεν ορίζεται για στοιχεία μικρότερα του n και επειδή είναι και φθίνουσα το m είναι το μέγιστο στοιχείο του $Im(f)$. Άρα $Im(f) \subseteq \{1, 2, \dots, m\}$ και συνεπώς είναι πεπερασμένο. Όμως κάθε πεπερασμένο σύνολο είναι αναδρομικό.

Για το δεύτερο κομμάτι θεωρούμε ότι $L \subseteq \mathbb{N}$ για να βγαίνει νόημα (το οποίο είναι OK αφού Σ^* και \mathbb{N} είναι ισομορφικά). Η απόδειξη του ζητούμενου είναι ακριβώς ίδια με παραπάνω, δηλαδή είτε η εικόνα των στοιχείων του L είναι το κενό σύνολο, είτε έχει όπως και πριν ελάχιστο στοιχείο.