1η Σειρά Ασκήσεων FoCS

Γιάννης Πολυχρονόπουλος 03121089

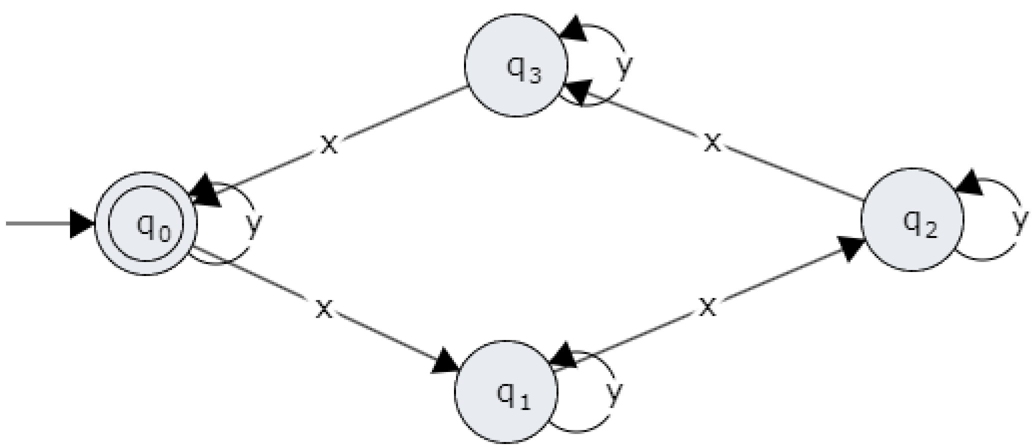
Άσκηση 1.

Κατασκευάστε DFA, κανονικές παραστάσεις και κανονικές γραμματικές για κάθε μία από τις παρακάτω γλώσσες:

(α) Σύνολο συμβολοσειρών του Σ1 = {x, y} των οποίων το πλήθος των ' x' είναι πολλαπλάσιο του 4.

(β) Σύνολο συμβολοσειρών του Σ2 = {a, b} που δεν περιέχουν δύο συνεχόμενα ' ab' .

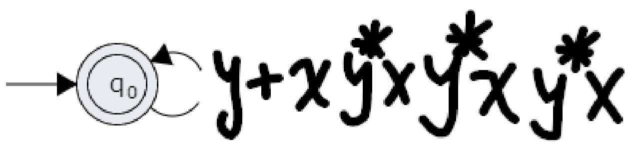
(α) DFA:



Επεξήγηση: Ουσιαστικά στο παραπάνω DFA κάθε κατάσταση αντιπροσωπεύει την πράξη $\#x \; mod 4$.

Τώρα έχουμε για την κανονική παράσταση:

- Διαγράφουμε τα q_1, q_2, q_3



- Διαγράφουμε το q_0

*

$(y+xyxy^*xy^*x)^*$

Για την κανονική γραμματική έχουμε το εξής:

$$G: V\{S, Q_1, Q_2, Q_3, Q_4\}, \qquad T\{x, y, \varepsilon\},$$

P:

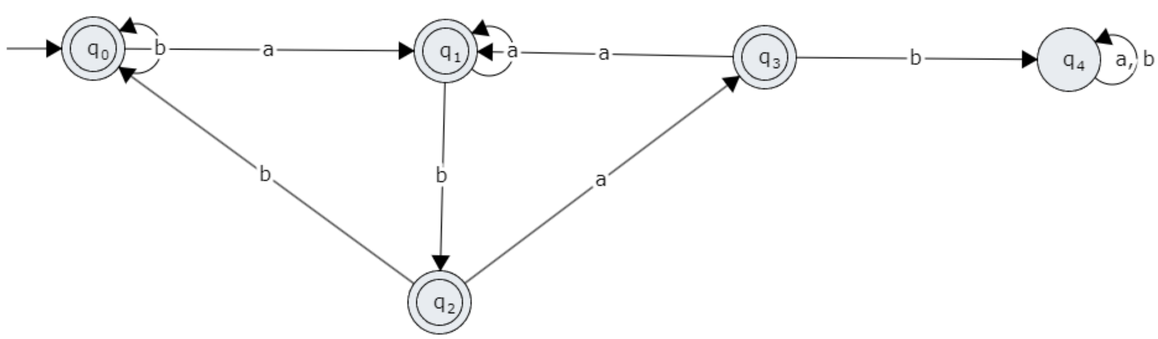
$S \rightarrow \varepsilon | y | xQ_1 | yS$

$Q_1 \rightarrow xQ_2 | yQ_1$

$Q_2 \rightarrow xQ_3 | yQ_2$

$Q_3 \rightarrow xS | yQ_3$

(β) DFA:



Επεξήγηση: Ξεκινάμε από την κατάσταση q_0 και έπειτα όταν διαβάσει 'abab' μέσα στη συμβολοσειρά μας θα καταλήξουμε στη κατάσταση q_4 ,όπου είναι το junk state (or dead state) του DFA μας.

