ΕΘΝΙΚΟ ΜΕΤΣΟΒΙΟ ΠΟΛΥΤΕΧΝΕΙΟ

ΣΧΟΛΗ ΗΛΕΚΤΡΟΛΟΓΩΝ ΜΗΧΑΝΙΚΩΝ ΚΑΙ ΜΗΧΑΝΙΚΩΝ ΥΠΟΛΟΓΙΣΤΩΝ

1η Σειρά Ασκήσεων

Θεμελιώδη Θέματα Επιστήμης Υπολογιστών

3ο ΕΞΑΜΗΝΟ

ΔΟΝΤΗ ΕΙΡΗΝΗ

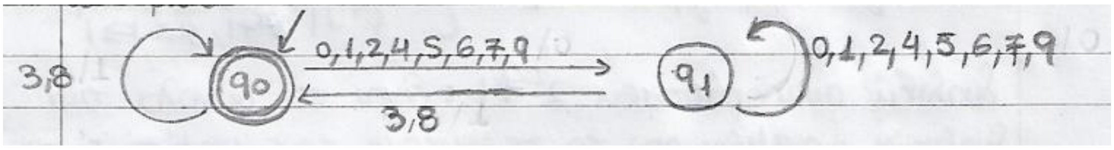
Α.Μ.:03119839

ΑΘΗΝΑ 2020-2021

Άσκηση 1:

Για να ισχύει ότι n mod 5 = 3, πρέπει το τρέχον ψηφίο που διαβάζεται να είναι είτε 3 είτε 8. Με άλλα λόγια, πρέπει ο αριθμός που θα διαβαστεί συνολικά να έχει ως τελευταίο ψηφίο το 3 ή το 8, γιατί αν n = m+p, όπου p είναι το τελευταίο ψηφίο του αριθμού n, τότε n mod 5 = m mod 5 + p mod 5 = p mod 5 .

Χρειαζόμαστε, λοιπόν, δύο καταστάσεις q0, q1 όσες και οι δυνατές εκβάσεις για να σχεδιάσουμε το πεπερασμένο αυτόματο.



Το πεπερασμένο αυτόματο αποδέχεται στην περίπτωση που η τελική κατάσταση είναι q0. Σε αντίθετη περίπτωση απορρίπτει.

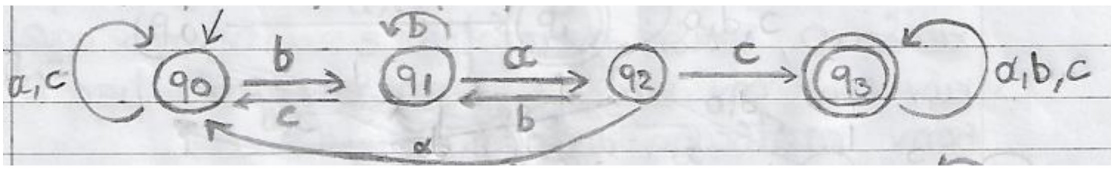
Εκτέλεση με είσοδο 403: (q0)403 → 4(q1)03 → 40(q1)3 → 403(q0) ΑΠΟΔΟΧΗ.

Άσκηση 2:

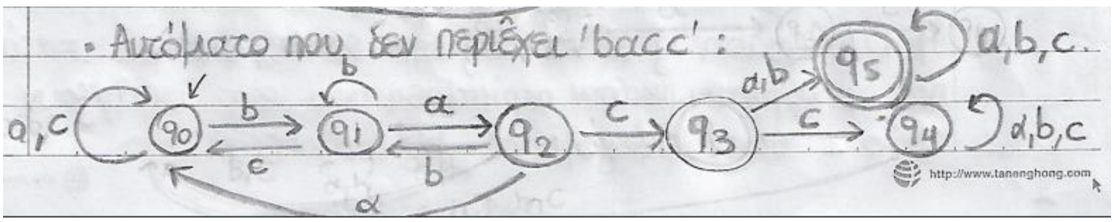
(i): DFA για L1 := {w∈{a,b,c}*| w περιέχει ‘bac’ και όχι ‘bacc’}.

