ΕΠΛ 211 – 4η Σειρά Ασκήσεων 18/4/2008

Πρόχειρες Λύσεις

(α)

Η L_1 είναι RE =Α.Α. εφόσον μπορούμε να φτιάξουμε μια Turing μηχανή, η οποία για είσοδο $\langle \rho(M), \rho(x), \rho(q) \rangle$ προσομοιώνει την M για είσοδο x και το δέχεται αν η προσομοίωση καταλήξει στην κατάσταση q. Δηλαδή αν στο τέλος του "τρεξίματος" η M θα βρισκόταν στην κατάσταση q.

Η L_1 δεν είναι C =Σ.Α. (άρα θα είναι απλά RE).
Αρκεί να παρατηρήσουμε ότι η συνάρτηση που απεικονίζει το $\langle \rho(M), \rho(x) \rangle$ στο $\langle \rho(M), \rho(x), \rho(h) \rangle$ καθορίζει μια συνάρτηση αναγωγής από το K_0 στο L_1, οπότε θα ισχύει $K_0 \leq_m L_1$.

β)

Η L_2 είναι αναδρομική και χαρακτηρίζεται σαν R. Για να το δείξουμε αυτό, αρκεί να θεωρήσουμε την μηχανή Turing η οποία κάνει τον ακόλουθο υπολογισμό:

Για είσοδο $\langle \rho(M), \rho(x) \rangle$ προσομοίωσε την M, για είσοδο x, για 19 βήματα.
Εάν η προσομοίωση "halts" κατά την διάρκεια των 19 αυτών βημάτων τότε κάνε αποδεκτή την είσοδο. Διαφορετικά κάνε reject.

ΕΠΛ 211 – 4η Σειρά Ασκήσεων 18/4/2008

Πρόχειρες Λύσεις

(α)

Η L_1 είναι RE Α.Α. εφόσον μπορούμε να φτιάξουμε μια Turing μηχανή, η οποία για είσοδο $<\rho(M), \rho(x), \rho(q)>$ προσομοιώνει την M για είσοδο x και το δέχεται αν η προσομοίωση καταλήξει στην κατάσταση q. Δηλαδή αν στο τέλος του "τρεξίματος" η M θα βρισκόταν στην κατάσταση q.

Η L_1 δεν είναι C Σ.Α. (άρα θα είναι απλά RE).
Αρκεί να παρατηρήσουμε ότι η συνάρτηση που απεικονίζει το $<\rho(M), \rho(x)>$ στο $<\rho(M), \rho(x), \rho(h)>$ καθορίζει μια συνάρτηση αναγωγής από το K_0 στο L_1, οπότε θα ισχύει $K_0 \leq_m L_1$.

(β)

Η L_2 είναι αναδρομική και χαρακτηρίζεται σαν R. Για να το δείξουμε αυτό, αρκεί να θεωρήσουμε την μηχανή Turing η οποία κάνει τον ακόλουθο υπολογισμό:

Για είσοδο $<\rho(M), \rho(x)>$ προσομοίωσε την M, για είσοδο x, για 19 βήματα.
Εάν η προσομοίωση "halts" κατά την διάρκεια των 19 αυτών βημάτων τότε κάνε αποδεκτή την είσοδο. Διαφορετικά κάνε reject.

(γ)

Η L_3 είναι R.E., εφόσον μπορούμε να φτιάξουμε μια μηχανή Turing, η οποία για είσοδο $<p(M)>$ χρησιμοποιεί απλά μια σταθερή dovetailing στρατηγική για να προσομοιώσει την M σε κάθε πιθανή είσοδο.
Εάν κάποια από αυτές τις προσομοιώσεις "halts", τότε αυτή η μηχανή Turing θα δέχεται (accept) την είσοδο της.

Η L_3 δεν μπορεί να είναι C και γιαυτό χαρακτηρίζεται σαν R.E.
Εάν η L_3 ήταν C τότε:
Έστω f η συνάρτηση που απεικονίζει το $<p(M), p(x)>$ στο $<p(M')>$, όπου M' η Turing μηχανή που υλοποιεί τον ακόλουθο υπολογισμό.