Επομένως οποιαδήποτε συμβολοσειρά που περιέχει δύο συνεχόμενα 'ab' μεταφέρεται στο junk state με αποτέλεσμα να απορρίπτεται.

Τώρα έχουμε για την κανονική παράσταση χρησιμοποιούμε το εξής θεώρημα

$$Arden's\ Theorem: R = Q + RP \Leftrightarrow R = QP^*$$

$$q_0 = \varepsilon + q_2 b + q_0 b \xLeftrightarrow{Arden's\ Theorem} q_0 = (\varepsilon + q_2 b)b^* \ (1)$$

$$q_1 = q_0 a + q_3 a + q_1 a \xLeftrightarrow{Arden's\ Theorem,(1)} q_1 = \big((\varepsilon + q_2 b)b^* + q_3\big)aa^* \ (2)$$

$$q_2 = q_1 b \ (3)$$

$$q_3 = q_2 a \overset{(3)}{\Leftrightarrow} q_3 = q_1 ba \ (4)$$

Το q4 δεν το χρησιμοποιώ διότι είναι junk state. Κατόπιν δουλεύουμε με τις σχέσεις μας.

$(2) \overset{(4),(3)}{\Longleftrightarrow} q_1 = \big((\varepsilon + q_1 bb)b^* + q_1 ba\big)aa^* = b^*aa^* + q_1 bbb^*aa^* + q_1 baaa^* \Leftrightarrow$

$q_1 = b^*aa^* + q_1 b(bb^* + a)aa^* \overset{Arden's\,Theorem}{\Longleftrightarrow} q_1 = b^*a^+(b(bb^* + a)a^+)^*(5)$

$(1) \overset{(3)}{\Leftrightarrow} q_0 = (\varepsilon + q_1 bb)b^* \overset{(5)}{\Leftrightarrow} q_0 = (\varepsilon + b^*a^+(b(bb^* + a)a^+)^*bb)b^*$

$(3) \overset{(5)}{\Leftrightarrow} q_2 = b^*a^+(b(bb^* + a)a^+)^*b$

$(4) \overset{(5)}{\Leftrightarrow} q_3 = b^*a^+(b(bb^* + a)a^+)^*ba$

Επομένως η κανονική παράσταση είναι η εξής:

$q_0 + q_1 + q_2 + q_3 = (\varepsilon + q_1 bb)b^* + q_1 + q_1 b + q_1 ba = b^* + q_1(bbb^* + \varepsilon + b + ba) =$

$= b^* + b^*a^+(b(bb^* + a)a^+)^*(bbb^* + \varepsilon + b + ba) =$

$= \boldsymbol{b}^*[\boldsymbol{\varepsilon} + \boldsymbol{b}^*\boldsymbol{a}^+(\boldsymbol{b}(\boldsymbol{bb}^* + \boldsymbol{a})\boldsymbol{a}^+)^*(\boldsymbol{bb}^+ + \boldsymbol{\varepsilon} + \boldsymbol{b} + \boldsymbol{ba})]$

Κανονική γραμματική:

$$G: V\{S, Q_1, Q_2, Q_3\}, \qquad T\{a, b, \varepsilon\}$$

P:

$S \rightarrow \varepsilon|b|bS|aQ_1$

$Q_1 \rightarrow \varepsilon|a|aQ_1|bQ_2$

$Q_2 \rightarrow \varepsilon|aQ_3|bQ_1$

$Q_3 \rightarrow \varepsilon|aQ_1$

Άσκηση 2.

Σχεδιάστε δύο DFA με αλφάβητο Σ = {0, 1, 2} που να διαβάζουν τα ψηφία ενός ακεραίου αριθμού n (τα περισσότερο σημαντικά πρώτα, τα λιγότερο σημαντικά τελευταία) δοσμένου σε τριαδικό σύστημα και να αποδέχονται ως εξής:

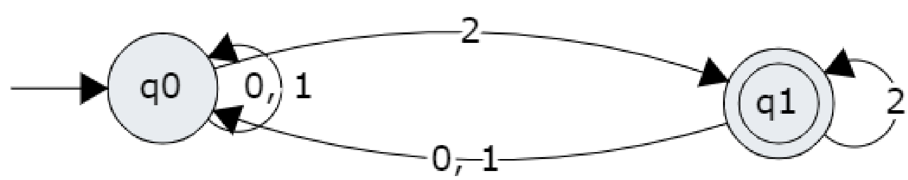
(α) το πρώτο εάν n mod 3 = 2,

(β) το δεύτερο εάν n mod 5 = 0.

Δώστε επίσης τις αντίστοιχες κανονικές παραστάσεις.

Σημείωση: Μπορεί να υπάρχουν αρχικά μηδενικά στον αριθμό εισόδου. Για παράδειγμα, η είσοδος 00202 θα πρέπει να γίνεται αποδεκτή και από τα δύο αυτόματα ενώ η 120 να γίνεται αποδεκτή μόνο από το δεύτερο.