Θα δημιουργήσουμε δύο απλούστερα DFA:

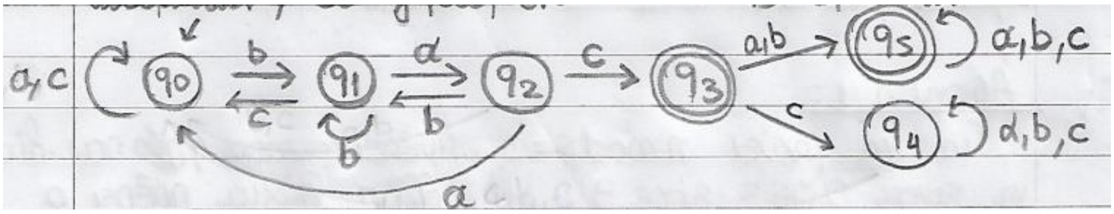
Αυτόματο που περιέχει ‘bac’:



Αυτόματο που δεν περιέχει ‘bacc’:

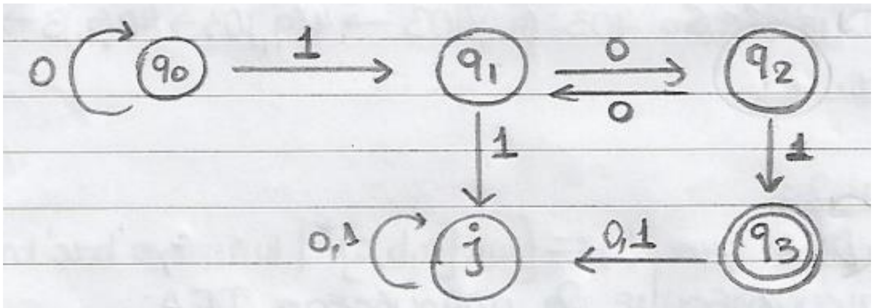


Παίρνοντας την τομή αυτών των δύο απλούστερων αυτομάτων, το ζητούμενο DFA της L1 είναι:



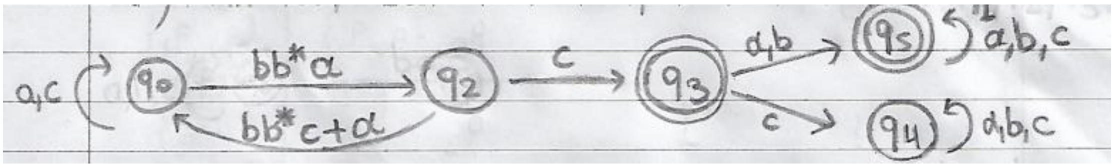
Ουσιαστικά, είναι παρόμοιο αυτόματο με εκείνο που δεν περιέχει την ‘bacc’ συμβολοσειρά. Αυτό είναι λογικό, γιατί το αυτόματο που δεν περιέχει τη συμβολοσειρά ‘bacc’ περιέχει όλες τις καταστάσεις του αυτομάτου που περιέχει όλες τις καταστάσεις του αυτόματου που περιέχει τη συμβολοσειρά ‘bac’. Π.χ. συμβολοσειρά αποδοχής: (q0)bacac → b(q1)acac → ba(q2)cac → bac (q3)ac → bac a(q5)c → bacac (q5).

DFA για τη γλώσσα L2 := {w∈{0,1}| w δυαδική αναπαράσταση 2^{2k} +1 για k ≥ 1}: είναι το σύνολο των δυαδικών αριθμών που το τελευταίο τους ψηφίο είναι 1 (περιττοί) και ανάμεσα στο αρχικό ψηφίο 1 να υπάρχει περιττό πλήθος 0. Π.χ. για k = 1: 4 + 1 = 5 στο δεκαδικό και στο δυαδικό: 101 και για k = 2: 16 + 1 = 17 στο δεκαδικό και στο δυαδικό: 10001



Η κατάσταση q2 πηγαίνει πάλι στην q1 σε περίπτωση εισόδου 0, επειδή χρειαζόμαστε περιττό αριθμό μηδενικών πριν την είσοδο 1, κατά την οποία μεταβαίνουμε στην κατάσταση αποδοχής q3. Η κατάσταση j είναι junk state, δηλαδή κατάσταση που απορρίπτει π.χ. στην περίπτωση που θα υπάρξουν πάνω από δύο 1 στην εισαγόμενη συμβολοσειρά.