"Για είσοδο w, σύγκρινε το w με το x. Εάν το w δεν είναι ίσο με το x, τότε προσομοίωσε την M για είσοδο x και σταμάτα (halt) αν και μόνο αν σταματάει (halts) η προσομοίωση. Διαφορετικά μπες σε άπειρο loop."
Τότε είναι εύκολο να δειχτεί ότι η f είναι μια συνάρτηση αναγωγής από το K_0 στην L_3.
Εναλλακτικά, από το θεώρημα του Rice προκύπτει ότι η L_3 είναι μη αναδρομική και συνεπώς, δεν μπορεί να είναι C.

(δ)

Η L_4 είναι C.

Μπορούμε να φτιάξουμε μια Turing μηχανή, η οποία για είσοδο $<\rho(M)>$ απλά προσομοιώνει την M για κάθε πιθανή είσοδο, μήκους 3 (οι είσοδοι αυτές είναι προφανώς, πεπερασμένες) και δέχεται (accepts) την είσοδο εάν όλες οι προσομοιώσεις σταματάνε (halt).

Η L_4 δεν είναι R.E. και συνεπώς χαρακτηρίζεται σαν C.
Ας θεωρήσουμε την συνάρτηση f η οποία απεικονίζει το $<\rho(M), \rho(x)>$ στο $<\rho(M')>$, όπου M' η μηχανή Turing που υλοποιεί τον ακόλουθο υπολογισμό:

"Για είσοδο w, έλεγξε αν |w| είναι μεγαλύτερο του 3. Εάν δεν είναι προσομοίωσε την M για είσοδο x και σταμάτα (halt) αν και μόνο αν η προσομοίωση σταματάει (halts). Αν |w|>3 κάνε ένα άπειρο loop."

Είναι απλό να δειχτεί ότι η f είναι μια συνάρτηση αναγωγής από το $\overline{K_0}$ στο L_4.

Διαφορετική αντιμετώπιση, θα ήταν να χρησιμοποιήσουμε το θεώρημα του Rice για να δείξουμε ότι η L_4 δεν είναι αναδρομική και συνεπώς δεν μπορεί να είναι RE (εφόσον είναι C).

ε) Υπάρχουν δυο τρόποι για να ~~λυθεί~~ αυτό το ερώτημα.

~~Ο πρώτος (και πιο επίπονος) είναι να κάνουμε κάποια επιχειρηματολογία βασιζόμενοι στο θεώρημα του Rice.~~

~~Ο δεύτερος είναι ο ακόλουθος:~~

Είναι πολύ εύκολο, να φτιάξουμε μια μηχανή Turing M' η οποία να δέχεται (accepts) το Σ^*.

Δεδομένης της M', η συνάρτηση που απεικονίζει το $<p(M)>$ στο $<p(M), p(M')>$ περιγράφει ουσιαστικά μια συνάρτηση αναγω- από το K_{tot} στο L_5

Αλλά το K_{tot} χαρακτηρίζεται σαν N, οπότε και το L_5 χαρα-τηρίζεται σαν N.

στ)

Η L_6 χαρακτηρίζεται σαν R. Η απόδειξη είναι απλή.
εαν $M = (K, \Sigma, s, \delta)$ τότε υπάρχουν μόνο $\underbrace{|K|}_{\text{πλήθος καταστάσεων}} \cdot \underbrace{|13|}_{\text{διαφορετικές θέσεις κεφαλής}} \cdot \underbrace{|\Sigma|^{13}}_{\text{διαφορετικά περιεχόμενα ταινίας}}$ configurations
ης M που χρησιμοποιούν το
ολύ 13 θέσεις της ταινίας
σόδου.
ions, εάν η M επαναλάβει ποτέ
δεύτερη φορά την ίδια ακριβώς
figuration, τότε θα κάνει ένα άπειρο loop, χωρίς να χρησιμοποιήσει
ί, επιπλέον θέσεις της ταινίας.
αι απλό να δείξουμε ότι μπορούμε να αποφασίσουμε την L_6
ecide) προσομοιώνοντας την M για $|K| \cdot |13| \cdot |\Sigma|^{13} + 1$ βήματα και
οντας δεκτή (we accept) την είσοδο, εάν η M θα είχε χρησιμο-
ιήσει το πολύ 13 θέσεις ταινίας, κατά την διάρκεια της προσομοί-
ης, ενώ διαφορετικά θα κάναμε reject.

(δ)

Η L_7 χαρακτηρίζεται σαν R.E.

