Θεμελιώδη Θέματα της Επιστήμης των Υπολογιστών

Γεώργιος Κυριακόπουλος ΑΜ: el18153

Πρώτη Σειρά Ασκήσεων

Άσκηση 1:

Για το αυτόματο θα χρησιμοποιήσω 6 καταστάσεις q_0, q_1, q_2, q_3, q_4, q_5, όπου q_i τέτοιο ώστε i = n mod 6, και n ένας αριθμός από το 0 έως το 9. Επομένως η κατάσταση που θα αποδέχεται θα είναι η q_2, όπου εν τέλει το υπόλοιπο της διαίρεσης του δοσμένου αριθμού με το 6 θα είναι 2. Για τη συνάρτηση μετάβασης έχουμε:

$\delta(q_i, x) = \{q_{(x+4*i)mod6}\}$

Η συνάρτηση βρέθηκε έπειτα από εφαρμογή μερικών παραδειγμάτων, δεδομένου του επιθυμητού αποτελέσματος για κάθε είσοδο και μέσω της επίλυσης σχέσεων που προκύπταν από αυτά. Επειδή έπρεπε να συσχετίσω τόσο την παρούσα κατάσταση όσο και την είσοδο με τη διαίρεση με το 6 ξεκίνησα με ένα γενικό τύπο της μορφής (x+m*i)mod6. Για παράδειγμα, έστω ο αριθμός 14 για τον οποίο πρέπει το αυτόματο να αποδέχεται. Ξεκινάμε από την κατάσταση q0 με είσοδο 1:

$\delta(q_0, 1) = \{q_{(1+m*0)mod6}\} = \{q_1\}$

Συνεχίζουμε με την κατάσταση q_1 με είσοδο 4, όπου θέλουμε να αποδέχεται άρα να πηγαίνει στην κατάσταση q_2:

$\delta(q_1, 4) = \{q_{(4+m)mod6}\} = \{q2\}$

Θέλουμε επομένως το (4+m)mod6 = 2. Οι αριθμοί για τους οποίους ισχύει n mod 6 = 2 είναι οι 2,8,14...

Για n = 2 έχουμε 4+m = 2 άρα m = -2 ενώ για n = 8 έχουμε 4+m = 8 άρα m = 4.

Κρατάω το πρώτο θετικό m = 4, άρα τελικά έχω τον τύπο για τη συνάρτηση μετάβασης: (x+4*i)mod6. Η εγκυρότητα του τύπου ελέγχθηκε και με πρόγραμμα σε C++.

Παρακάτω ο πίνακας μεταβάσεων του αυτομάτου:

FA	0	1	2	3	4	5	6	7	8	9
$-q_0$	q_0	q_1	q_2	q_3	q_4	q_5	q_0	q_1	q_2	q_3
q_1	q_4	q_5	q_0	q_1	q_2	q_3	q_4	q_5	q_0	q_1
q_2	q_2	q_3	q_4	q_5	q_0	q_1	q_2	q_3	q_4	q_5
q_3	q_0	q_1	q_2	q_3	q_4	q_5	q_0	q_1	q_2	q_3
q_4	q_4	q_5	q_0	q_1	q_2	q_3	q_4	q_5	q_0	q_1
q_5	q_2	q_3	q_4	q_5	q_0	q_1	q_2	q_3	q_4	q_5

Από το διάγραμμα καταστάσεων είναι φανερό ότι οι καταστάσεις q_0,q_3 και q_1,q_4 έχουν ίδιες μεταβάσεις με ίδια είσοδο και δεν είναι τερματικές, άρα μπορούν να συγχωνευτούν σε μία κατάσταση το κάθε ζευγάρι. Οι καταστάσεις q_2,q_5 έχουν μεν ίδιες μεταβάσεις με ίδια είσοδο, αλλά η q_2 είναι τερματική, ενώ η q_5 όχι και επομένως δεν συγχωνεύονται. Άρα ο τελικός πίνακας μεταβάσεων του αυτομάτου θα είναι ως εξής:

M-FA	0	1	2	3	4	5	6	7	8	9
-q_0q_3	q_0	q_1	q_2	q_3	q_4	q_5	q_0	q_1	q_2	q_3
q_1q_4	q_4	q_5	q_0	q_1	q_2	q_3	q_4	q_5	q_0	q_1
q_2	q_2	q_3	q_4	q_5	q_0	q_1	q_2	q_3	q_4	q_5
q_5	q_2	q_3	q_4	q_5	q_0	q_1	q_2	q_3	q_4	q_5

Άσκηση 2:

(στους παρακάτω πίνακες με -A συμβολίζεται μία αρχική κατάσταση A, ενώ με **A** μία τελική κατάσταση A)

i)

L1 = {w ∈ {0, 1} ∗ | κάθε '0' που εμφανίζεται στην w ακολουθείται από τουλάχιστον δύο '1'}

NFA	0	1
-A	B	A
B	-	C
C	-	D
D	B	D

L2 = {w ∈ {c, d, e} ∗ | η w περιέχει την συμβολοσειρά 'dce' και όχι τη συμβολοσειρά 'ecd'}

NFA	c	d	e
-A	A	A, B	A, E
B	C	-	-
C	-	-	D
D	D, F	D	D, E
E	F	-	-
F	-	X	-
X	X	X	X

L3 = {w ∈ {0, 1} ∗ | η w περιέχει δύο τουλάχιστον εμφανίσεις της συμβολοσειράς ’001’}

NFA	0	1
-A	A, B	A
B	C	-
C	-	D
D	D, E	D
E	F	-
F	-	X
X	X	X

L4 = {w ∈ {a, b} ∗ | η w αρχίζει με ’a’ και είναι περιττού μήκους ή αρχίζει με ’b’ και είναι άρτιου μήκους}

NFA	a	b
-A	B	D
B	C	C
C	B	B
D	E	E
E	D	D

ii)

L1 = {w ∈ {0, 1} ∗ | κάθε ’0’ που εμφανίζεται στην w ακολουθείται από τουλάχιστον δύο ’1’}

DFA	0	1
-A	B	A
B	X	C
C	X	D
D	B	D
X	X	X

Με τη γνωστή διαδικασία ελαχιστοποίησης DFA με τον τριγωνικό πίνακα, αποδεικνύεται εύκολα ότι αυτό το DFA είναι το ελάχιστο δυνατό, καθώς όλες οι θέσεις του πίνακα γεμίζουν και δεν υπάρχουν καταστάσεις που να μπορούν να συγχωνευτούν.

L2 = {w ∈ {c, d, e} ∗ | η w περιέχει την συμβολοσειρά ’dce’ και όχι τη συμβολοσειρά ’ecd’}

Για την κατασκευή του DFA θα χρησιμοποιήσω την ένωση δύο άλλων μικρότερων DFA, του DFA1 που αποδέχεται λέξεις που περιέχουν τη συμβολοσειρά ‘dce’ και το DFA2 που απορρίπτει κάθε λέξη που περιέχει τη συμβολοσειρά ‘ecd’.

DFA1	c	d	e
-A	A	B	A
B	C	B	A
C	A	B	D
D	D	D	D

DFA2	c	d	e
-A	A	A	B
B	C	A	B
C	A	D	B
D	D	D	D

DFA	c	d	e
-AA	AA	BA	AB
BA	CA	BA	AB
AB	AC	BA	AB
CA	AA	BA	DB
AC	AA	BD	AB
DB	DC	DA	DB
BD	CD	BD	AD
DC	DA	DD	DB
DA	DA	DA	DB
CD	AD	BD	DD
AD	AD	BD	AD
DD	DD	DD	DD

Το DFA αυτό προφανώς δεν είναι το ελάχιστο. Με τη διαδικασία ελαχιστοποίησης με τον τριγωνικό πίνακα καταλήγουμε ότι BD = CD = AD = DD, μία συγχωνευμένη κατάσταση που θα συμβολίζω ως DD. Το ελάχιστο DFA είναι το παρακάτω:

M-DFA	c	d	e
-AA	AA	BA	AB
AB	AC	BA	AB
AC	AA	DD	AB
DD	DD	DD	DD
BA	CA	BA	AB
CA	AA	BA	DB
DC	DA	DD	DB
DB	DC	DA	DB
DA	DA	DA	DB

L3 = {w ∈ {0, 1} ∗ | η w περιέχει δύο τουλάχιστον εμφανίσεις της συμβολοσειράς ’001’}

DFA	0	1
-A	AB	A
AB	ABC	A
ABC	ABC	AD
AD	ABDE	AD
ABDE	ABCDEF	AD
ABCDEF	ABCDEF	ADX
ADX	ABDEX	ADX
ABDEX	ABCDEFX	ADX
ABCDEFX	ABCDEFX	ADX

Το DFA αυτό δεν είναι το ελάχιστο. Με τη γνωστή διαδικασία ελαχιστοποίησης με τον τριγωνικό πίνακα βρίσκουμε το παρακάτω ελάχιστο DFA, δεδομένου ότι προκύπτει ADX = ABDEX = ABCDEFX (την συγχωνευμένη αυτή κατάσταση συμβολίζω ως ABCDEFX):

M-DFA	0	1
-A	AB	A
AB	ABC	A
ABC	ABC	AD
AD	ABDE	AD
ABDE	ADBCDEF	AD
ABCDEF	ABCDEF	ABCDEFX
ABCDEFX	ABCDEFX	ABCDEFX

L4 = {w ∈ {a, b} ∗ | η w αρχίζει με ’a’ και είναι περιττού μήκους ή αρχίζει με ’b’ και είναι άρτιου μήκους}

DFA	a	b
-A	B	D
B	C	C
C	B	B
D	E	E
E	D	D

Το DFA είναι ίδιο με το NFA στην προκειμένη περίπτωση, όμως δεν είναι το ελάχιστο. Αυτό θα το βρούμε με τη γνωστή διαδικασία με τον τριγωνικό πίνακα όπου εν τέλει θα πάρουμε το παρακάτω απλοποιημένο DFA, αφού λάβουμε υπόψη ότι προκύπτει D=C και E=B:

M-DFA	a	b
-A	EB	DC
EB	DC	DC
DC	EB	EB

iii)

L1 = {w ∈ {0, 1} ∗ | κάθε ’0’ που εμφανίζεται στην w ακολουθείται από τουλάχιστον δύο ’1’}

D = D1 + C1 | C = B1 | B = A0 + D0 | A = ε + A1

Από τις παραπάνω σχέσεις, το Arden’s Theorem (R = Q +RP -> R = QP*) και την ιδιότητα εR = R λαμβάνουμε την παρακάτω κανονική παράσταση για τη γλώσσα L1:

D = 1*011(1+011)*

L2 = {w ∈ {c, d, e} ∗ | η w περιέχει την συμβολοσειρά ’dce’ και όχι τη συμβολοσειρά ’ecd’}

D = Dc + Dd + De + Ce | C = Bc | B = Ad | A = Ac + Ad + Ae + ε

Από τις παραπάνω σχέσεις, το Arden’s Theorem (R = Q +RP -> R = QP*) και την ιδιότητα εR = R λαμβάνουμε την παρακάτω κανονική παράσταση για τη γλώσσα L2:

D = (c+d+e)*dce(c+d+e)*

Πρέπει ωστόσο αριστερά το (c+d+e)* να μην περιέχει τη συμβολοσειρά ‘ecd’ και δεξία το (c+d+e)* να μην είναι ίσο με cd(c+d+e)* ούτε και να περιέχει την συμβολοσειρά ‘ecd’.