(ii) L1: Απαλείφοντας τις ενδιάμεσες καταστάσεις q1:



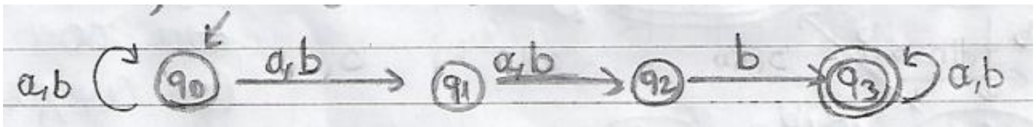
Οπότε, προκύπτει η παρακάτω κανονική έκφραση:

[(a+c)+bb*a(bb*c+a)]bb*ac[(a+b)(a+b+c)*+c(a+b+c)*]

L2: Αδιαφορούμε για την κατάσταση j:

Απαλείφοντας διαδοχικά τις καταστάσεις q0, q2, q1 και q3, τότε η κανονική έκφραση για L2 = 0*10(00)*1.

(iii) NFA για L3:



Επειδή η συμβολοσειρά έχει μήκος τουλάχιστον 4, τότε χρειαζόμαστε 4 καταστάσεις.

Με βάση το θεώρημα Rabin-Scott, για κάθε NFA, υπάρχει DFA που αποδέχεται την ίδια γλώσσα.

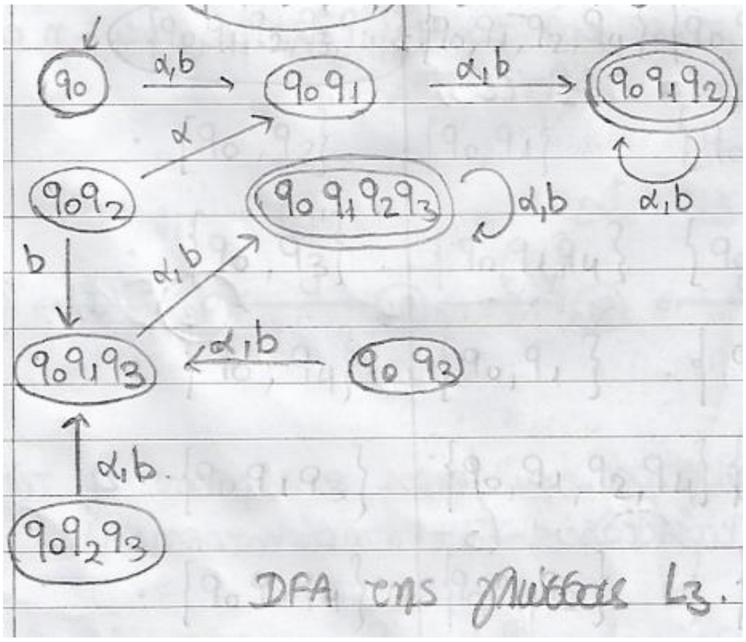
Δημιουργούμε τον πίνακα:

	a	b
q_0	$\{q_0, q_1\}$	$\{q_0, q_1\}$
q_1	$\{q_2\}$	$\{q_2\}$
q_2	$\emptyset$	$\{q_3\}$
q_3	$\{q_3\}$	$\{q_3\}$

Εξετάζοντας μόνο προσβάσιμα από {q0} σύνολα καταστάσεων έχουμε:

Q'/Σ	a	b
$\{q_0\}$	$\{q_0, q_1\}$	$\{q_0, q_1\}$
$\{q_0, q_1\}$	$\{q_0, q_1, q_2\}$	$\{q_0, q_1, q_2\}$
$\{q_0, q_2\}$	$\{q_0, q_1\}$	$\{q_0, q_1, q_3\}$
$\{q_0, q_3\}$	$\{q_0, q_1, q_3\}$	$\{q_0, q_1, q_3\}$
$\{q_0, q_1, q_2\}$	$\{q_0, q_1, q_2\}$	$\{q_0, q_1, q_2\}$
$\{q_0, q_1, q_3\}$	$\{q_0, q_1, q_2, q_3\}$	$\{q_0, q_1, q_2, q_3\}$
$\{q_0, q_2, q_3\}$	$\{q_0, q_1, q_3\}$	$\{q_0, q_1, q_3\}$
$\{q_0, q_1, q_2, q_3\}$	$\{q_0, q_1, q_2, q_3\}$	$\{q_0, q_1, q_2, q_3\}$