Μπορούμε να δείξουμε ότι μια μηχανή Turing, χρησιμοποιεί μόνο ένα πεπερασμένο πλήθος θέσεων ταινίας εάν και μόνο εάν είτε σταματάει (halts) είτε επαναλαμβάνει μια configuration. Συνεπώς, θα μπορούσαμε να δεχτούμε (accept) την L_7, "τρέχοντας" μια προσομοίωση, η οποία καταγράφει κάθε configuration της M (κατά την διάρκεια της προσομοίωσης). Επίσης, εξετάζει για να δει αν επαναλαμβάνει κάποια configuration η οποία έχει ήδη καταγραφεί.

-άν η προσομοίωση σταματήσει (halts) ή εάν επαναληφθεί μια configuration, τότε το $<p(M), p(x)>$ είναι στοιχείο του L_7.

Για να δείξουμε ότι η L_7 δεν είναι C χρειάζεται ένα τρυκ. Η ιδέα είναι να βρούμε μια συνάρτηση αναγωγής από το K_0 στο L_7. Δυστυχώς, η προφανής συνάρτηση που απεικονίζει το $<p(M), p(x)>$ στο $<p(M), p(x)>$ δεν δουλεύει, εφόσον το $<p(M), p(x)>$ μπορεί να είναι στο L_7, ακόμα και εάν η M δεν σταματάει (halts) ποτέ για είσοδο x.

Αυτό που χρειαζόμαστε για να παρακάμψουμε αυτό το πρόβλημα, είναι μια μέθοδος προσομοίωσης η οποία απαιτεί έναν άπειρο αριθμό θέσεων ταινίας για να προσομοιώσει "non halting" υπολογισμούς μιας Turing μηχανής M. Υπάρχουν αρκετοί τρόποι για να υλοποιηθεί αυτό. Ένας απλός, είναι: να υποχρεώσουμε την προσομοίωση να σταματάει ανάμεσα σε κάθε προσομοιούμενο βήμα της M, και να κάνει shift όλην την μη κενή ταινία ένα τετράγωνο (μια θέση) προς τα δεξιά.

η)
Η L_8 χαρακτηρίζεται σαν N.

Έστω η συνάρτηση f, η οποία απεικονίζει το <ρ(M)> στο <ρ(M')>, όπου M' μια μηχανή Turing η οποία υλοποιεί τον ακόλουθο υπολογισμό.

"Για είσοδο n, προσομοίωσε την M για είσοδο το n-οστό string του αλφάβητου της M και επίστρεψε 1, αν ο υπολογισμός σταματήσει (halts)."

Λέγοντας n-οστό string εννοούμε σύμφωνα με κάποια κανονική διάταξη των strings του αλφάβητου της M.

Η M' υπολογίζει μια total function από το N στο N εάν και μόνο εάν η M σταματάει (halts) σε όλα τα δυνατά strings εισόδου.

Συνεπώς, η f είναι μια συνάρτηση αναγωγής από το K_{tot} στο L_8.

Επομένως, (δεδομένου ότι η L_8 δεν είναι R.E. και δεν είναι ( ) η L_8 χαρακτηρίζεται σαν N.

(θ)

Η L_g χαρακτηρίζεται σαν Ν.

~~Μπορούμε να δείξουμε ότι η L_g δεν είναι αναδρομική (όχι decidable) εφαρμόζοντας το Θεώρημα του Rice για την "ιδιότητα" (property)~~

$$\cancel{P(L_g) = \begin{cases} T, \text{ if } \alpha b \in L_g \cap \{b\alpha\} = \emptyset \\ F, \text{ διαφορετικά} \end{cases}}$$

Τώρα θα δείξουμε ότι η L_g δεν είναι C.

Έστω η f που απεικονίζει το $<p(M), p(x)>$ στο $<p(M')>$, όπου M' είναι μια Turing μηχανή η οποία υλοποιεί τον ακόλουθο υπολογισμό:

"Για είσοδο w, σύγκρινε το w με το αb.
Εάν το w ισούται με το αb, προσομοίωσε την M για είσοδο x και σταμάτα (halt) εάν και μόνο εάν σταματάει (halts) η προσομοίωση.
Εάν το w δεν ισούται με το αb, κάνε μια άπειρη ανακύκλωση."