(α) DFA:



Επεξήγηση: Έστω ένας αριθμός στο τριαδικό σύστημα $n = n_\kappa \dots n_0$ όπου $n_\kappa \dots n_0$ τα ψηφία του. Στο δεκαδικό σύστημα ο αριθμός n θα είναι $3^\kappa n_\kappa + 3^{\kappa-1} n_{k-1} + \cdots + 3^0 n_0$. Παρατηρούμε πως το $n \ mod \ 3$ θα ισούται με το λιγότερο σημαντικό ψηφίο του τριαδικού αριθμού, καθώς όλα τα υπόλοιπα στοιχεία του αθροίσματος είναι πολλαπλάσια του 3. Μπορώ να τον γράψω ως εξής:

$$3(3^{\kappa-1} n_\kappa + 3^{\kappa-2} n_{k-1} + \cdots + n_1) + \ n_0$$

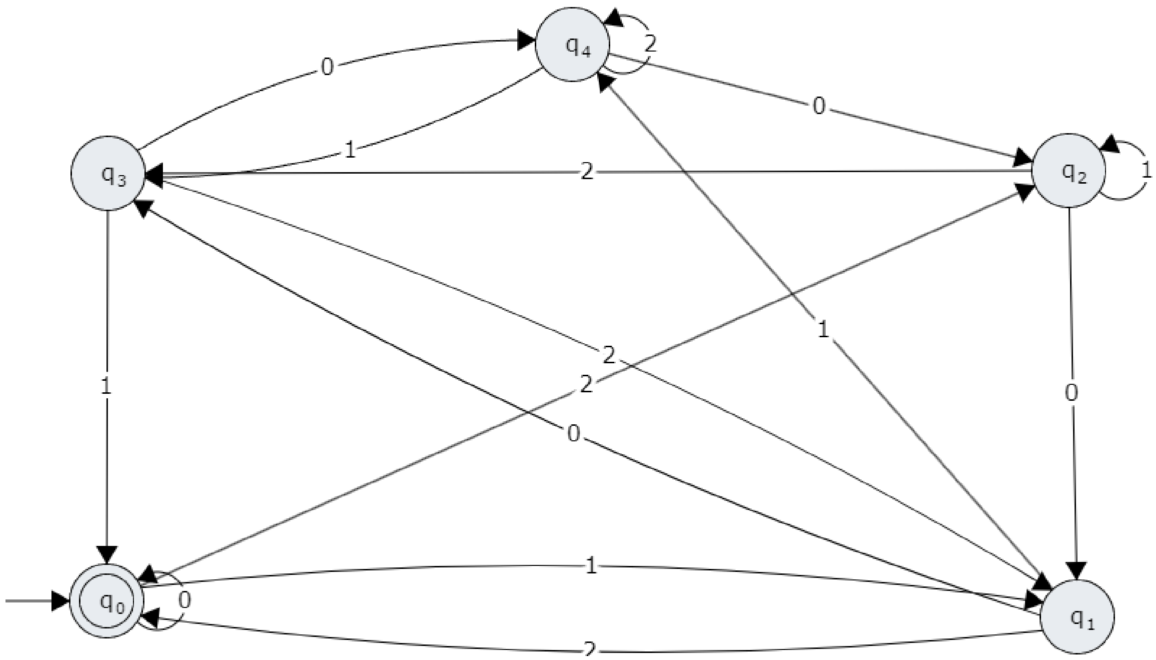
Άρα φτιάχνουμε ένα DFA όπου αποδέχεται τριαδικούς αριθμούς με το ελάχιστα σημαντικό ψηφίο να είναι 2.

Για την κανονική παράσταση:

$$q_1 = (0 + 1 + 2)^* 2$$

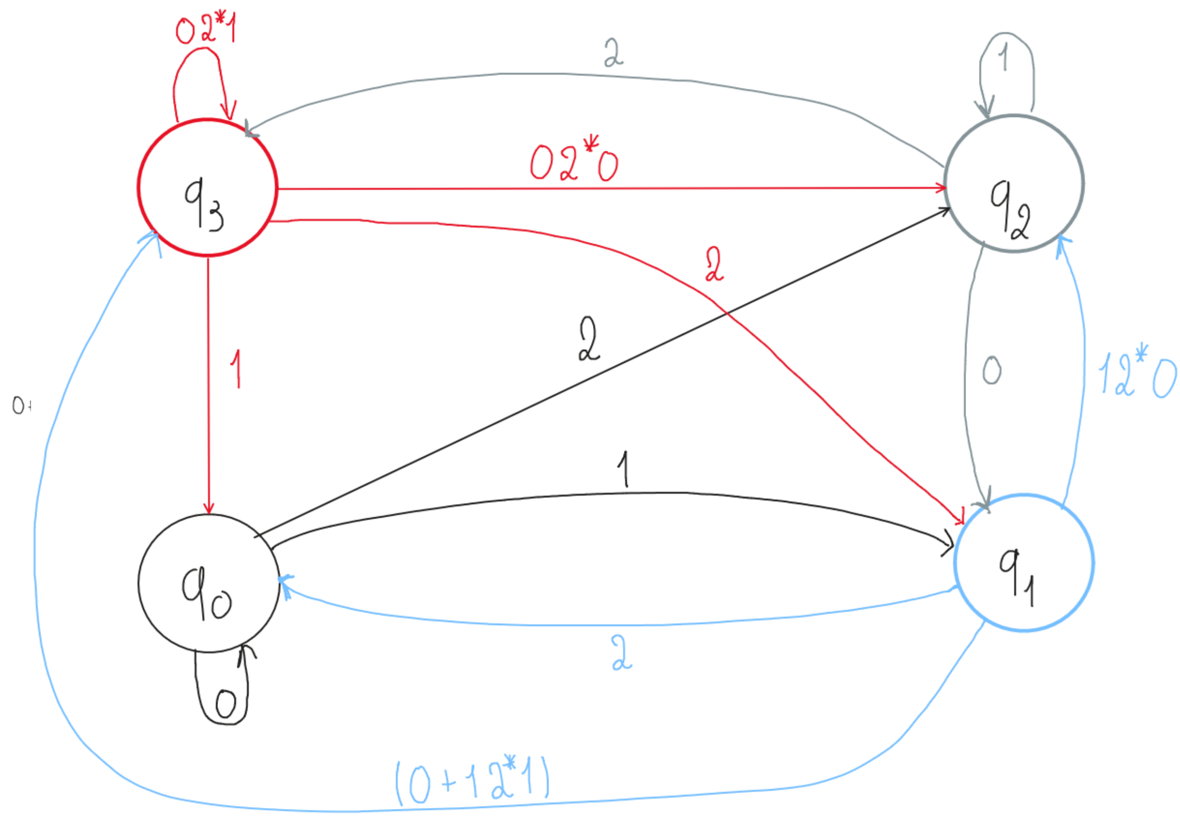
Αυτή η κανονική παράσταση αντιπροσωπεύει κάθε τριαδικό αριθμό όπου το λιγότερο σημαντικό ψηφίο είναι 2.

(β) DFA:

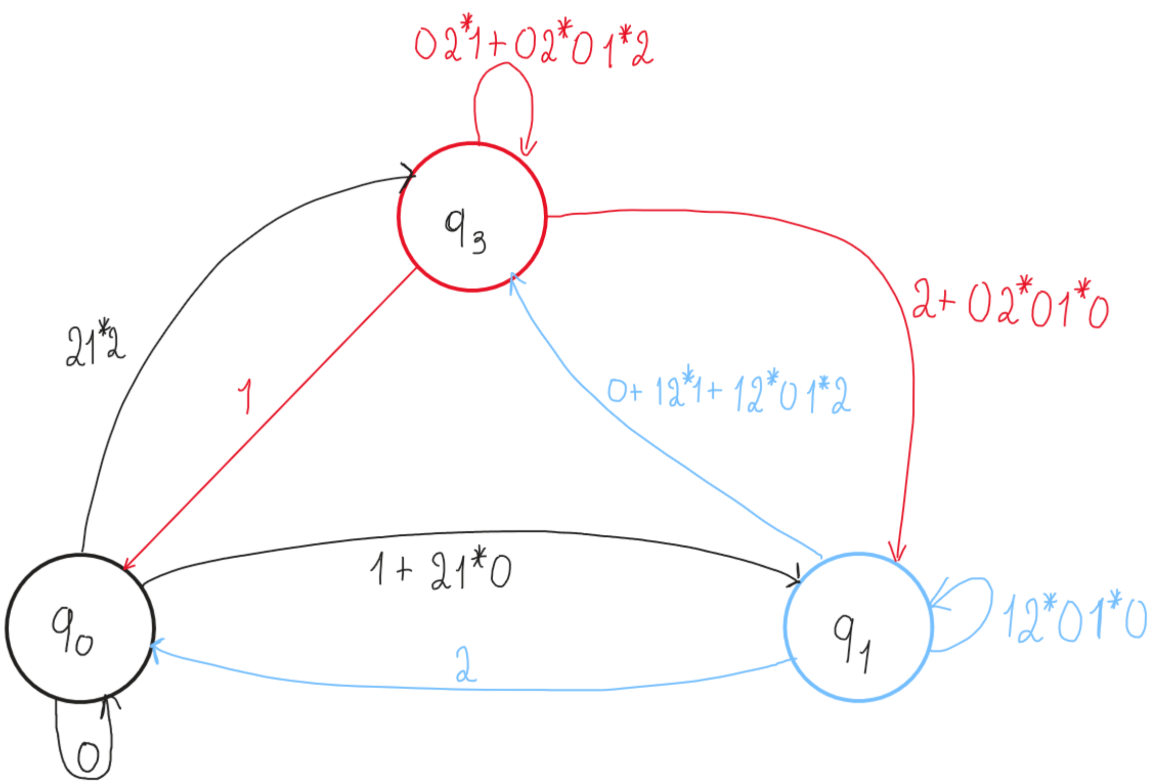


Επεξήγηση: Οι καταστάσεις $\{q_0, q_1, q_2, q_3, q_4\}$ αντιπροσωπεύουν το υπόλοιπο της διαίρεσης n/5. Επομένως για να αποδεχόμαστε έναν αριθμό n πρέπει να καταλήγει στην κατάσταση q_0 που σημαίνει $n \; mod \; 5 = 0$.

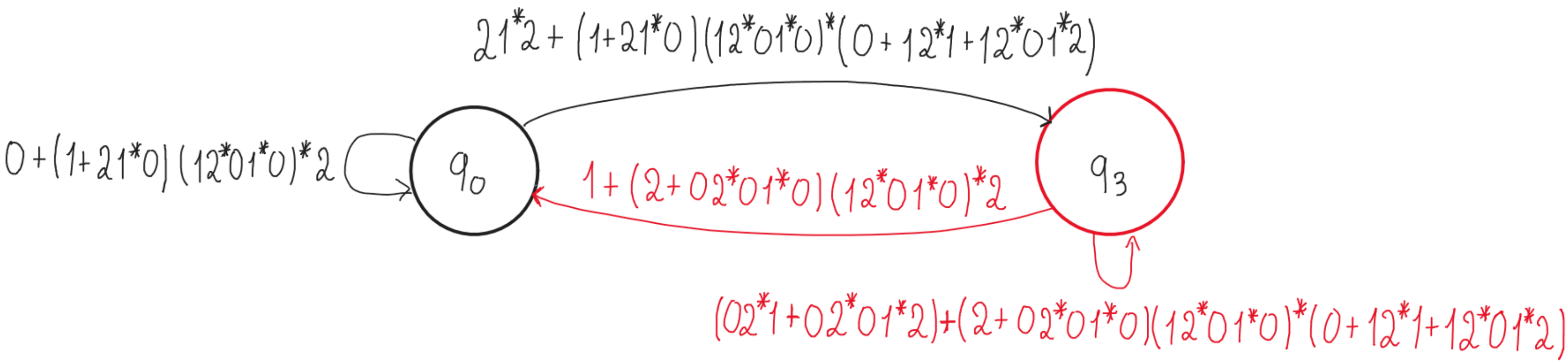
Base 10	Base 3	Remainder(%5)	Beginning-State	End-State	Transition
0	0	0	q_0	q_0	$\delta(q_0, 0) = q_0$
1	1	1	q_0	q_1	$\delta(q_0, 1) = q_1$
2	2	2	q_0	q_2	$\delta(q_0, 2) = q_2$
3	10	3	q_1	q_3	$\delta(q_1, 0) = q_3$
4	11	4	q_1	q_4	$\delta(q_1, 1) = q_4$
5	12	0	q_1	q_0	$\delta(q_1, 2) = q_0$
6	20	1	q_2	q_1	$\delta(q_2, 0) = q_1$
7	21	2	q_2	q_2	$\delta(q_2, 1) = q_2$
8	22	3	q_2	q_3	$\delta(q_2, 2) = q_3$
9	100	4	q_3	q_4	$\delta(q_3, 0) = q_4$
10	101	0	q_3	q_0	$\delta(q_3, 1) = q_0$
11	102	1	q_3	q_1	$\delta(q_3, 2) = q_1$
12	110	2	q_4	q_2	$\delta(q_4, 0) = q_2$
13	111	3	q_4	q_3	$\delta(q_4, 1) = q_3$
14	112	4	q_4	q_4	$\delta(q_4, 2) = q_4$

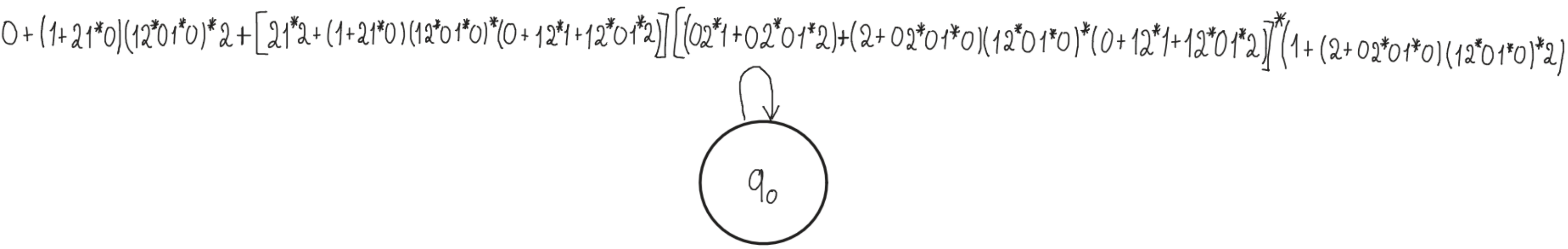


02*1
2
1
02*0
q3
q2
2
2
1
0
12*0
0+
1
q0
q1
2
0
(0+12*1)