Οπότε, το DFA (το οποίο δεν είναι το ελάχιστο) της γλώσσας L3 είναι το παρακάτω:



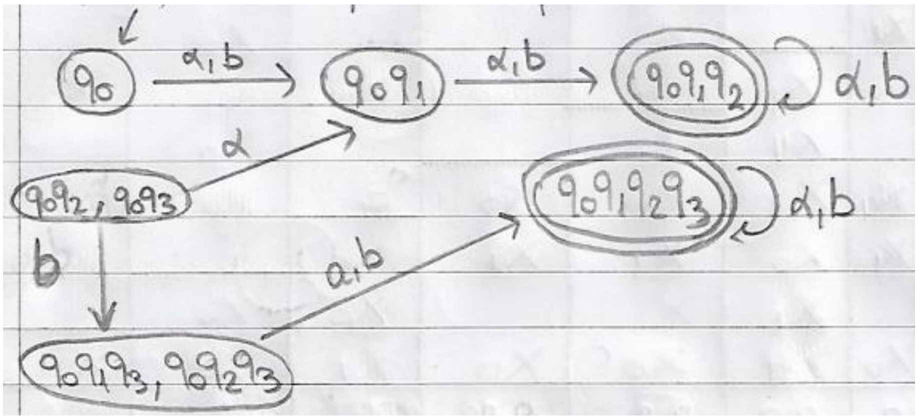
Δημιουργώντας τον τριγωνικό πίνακα, έχουμε:

q_0								
q_0q_1	X_1							
q_0q_2		X_1						
q_0q_3		X_1						
$q_0q_1q_2$	X_0	X_0	X_0	X_0				
$q_0q_1q_3$	X_1		X_1	X_1	X_0			
$q_0q_2q_3$		X_1			X_0			
$q_0q_1q_2q_3$	X_0	X_0	X_0	X_0		X_0	X_0	
	q_0	q_0q_1	q_0q_2	q_0q_3	$q_0q_1q_2$	$q_0q_1q_3$	$q_0q_2q_3$	$q_0q_1q_2q_3$

Καταρχάς, ξεχωρίζουμε τελικές με μη τελικές καταστάσεις (0-διακρίσιμες καταστάσεις) με X0. Έπειτα, φτιάχνουμε τον παρακάτω πίνακα. Αν τα ζεύγη (δ(B,a), δ(C,a)) ή/και (δ(B,b), δ(C,b)) είναι στον παρακάτω πίνακα σημειωμένα με X0, τότε σημειώνουμε στο (B,C) με X1 (1-διακρίσιμες καταστάσεις). Όσα ζεύγη περισσεύουν, ξαναελέγχουμε παραπάνω αν τα ζεύγη (δ(B,a), δ(C,a)) ή/και (δ(B,b), δ(C,b)) είναι στον παρακάτω πίνακα σημειωμένα με X1 και σημειώνουμε στο (B,C) με X2 (2-διακρίσιμες καταστάσεις) κ.ο.κ. Παρατηρούμε ότι η κατάσταση q0q1q3 ταυτίζεται με την q0q3 από τον παραπάνω πίνακα.