L3 = {w ∈ {0, 1} ∗ | η w περιέχει δύο τουλάχιστον εμφανίσεις της συμβολοσειράς ’001’}

X = X0 + X1 + F1 | F = E0 | E = D0 | D = D0 + D1 + C1 | C = B0 | B = A0 | A = ε + A0 + A1

Από τις παραπάνω σχέσεις, το Arden’s Theorem (R = Q +RP -> R = QP*) και την ιδιότητα εR = R λαμβάνουμε την παρακάτω κανονική παράσταση για τη γλώσσα L3:

X = (0+1)*001(0+1)*001(0+1)*

L4 = {w ∈ {a, b} ∗ | η w αρχίζει με ’a’ και είναι περιττού μήκους ή με ’b’ και είναι άρτιου}

EB = Aa + DCa + DCb | A = ε | DC = Ab + EBa + Ebb

Από τις παραπάνω σχέσεις, το Arden’s Theorem (R = Q +RP -> R = QP*) και την ιδιότητα εR = R λαμβάνουμε την παρακάτω κανονική παράσταση για τη γλώσσα L4:

EB = [a+b(a+b)][(a+b)(a+b)]*

Άσκηση 3:

α) L = {w ∈ {0, 1} ∗ : το πλήθος των 0 στην w είναι διπλάσιο από το πλήθος των 1}

Έχουμε τη $0^{2n}1^n$. Υποθέτουμε ότι είναι κανονική και χρησιμοποιούμε το Pumping Lemma για να αποδείξουμε ότι είναι μη κανονική. Παίρνουμε σπάσιμο xyz: $x = 0^{2n-k}$, $y = 0^k$, $z= 1^n$. Από το PL για κάθε $i \geq 0$ πρέπει $xy^iz \in L$. Όμως $xy^iz = 0^{2n+k(i-1)}1^n$ και για i=2 έχουμε $xy^2z = 0^{2n+k}1^n$, όπου παρατηρούμε ότι 2n+k>2n, αφού k>0, άρα δεν τηρούνται οι αναλογίες με το διπλάσιο πλήθος 0. Επομένως, η γλώσσα L δεν είναι κανονική.

β) L = {w ∈ {0, 1} ∗ : η w δεν είναι παλινδρομική}

Υποθέτουμε ότι η γλώσσα είναι κανονική. Τότε το συμπλήρωμα της που είναι η L' = {w ∈ {0, 1} ∗ : η w είναι παλινδρομική} πρέπει να είναι και αυτή κανονική. Όμως, εύκολα αποδεικνύουμε με τη χρήση του Pumping Lemma ότι αυτή δεν είναι κανονική.

Έστω η 0^n10^n, όπου είναι παλινδρομική. Παίρνουμε σπάσιμο xyz: $x = 0^{n-k}$, $y = 0^k$, $z = 10^n$. Από το PL για κάθε $i \geq 0$ πρέπει $xy^iz \in L$. Όμως $xy^iz = 0^{n+k(i-1)}10^n$ και για i=2 έχουμε $xy^2z = 0^{n+k}10^n$, όπου παρατηρούμε ότι το xy^2z δεν είναι παλινδρομικό, αφού k>0. Επομένως, δεν είναι κανονική. Άρα σύμφωνα και με το πρώτο μέρος η γλώσσα L δεν είναι κανονική, λόγω άτοπου.

γ) L = {ww : w ∈ {0, 1} ∗ το μήκος της w είναι $\leq 10^{100}$}

Υποθέτουμε ότι η γλώσσα είναι κανονική. Θα αποδείξουμε ότι δεν είναι μέσω του Pumping Lemma. Έστω η 0^n10^n1 όπου προφανώς ανήκει στην L, όπου $2n+2 \leq 10^{100}$. Παίρνουμε σπάσιμο xyz: $x = 0^{n-k}$, $y = 0^k$, $z = 10^n1$. Από το PL για κάθε $i \geq 0$ πρέπει $xy^iz \in L$. Όμως $xy^iz = 0^{n+k(i-1)}10^n1$ και για i=2 έχουμε $xy^2z = 0^{n+k}10^n1$, όπου παρατηρούμε ότι το xy^2z δεν ανήκει στην γλώσσα L, αφού k>0. Επομένως, η γλώσσα L δεν είναι κανονική.