02*1+02*01*2
q3
2+02*01*0
21*2
1
0+12*1+12*01*2
1+21*0
q0
q1
12*01*0
2
0





$$\{0+(1+21^*0)(12^*01^*0)^*2+[21^*2+(1+21^*0)(12^*01^*0)^*(0+12^*1+12^*01^*2)][(02^*1+02^*01^*2)+(2+02^*01^*0)(12^*01^*0)^*(0+12^*1+12^*01^*2)]^*(1+(2+02^*01^*0)(12^*01^*0)^*2)\}^*$$

Άσκηση 3.

Δίνονται οι παρακάτω γλώσσες:

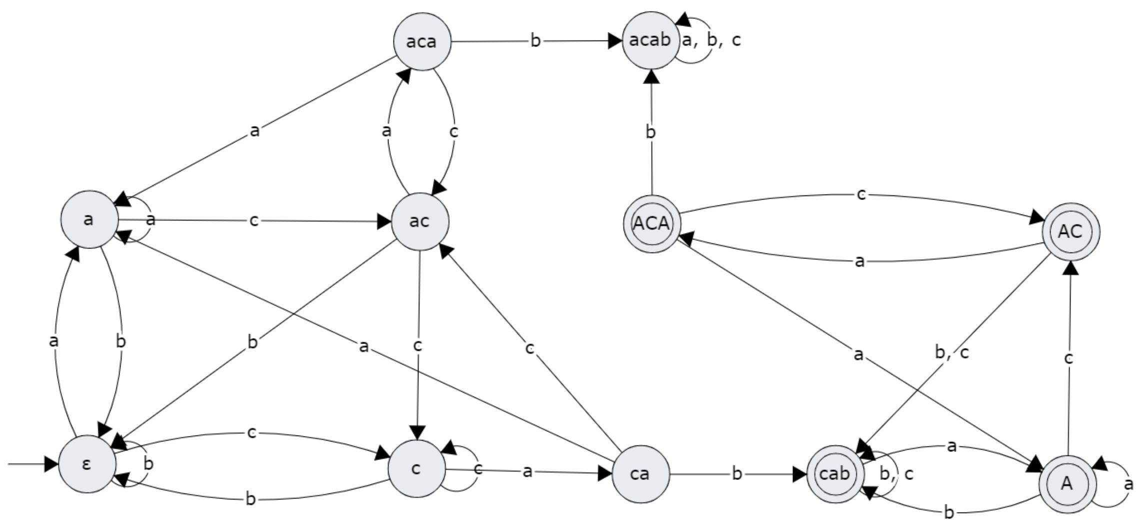
$L_1 = \{w \in \{a, b, c\}^* |$ η w περιέχει την συμβολοσειρά ′ cab′ αλλά όχι τη συμβολοσειρά ′ acab′$\}$

$L_2 = \{w \in \{0, 1\}^* |$ η w είναι δυαδική αναπαράσταση ακεραίου με τιμή 4^{κ}, για $k \geq 1\}$

Κατασκευάστε DFA με όσο το δυνατόν λιγότερες καταστάσεις για τις γλώσσες L_1 και L_2. Αποδείξτε την ελαχιστότητα του αυτομάτου σας.

Υπόδειξη: θυμηθείτε ότι μπορείτε να σχεδιάσετε και να συνδυάσετε αυτόματα για απλούστερες γλώσσες (με τουλάχιστον δύο τρόπους).

- Για την L_1 DFA:

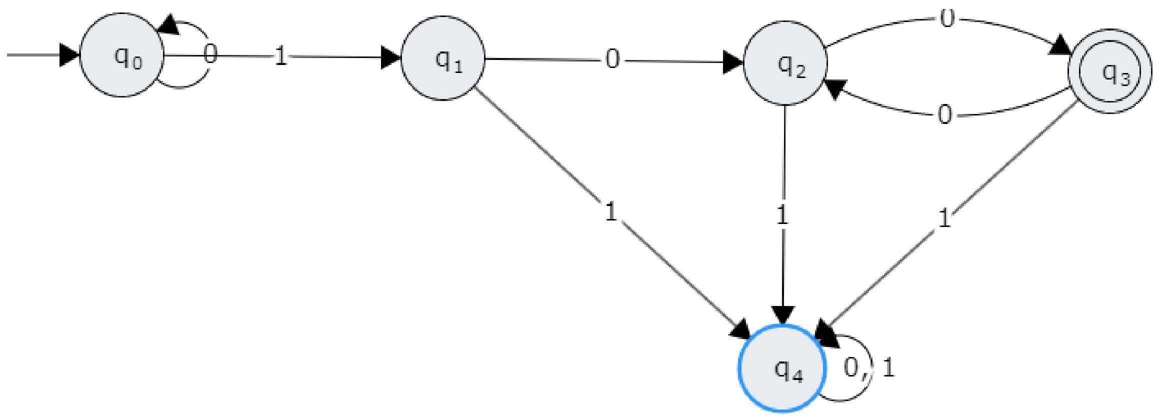


Επεξήγηση: Το παραπάνω DFA απορρίπτει τις συμβολοσειρές που περιέχουν ‘acab’ με τη χρήση της κατάστασης acab που αποτελεί junk state, ενώ παράλληλα, αποδέχεται τις συμβολοσειρές που περιέχουν ‘cab’ μέσω της κατάστασης cab που αποτελεί τελική κατάσταση.

	ε	c	ca	cab	A	AC	ACA	$acab$	aca	ac
c	X_2									
ca	X_1	X_1								
cab	X_0	X_0	X_0							
A	X_0	X_0	X_0	X_3						
AC	X_0	X_0	X_0	X_2	X_2					
ACA	X_0	X_0	X_0	X_1	X_1	X_1				
$acab$	X_3	X_2	X_1	X_0	X_0	X_0	X_0			
aca	X_3	X_2	X_1	X_0	X_0	X_0	X_0	X_4		
ac	X_4	X_2	X_1	X_0	X_0	X_0	X_0	X_3	X_3	
a	X_3	X_2	X_1	X_0	X_0	X_0	X_0	X_3	X_3	X_4

Παρατηρούμε πως ο πίνακάς μας δεν έχει κενά. Επομένως το παραπάνω DFA είναι ελαχιστοποιημένο.

- Για την L_2 DFA:



Επεξήγηση: Οι δυαδικοί αριθμοί που αναπαριστούν ακέραιους με τιμή $4^{\kappa} = 2^{2\kappa}$, για $k \geq 1$ είναι της μορφής: $10 \ldots 0$ με 2k μηδενικά. Επομένως το DFA μας πρέπει να απορρίπτει οποιαδήποτε άλλη περίπτωση, ιδιότητα που ικανοποιεί το παραπάνω DFA με την χρήση της κατάστασης q_4, που αποτελεί junk state.

Κατασκευάζουμε τον τριγωνικό πίνακα αποσκοπώντας την ελαχιστοποίηση του DFA.

	q_0	q_1	q_2	q_3
q_1	X_2			
q_2	X_1	X_1		
q_3	X_0	X_0	X_0	
q_4	X_3	X_2	X_1	X_0

Παρατηρούμε πως ο πίνακάς μας δεν έχει κενά. Επομένως το παραπάνω DFA είναι ελαχιστοποιημένο.

Άσκηση 4.

Είναι κανονικές οι παρακάτω γλώσσες; Αν μια γλώσσα δεν είναι κανονική, να το αποδείξετε χρησιμοποιώντας είτε το Λήμμα Άντλησης είτε κάποια ιδιότητα κλειστότητας είτε συνδυασμό τους. Αν μια γλώσσα είναι κανονική, να γράψετε κανονική γραμματική ή να σχεδιάσετε NFA/DFA.

(α) Το συμπλήρωμα της γλώσσας $L_1 = \{(\alpha^n b^2)^2 : n \in \mathbb{N}^*\}$

(β) Το συμπλήρωμα της γλώσσας $L_2 = \{a^n b^m c^k : n, m, k \in \mathbb{N}^* \text{ τέτοιοι ώστε } n \neq m \text{ και } n \neq k\}$

(γ) $L_3 = \{wcx : x \in \{a, b\}^* \text{ και } w \text{ υποσυμβολοσειρά της } x\}$

(α) $L_1 = \{(\alpha^n b^2)^2 : n \in \mathbb{N}^*\}$

Θεωρούμε πως η L_1 είναι κανονική γλώσσα. Τότε από το Pumping Lemma:

- Το pumping Length υπάρχει και έστω πως είναι n.
- Διαλέγω από την γλώσσα L_1 κατάλληλο string $z = a^n b^2 a^n b^2 \text{ με } |z| = 2n + 4 > n$
- Χωρίζω το z σε uvw με $|uv| \leq n \text{ και } |v| > 0$.
- Συγκεκριμένα, $u = a^{l_1}, v = a^{l_2}, w = a^{n-l_1-l_2} b^2 a^n b^2$
- Για να ανήκει το $uv^i w$ στην L_1 πρέπει: $l_1 + i \cdot l_2 + n - l_1 - l_2 = n \Leftrightarrow (i-1)l_2 = 0$

- $\xLeftrightarrow{\text{Για } i \neq 1} l_2 = 0$ άτοπο διότι πρέπει $|v| > 0 \Leftrightarrow l_2 > 0$
- Επομένως η L_1 δεν είναι κανονική γλώσσα και συνεπώς ούτε το συμπλήρωμά της.