	(B,C)	(δ(B,a),δ(C,a))	(δ(B,b),δ(C,b))
X_1	(q_0q_1,q_0)	$(q_0q_1q_2,q_0q_1)$	$(q_0q_1q_2,q_0q_1)$
	(q_0q_2,q_0)	(q_0q_1,q_0q_1)	$(q_0q_1q_3,q_0q_1)$
	(q_0q_3,q_0)	$(q_0q_1q_3,q_0q_1)$	$(q_0q_1q_3,q_0q_1)$
X_1	$(q_0q_1q_3,q_0)$	$(q_0q_1q_2q_3,q_0q_1)$	$(q_0q_1q_2q_3,q_0q_1)$
	$(q_0q_2q_3,q_0)$	$(q_0q_1q_3,q_0q_1)$	$(q_0q_1q_3,q_0q_1)$
X_1	(q_0q_2,q_0q_1)	$(q_0q_1,q_0q_1q_2)$	$(q_0q_1q_3,q_0q_1q_2)$
X_1	(q_0q_3,q_0q_1)	$(q_0q_1q_3,q_0q_1q_2)$	$(q_0q_1q_3,q_0q_1q_2)$
	$(q_0q_1q_3,q_0q_1)$	$(q_0q_1q_2q_3,q_0q_1q_2)$	$(q_0q_1q_2q_3,q_0q_1q_2)$
X_1	$(q_0q_2q_3,q_0q_1)$	$(q_0q_1q_3,q_0q_1q_2)$	$(q_0q_1q_3,q_0q_1q_2)$
	(q_0q_3,q_0q_2)	$(q_0q_1q_3,q_0q_1)$	$(q_0q_1q_3,q_0q_1q_3)$
X_1	$(q_0q_1q_3,q_0q_2)$	$(q_0q_1q_2q_3,q_0q_1)$	$(q_0q_1q_2q_3,q_0q_1q_3)$
	$(q_0q_2q_3,q_0q_2)$	$(q_0q_1q_3,q_0q_1)$	$(q_0q_1q_3,q_0q_1q_3)$
X_1	$(q_0q_1q_3,q_0q_3)$	$(q_0q_1q_2q_3,q_0q_1q_3)$	$(q_0q_1q_2q_3,q_0q_1q_3)$
	$(q_0q_2q_3,q_0q_3)$	$(q_0q_1q_3,q_0q_1q_3)$	$(q_0q_1q_3,q_0q_1q_3)$
	$(q_0q_1q_2q_3,q_0q_1q_2)$	$(q_0q_1q_2q_3,q_0q_1q_2)$	$(q_0q_1q_2q_3,q_0q_1q_2)$
	$(q_0q_2q_3,q_0q_1q_3)$	$(q_0q_1q_3,q_0q_1q_2q_3)$	$(q_0q_1q_3,q_0q_1q_2q_3)$

Οπότε, το ελαχιστοποιημένο DFA είναι:



<u>Άσκηση 3:</u>

A) L0 = $\{0^n 1^+ 0^m: n, m \geq 1, n \neq m\}$.

Κανονική γλώσσα, είναι η γλώσσα που αναγνωρίζεται από DFA. Η παραπάνω γλώσσα δεν είναι κανονική, καθώς δεν μπορεί να περιγραφεί με πεπερασμένη μνήμη όπως συμβαίνει με πεπερασμένα αυτόματα. Χρησιμοποιούμε τη βοήθεια του Λήμματος Άντλησης. Έστω ότι η παραπάνω είναι κανονική. Τότε, υπάρχει φυσικός αριθμός N, τέτοιος ώστε αν z∈L0 και $|z| \geq n$ να ισχύουν τα συμπεράσματα του Λήμματος Άντλησης. Έχουμε $z = 0^n 1^+ 0^{n+1}$ και $|z| \geq n$, οπότε z = uvw με $|uv| \leq n$ και μη κενό v, τέτοια ώστε για κάθε i∈ N έχουμε $uv^iw \in$ L0, άτοπο για i=0.

B) L1 = $\{0^n 1^+ 0^m: n, m \geq 1, n=m\}$.

Έστω La = 0 0 0 0 0 (n στον αριθμό) $n \geq 1$, τότε η La δεν είναι κανονική καθώς δεν δίνεται ένα άνω όριο για τις τιμές του n. Οπότε, η πράξη $La1^+La$ δεν είναι κανονική γλώσσα. Οπότε, η L1 δεν είναι κανονική γλώσσα.

Γ) L2 = $\{0^+1^+0^+\}$.

Η γλώσσα L2 είναι κανονική, γιατί αποτελείται από τις πράξεις παράθεση και άστρο του Kleene. Συγκεκριμένα, εφόσον τα 0 και 1 είναι κανονικές γλώσσες, τότε 00* και

11* επίσης, γιατί τα άστρα του Kleene κανονικών γλωσσών είναι κανονικές γλώσσες. Επίσης, η παράθεση κανονικών γλωσσών είναι πάλι κανονική γλώσσα. Οπότε, και η L2 είναι κανονική γλώσσα.

Δ) L3 = {ww: w∈{0,1}*, το μήκος του w είναι ≤ $2^{2^{1000}}$}.

Εφόσον η w έχει πεπερασμένο μήκος, τότε και η ww θα έχει πεπερασμένο μήκος και συνεπώς πεπερασμένη μνήμη. Οπότε, η L3 αναγνωρίζεται από πεπερασμένο αυτόματο και συνεπώς είναι κανονική γλώσσα.