δ) L = {w ∈ {0, 1} ∗ : 0^n1^m όπου $n \neq m$ }

Υποθέτουμε ότι η γλώσσα είναι κανονική. Θα αποδείξουμε ότι δεν είναι μέσω του Pumping Lemma. Έστω δύο περιπτώσεις, $n<m$ και $n>m$. Για την πρώτη περίπτωση, έστω η 0^n1^{n+1} όπου ανήκει στη γλώσσα L. Παίρνουμε σπάσιμο xyz: $x = 0^{n-k}$, $y = 0^k$, $z = 1^{n+1}$. Από το PL για κάθε $i \geq 0$ πρέπει $xy^iz \in L$. Όμως $xy^iz = 0^{n+k(i-1)}1^{n+1}$ και για i=3 έχουμε $xy^3z = 0^{n+2k}1^{n+1}$, όπου k>0 και συγκεκριμένα k=1,2,3,… άρα σίγουρα το πλήθος των 0 θα είναι μεγαλύτερο από το πλήθος των άσσων (2k>1). Για τη δεύτερη περίπτωση, έστω η 0^n1^{n-1} όπου ανήκει στη γλώσσα L. Παίρνουμε σπάσιμο xyz: $x = 0^{n-k}$, $y = 0^k$, $z = 1^{n-1}$. Από το PL για κάθε $i \geq 0$ πρέπει $xy^iz \in L$. Όμως $xy^iz = 0^{n+k(i-1)}1^{n-1}$ και για i=0 έχουμε $xy^0z = 0^{n-k}1^{n-1}$, όπου k>0 και συγκεκριμένα k=1,2,3,… άρα σίγουρα το πλήθος των 1 θα είναι μεγαλύτερο από το πλήθος των μηδενικών (-k≤-1). Επομένως η γλώσσα L δεν είναι κανονική.

Άσκηση 4:

α) Η γραμματική G παράγει μία γλώσσα οι λέξεις της οποίας έχουν πρόθεμα με μεγαλύτερο ή ίσο πλήθος α από β. Στην ελάχιστη περίπτωση έχουμε την εξής παραγωγή: S -> aS -> aε -> a ή S -> aSbS -> aεbε -> ab. Από εκεί και πέρα με κάθε επιπλέον κίνηση προσθέτουμε τουλάχιστον 1 a ή 1 a και 1 b επομένως σίγουρα θα έχουμε μεγαλύτερο ή ίσο πλήθος a από b στο πρόθεμα της λέξης.

β) Η γραμματική για τη γλώσσα $\{a^{3n}b^{2n} \mid n\geq 1\}$ είναι η εξής:

G: S -> aaaAbb

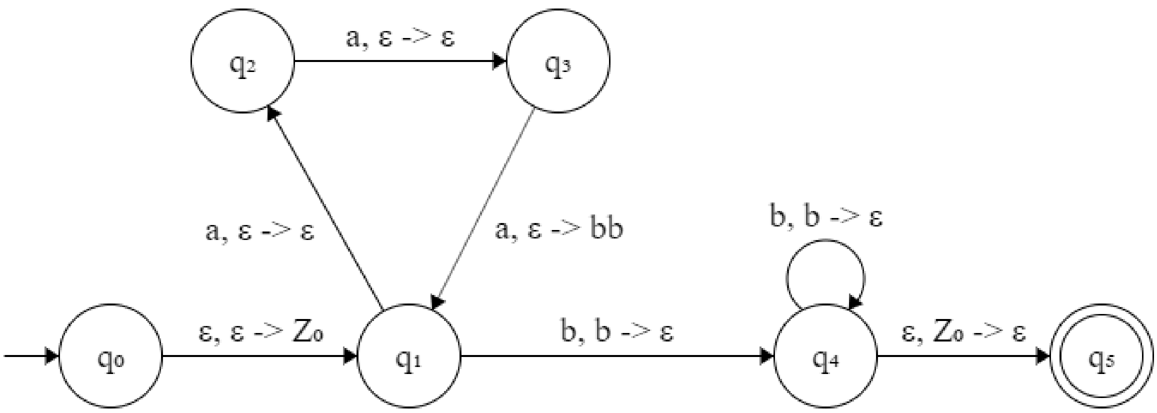
A -> S | ε.