Η μηχανή M' σίγουρα θα μπαίνει σε άπειρη ανακύκλωση για κάθε είσοδο, εκτός από αb, άρα θα μπαίνει σε άπειρη ανακύκλωση για είσοδο $b\alpha$.

Επιπλέον, η M' δέχεται (accepts) το αb αν και μόνο αν η M σταματάει (halts) στο x.

Συνεπώς, το $<p(M')>$ ανήκει στην γλώσσα αν και μόνον αν η M σταματάει (halts) στο x, και συνεπώς η f είναι μια συνάρτηση αναγωγής από το K_0 στο L_g.

Για να δείξουμε ότι η L_g δεν είναι R.E. θεωρούμε την συνάρτηση g η οποία απεικονίζει το $<p(M), p(x)>$ στο

$<\rho(M'')>$ όπου M'' μια Turing μηχανή η οποία υλοποιεί τον ακόλουθο υπολογισμό:

"Για είσοδο w, σύγκρινε το w με το $b\alpha$ και σταμάτα (halt) εάν δεν είναι ίσα.
Εάν το w είναι $b\alpha$ προσομοίωσε την M για είσοδο x και σταμάτα (halt) αν και μόνο αν σταματήσει (halt) η προσομοίωση."

Η μηχανή M'' σίγουρα σταματάει για κάθε είσοδο εκτός από $b\alpha$, και συνεπώς πρέπει να σταματήσει για αb.
Επιπλέον η M'' δεν σταματάει (halt) για $b\alpha$ αν και μόνο αν η M δεν σταματάει στο x.
Συνεπώς, το $<\rho(M'')$ ανήκει στην γλώσσα αν και μόνο αν η M δεν σταματάει στο x, και συνεπώς η g είναι μια συνάρτηση αναγωγής από το $\overline{K_0}$ στο L_g.

(i) Η L_{10} χαρακτηρίζεται σαν Ν.

Πρώτα θα δείξουμε ότι δεν είναι C $= Σ.Α$ βρίσκοντας μια συνάρτηση αναγωγής από το K_0 στο L_{10}.

Έστω $<\rho(M), \rho(x)>$ στο K_0.

Κατασκευάζουμε το $<\rho(M')>$, όπου M' είναι η μηχανή Turing που υλοποιεί τον ακόλουθο υπολογισμό:

"Για είσοδο w, σταμάτα (halt) εάν το w δεν περιέχει κάποιο α. Διαφορετικά, προσομοίωσε την Μ για είσοδο x και σταμάτα (halt) εάν και μόνον εάν σταματάει (halts) η προσομοίωση."

Είναι εύκολο να δείξουμε ότι η M' σταματάει (halts) σε κάθε string που περιέχει ένα α, εάν και μόνον εάν η Μ σταματάει (halts) για είσοδο x.

Αυτό περιγράφει την συνάρτηση αναγωγής από το K_0 στο L_{10}

Τέλος, θα δείξουμε ότι η L_{10} δεν είναι R.E. $= Α.Α.$

Έστω $K_1 = \{ <\rho(M)> : M$ halts στο empty string $\}$.

Παρατηρούμε ότι το K_1 είναι R.E. αλλά όχι C.

Συνεπώς, αν βρούμε μια συνάρτηση αναγωγής από το $\overline{K_1}$ στο L_{10} θα έχουμε ότι η L_{10} δεν είναι R.E.

Έστω $<\rho(M)>$ στο $\overline{K_1}$. Κατασκευάζουμε το $<\rho(M'')>$, όπου M'' η Turing μηχανή που υλοποιεί τον ακόλουθο υπολογισμό.

"Για είσοδο w, σταμάτα (halt) εάν το w δεν περιέχει α. Διαφορετικά, προσομοίωσε την Μ στο empty string για $|w|$ βήματα. Εάν η Μ δεν σταματάει (halts) σε αυτόν τον αριθμό βημάτων, σταμάτα (halt), διαφορετικά μπες σε ένα άπειρο loop."

Συνεπώς, η M" σταματάει (halts) σε κάθε string που περιέχει α α, εάν και μόνον εάν, η M δεν σταματάει στο empty string.
...υτό περιγράφει την συνάρτηση αναγωγής από το $\overline{K_1}$ στο L_{10}
...αι αποδεικνύει ότι το πρόβλημα δεν είναι R.E.
...πομένως, η L_{10} χαρακτηρίζεται σαν N.