<u>Άσκηση 4:</u>

Α) Το αρχικό σύμβολο S αντικαθίσταται από τη συμβολοσειρά aaA, όπου το Α αντικαθίσταται από τις συμβολοσειρές aa και aaA και B. Η Β αντικαθίσταται με τις συμβολοσειρές b και bB.

Μία δυνατή ακολουθία που παράγεται είναι: S → aab

Οπότε, η γλώσσα που παράγεται είναι: L(G) = {$a^n b^m$| n,m ∈ N και m ≥ 0 και n ≥ 2}.

Β) G1: S → 0 B|1*A B|ε, A → 0|1* και B → (1^+ A)*

Γ) G2: S → 0^+ A|A|ε, A → (B B B)* και B → 10*

<u>Άσκηση 5:</u>

Βρίσκουμε δύο γλώσσες L1 και L2 οι οποίες είναι ανεξάρτητες συμφραζομένων. Η κλάση γλωσσών είναι κλειστή ως προς τις πράξεις ένωση και παράθεση, αν είναι context free η L, όπου α) L = L1 ∨ L2 και β) L = L1 L2. Έστω L1, L2 είναι γλώσσες του ίδιου αλφαβήτου Σ. Έστω L1, L2 και L παράγονται από context-free γραμματικές αντίστοιχα G1, G2 και G.

A) Οπότε, για το αρχικό σύμβολο S της G ισχύει:
$S \rightarrow S1|S2$ και αφού L1 και L2 context-free, τότε η $L = L1 \vee L2$ context-free.

B) Όμοια με πριν για το αρχικό σύμβολο S της G ισχύει:
$S \rightarrow S1\ S2$ και συνεπώς η $L = L1\ L2$. Με άλλα λόγια, η L γλώσσα αποτελείται από τις συμβολοσειρές της L1 που ακολουθούνται από τις συμβολοσειρές της L2. Οπότε, η L ικανοποιείται, αν ικανοποιούνται οι γλώσσες L1 και L2. Άρα, η L είναι context-free.

Οπότε, η κλάση των γλωσσών CF είναι κλειστή ως προς την ένωση και την παράθεση.

Άσκηση 6:

Έστω ότι υπάρχει πρόγραμμα Π που παίρνει είσοδο ένα πρόγραμμα Π’ και βρίσκει αν το Π’ τερματίζει με είσοδο την κενή συμβολοσειρά. Θα αποδείξουμε ότι αυτό δε γίνεται, βασισμένοι στο Halting Problem το οποίο είναι μη επιλύσιμο. Έστω ότι δημιουργούμε πρόγραμμα Π που δέχεται εισόδους e. Θέλουμε να καταλάβουμε αν θα τερματίσει με είσοδο την κενή συμβολοσειρά. Φτιάχνουμε ένα πρόγραμμα Π’ όπου το Π’ αποθηκεύει την κενή συμβολοσειρά, αρχικά ως σταθερά. Όταν η Π’ εκτελείται για είσοδο e, η είσοδος αυτή συγκρίνεται με την κενή συμβολοσειρά. Εφόσον e ≠ κενή συμβολοσειρά, το Π’ σταματά την εκτέλεση. Αντίθετα, εκτελείται το πρόγραμμα Π για e = κενή συμβολοσειρά. Δίνοντας, λοιπόν, το πρόγραμμα Π’ ως είσοδο στο Π μπορούμε να επιλύσουμε το Halting Problem. Με άλλα λόγια, αν το πρόγραμμα Π δεν τερματίζει για είσοδο την κενή συμβολοσειρά, τότε το πρόγραμμα Π’ θα τερματίζει για είσοδο την κενή συμβολοσειρά. Άρα, το Π θα δείξει θετική απάντηση. Από την άλλη, αν το Π τερματίζει για είσοδο την κενή συμβολοσειρά, τότε το πρόγραμμα Π’ τερματίζει για όλες τις εισόδους πέρα από την κενή συμβολοσειρά. Οπότε, το Π θα εμφανίσει αρνητική απάντηση. Το πρόβλημα τερματισμού, όμως, είναι μη επιλύσιμο πρόβλημα, οπότε δεν μπορεί να υπάρξει το πρόγραμμα Π. Με άλλα λόγια, το πρόβλημα ελέγχου, αν ένα πρόγραμμα τερματίζει με είσοδο την κενή συμβολοσειρά, είναι μη επιλύσιμο.