Επειδή έχουμε $n\geq 1$ πρέπει η γραμματική να παράγει το ελάχιστο aaabb που το κάνει με την εξής παραγωγή: S -> aaaAbb -> aaaεbb -> aaabb.

Σε διαφορετική περίπτωση όπου A βάζει πάλι το S αυξάνοντας ουσιαστικά το n κατά 1, δηλαδή πηγαίνοντας από 3a/2b σε 6a/4b κτλ.

Θα αποδείξω ότι η γλώσσα αυτή είναι μη-κανονική με τη χρήση του Pumping Lemma, επομένως δεν θα υπάρχει κάποιο πεπερασμένο αυτόματο που να την αποδέχεται. Υποθέτω ότι είναι κανονική. Παίρνω σπάσιμο xyz: $x = a^{3n-k}$, $y = a^k$, $z = b^{2n}$. Από το PL για κάθε $i\geq 0$ πρέπει $xy^iz \in L$. Όμως $xy^iz = a^{3n+k(i-1)}b^{2n}$ και για i=2 έχουμε $xy^2z = a^{3n+k}b^{2n}$, όπου παρατηρούμε ότι δεν ανήκει στη γλώσσα, αφού k>0. Άρα η γλώσσα δεν είναι κανονική.

Θα χρησιμοποιήσω, επομένως, ένα αυτόματο στοίβας για την αποδοχή αυτής της γλώσσας, καθώς είναι μία γλώσσα χωρίς συμφραζόμενα, όπως φαίνεται άλλωστε από τη γραμματική που την παράγει. Ουσιαστικά ο τρόπος λειτουργίας του είναι ότι για κάθε 3 a που θα διαβάζει θα προσθέτουμε 2 b στη στοίβα και για κάθε b που διαβάζει θα αφαιρεί και 1 b από τη στοίβα. Έτσι αν η είσοδος έχει αναλογία $a^{3n}b^{2n}$ στο τέλος η στοίβα θα είναι κενή και θα το αυτόματο θα αποδέχεται.



γ) Η γλώσσα του ερωτήματος αυτού παράγεται από την εξής γραμματική:

G: S -> 0S1 | 0S0 | 1S0 | 1S1 | 111.

Η γραμματική αυτή παράγει τόσο την ελάχιστη λέξη 111 όσο και κάθε άλλη που ανήκει στη γλώσσα αυτή, ξεκινώντας με τους χαρακτήρες απέξω (αυτούς που βρίσκονται στα άκρα της λέξης) και συνεχίζοντας προς τα μέσα μέχρι να τοποθετήσει το 111.

Άσκηση 5:

Έστω γλώσσα χωρίς συμφραζόμενα L = L(G), όπου G = (V, T, P, S) μία γραμματική χωρίς συμφραζόμενα. Θα κατασκευάσουμε την G^R = (V, T, P^R, S), όπου P^R είναι η αντίστροφη κάθε παραγωγής της P. Δηλαδή, αν A -> a είναι μια παραγωγή της G, τότε A -> a^R είναι μια παραγωγή της G^R. Εύκολα φαίνεται πως μία λέξη w παράγεται από την G αν και μόνον αν η λέξη w^R παράγεται από την G^R. Επομένως έχουμε ότι L(G^R) = L^R και άρα η L^R είναι μια γλώσσα χωρίς συμφραζόμενα, αφού η G^R είναι μια γραμματική χωρίς συμφραζόμενα.

Άσκηση 7:

Αρχή

Κατασκεύασε έναν πίνακα A[n], όπου n ο συνολικός αριθμός των μεταβλητών του input.

Αρχικοποίησε όλες τις θέσεις του A ως ‘E’.