Άσκηση 7:

Μια φράση λέγεται φράση Horn, αν έχει το πολύ ένα θετικό literal. Ένας τύπος λέγεται ικανοποιήσιμος αν υπάρχει απονομή αληθοτιμών που τον καθιστά αληθή. Με άλλα λόγια, φ ικανοποιήσιμος αν και μόνο αν ¬φ δεν είναι αληθής για κάθε απονομή αληθοτιμών σε μεταβλητές. Αναθέτουμε αληθοτιμή true σε κάθε θετικό literal και false σε κάθε αρνητικό. Στην περίπτωση που βρούμε τύπο που να είναι ικανοποιήσιμος, επιστρέφουμε 'Ναι', ενώ αντίθετα 'Όχι'.

Πρόγραμμα Άσκηση 7

Μετρητής = 0

Μέτρησε το πλήθος n των παρενθέσεων

Για κάθε όρο ανάμεσα σε παρενθέσεις

/*'Ορος true*/

Αν υπάρχει x και ¬x στον όρο: μετρητής++

Αλλιώς συνέχισε στον επόμενο όρο

Τέλος επανάληψης.

Αν ο μετρητής = n – 1, τότε τύπωσε 'Ναι'

/*Επειδή έχουμε φράση Horn, max ένας όρος θα είναι θετικό literal*/

Μέτρησε το πλήθος m όλων των xk μεταβλητών

Όρισε πίνακα Π[m]

Βρες όρο που έχει μόνο ένα θετικό literal xk και θεώρησε Π[k]=true

/*Αφού οι υπόλοιποι όροι είναι σίγουρα true */

Πρόσθεσε τυχαίες τιμές στον υπόλοιπο πίνακα Π[m]

Τύπωσε Π[m]

Αλλιώς τύπωσε 'Όχι'

Τέλος προγράμματος.

Άσκηση 8:

A) Η κλίκα με το ανεξάρτητο σύνολο, αποτελούν συμπληρωματικά είδη υπογραφημάτων, γιατί το ανεξάρτητο σύνολο δεν έχει ακμές ανάμεσα σε k κορυφές, ενώ η κλίκα έχει όλες τις k κορυφές συνδεδεμένες με ακμές. Έστω ότι έχουμε ένα γράφο G(V,E), ο οποίος διαθέτει κλίκα μεγέθους k. Γνωρίζουμε ότι η κλίκα είναι NP-πλήρες ή γενικότερα NP, δηλαδή δίνει θετική απάντηση για τον γράφο G σε πολυωνυμικό χρόνο. Αν δημιουργήσουμε έπειτα γράφο G' αντίστροφο του G, τότε ο G' γράφος θα περιλαμβάνει ανεξάρτητο σύνολο αντίστοιχου μεγέθους. Οπότε, η Clique μετατράπηκε σε Independent Set. Η είσοδος της Clique είναι ο γράφος G, ενώ η έξοδός του (που παράλληλα είναι και είσοδος του Independent Set) είναι γράφος G'. Η μετατροπή της κλίκας σε ανεξάρτητο σύνολο κοστίζει πολυωνυμικό χρόνο, αφού απαιτούνται μόνο οι αφαιρέσεις ακμών για την μετατροπή αυτή. Οπότε, η Independent Set λύνεται σε πολυωνυμικό χρόνο, αφού η μετάβαση σε αυτό χρειάζεται πολυωνυμικό κόστος σε χρόνο. Με άλλα λόγια, αν το Clique είναι NP-πλήρες, τότε και το Independent Set είναι NP-πλήρες.

B) Το πρόβλημα Dense Subgraph είναι NP, γιατί χρειάζεται στην χειρότερη περίπτωση να ελέγξει όλες τις πιθανές ακμές, ενώ στην καλύτερη μόνο τις ακμές. Με άλλα λόγια, το πρόβλημα Dense Subgraph έχει πολυωνυμικό κόστος σε χρόνο. Ο αλγόριθμος μετάβασης από την Clique στην Dense Subgraph, απαιτεί πολυωνυμικό χρόνο (b-a), γιατί πρέπει να μετατρέψει τις a ακμές της κλίκας σε b. Οπότε, το Dense Subgraph είναι NP-πλήρες, όπως το Clique.