Διάτρεξε το input μέχρι να βρεις ‘)’ :

    Αν είναι x_i:

        Αν A[i] == ‘E’ τότε A[i] == ‘T’.

        Αλλιώς αν A[i] == ‘F’:

            Θέσε όλες τις θέσεις του A ως ‘E’ και συνέχισε να διατρέχεις το input μέχρι να βρεις ‘(‘.

            Αν δεν βρεις ‘(‘ τύπωσε ‘OXI’ και λήξε το πρόγραμμα.

    Αν είναι -x_i:

        Αν A[i] == ‘E’ τότε A[i] = ‘F’.

        Αλλιώς αν A[i] == ‘T’:

            Θέσε όλες τις θέσεις του A ως ‘E’ και συνέχισε να διατρέχεις το input μέχρι να βρεις ‘(‘.

            Αν δεν βρεις ‘(‘ τύπωσε ‘OXI’ και λήξε το πρόγραμμα.

Διάτρεξε τον πίνακα A και αν A[i] == ‘E’ τότε A[i] = ‘F’.

Τύπωσε ‘NAI’ και τύπωσε και τον πίνακα A.

Τέλος

Ο παραπάνω αλγόριθμος ελέγχει ουσιαστικά εάν υπάρχει έστω μία σύζευξη ν μεταβλητών χωρίς να περιέχει τις μεταβλητές x_i και $-x_i$ ταυτόχρονα. Σε αυτήν την περίπτωση τυπώνει 'ΝΑΙ' και τυπώνει και τις τιμές του πίνακα που είναι ορισμένες έτσι ώστε να ικανοποιείται ο τύπος, έχοντας συμπληρώσει με 'F' τις αδιάφορες τιμές. Σε περίπτωση που δεν καταφέρει να βρει μια τέτοια σύζευξη, δηλαδή εάν σε κάθε σύζευξη υπάρχει τουλάχιστον ένα ζευγάρι μεταβλητών x_i και $-x_i$, τότε τυπώνει ΌΧΙ'. Στο ενδιάμεσο, κάθε φορά που μία σύζευξη περιέχει τουλάχιστον ένα τέτοιο ζευγάρι, αδειάζει τον πίνακα θέτοντας όλες τις θέσεις του ίσες με 'E' και αναζητάει την επόμενη σύζευξη, μέχρι είτε να βρει μια που να ικανοποιεί τον τύπο ή μέχρι να μην υπάρχει μία τέτοια και να λήξει το πρόγραμμα.

Άσκηση 8:

α) Ουσιαστικά πρέπει να αποδείξουμε ότι δοσμένης μίας λύσης σε ένα Clique πρόβλημα, μπορούμε σε πολυωνυμικό χρόνο να το υποβιβάσουμε σε κάποιο άλλο NP-πλήρες πρόβλημα, συγκεκριμένα το NP-πλήρες Independent Set πρόβλημα. Είναι προφανές ότι εάν ένας γράφος G έχει independent set x-κορυφών τότε ο συμπληρωματικός του γράφος G' έχει clique x-κορυφών. Επομένως, για να αποδείξουμε εάν ένας γράφος έχει όντως μία κλίκα x-κορυφών, αρκεί να φτιάξουμε τον συμπληρωματικό του γράφο και να ελέγξουμε εάν έχει independent set x-κορυφών. Η διαδικασία παραγωγής του συμπληρωματικού γράφου είναι πολυωνυμικής τάξης, καθώς δεδομένου ενός γράφου και του πίνακα γειτνίασης του διατρέχουμε όλες τις θέσεις του πίνακα αντιστρέφοντας το περιεχόμενο από 0 σε 1 και από 1 σε 0 -> $O(N^2)$. Επομένως, σε πολυωνυμικό χρόνο μπορώ να υποβιβάσω το clique πρόβλημα σε independent set πρόβλημα το οποίο ως γνωστόν σε πολυωνυμικό χρόνο μπορεί να επιβεβαιώσει τη λύση, αφού είναι NP-πλήρες. Άρα το clique πρόβλημα είναι και αυτό NP-πλήρες.