(β) $L_2' = \{a^n b^m c^k : n, m, k \in \mathbb{N} \text{ τέτοιοι ώστε } n = m \text{ ή } n = k\}$

Θεωρούμε πως η L_2' είναι κανονική γλώσσα. Τότε από το Pumping Lemma:

Το pumping Length υπάρχει και έστω πως είναι n.

Έχω 3 περιπτώσεις

(1) Για $n = m$ και $n = k$

- Διαλέγω από την γλώσσα L_2' κατάλληλο string $z = a^n b^n c^n \text{ με } |z| = 3n > n$
- Χωρίζω το z σε uvw με $|uv| \leq n \text{ και } |v| > 0$.
- Συγκεκριμένα, $u = a^{l_1}, v = a^{l_2}, w = \alpha^{n-l_1-l_2} b^n c^n$

- Για να ανήκει το $uv^i w$ στην L_2' πρέπει: $l_1 + i \cdot l_2 + n - l_1 - l_2 = n \Leftrightarrow (i-1)l_2 = 0$
- $\xLeftrightarrow{\text{Για } i \neq 1} l_2 = 0$ άτοπο διότι πρέπει $|v| > 0 \Leftrightarrow l_2 > 0$

(2) Για $n = m$ και $n \neq k$

- Διαλέγω από την γλώσσα L_2' κατάλληλο string $z = a^n b^n c^\kappa \text{ με } |z| = 2n + \kappa > n$
- Χωρίζω το z σε uvw με $|uv| \leq n \text{ και } |v| > 0$.
- Συγκεκριμένα, $u = a^{l_1}, v = a^{l_2}, w = \alpha^{n-l_1-l_2} b^n c^\kappa$

- Για να ανήκει το $uv^i w$ στην L_2' πρέπει: $l_1 + i \cdot l_2 + n - l_1 - l_2 = n \Leftrightarrow (i-1)l_2 = 0$
- $\xLeftrightarrow{\text{Για } i \neq 1} l_2 = 0$ άτοπο διότι πρέπει $|v| > 0 \Leftrightarrow l_2 > 0$

(3) Για $n \neq m$ και $n = k$

- Διαλέγω από την γλώσσα L_2' κατάλληλο string $z = a^n b^m c^n$ $με$ $|z| = 3n > n$
- Χωρίζω το z σε uvw με $|uv| \leq n$ $και$ $|v| > 0$.
- Συγκεκριμένα, $u = a^{l_1}, v = a^{l_2}, w = \alpha^{n-l_1-l_2} b^m c^n$

- Για να ανήκει το $uv^i w$ στην L_2' πρέπει: $l_1 + i \cdot l_2 + n - l_1 - l_2 = n \Leftrightarrow (i-1)l_2 = 0$
- $\overset{Για\ i\neq 1}{\Longleftrightarrow} l_2 = 0$ άτοπο διότι πρέπει $|v| > 0 \Leftrightarrow l_2 > 0$

Επομένως η L_2' δεν είναι κανονική γλώσσα.

Ας πάρουμε τώρα την γλώσσα $L = \{w \in \{a, b, c\}^* : w = a^+ b^+ c^+\}$ που είναι κανονική. Τότε:

$L_2' = L \cap L_2^c = μη\ κανονική\ γλώσσα.$

Άρα το συμπλήρωμα της L_2($δηλαδη$ L_2^c) δεν είναι κανονική γλώσσα.

(γ)Θεωρούμε πως η L_3 είναι κανονική γλώσσα. Τότε από το Pumping Lemma:

- Το pumping Length υπάρχει και έστω πως είναι n.
- Διαλέγω από την γλώσσα L_3 κατάλληλο string $z = a^n c\, a^n b^n$ $με$ $|z| = 3n + 1 > n$
- Χωρίζω το z σε uvw με $|uv| \leq n$ $και$ $|v| > 0$.
- Συγκεκριμένα, $u = a^{l_1}, v = a^{l_2}, w = a^{n-l_1-l_2} c a^n b^n$

- Για να ανήκει το $uv^i w$ στην L_2' πρέπει: $l_1 + i \cdot l_2 + n - l_1 - l_2 \leq n \Leftrightarrow (i-1)l_2 \leq 0$
- $\overset{Για\ i\neq 1}{\Longleftrightarrow} l_2 \leq 0$ άτοπο διότι πρέπει $|v| > 0 \Leftrightarrow l_2 > 0$
- Επομένως η L_3 δεν είναι κανονική γλώσσα.

Άσκηση 5.

Περιγράψτε σε φυσική γλώσσα τη γλώσσα που παράγει καθεμιά από τις παρακάτω γραμματικές:

(α) G_1: S → aA, A → a | aA | B, B → bb | bbB

(β) G_2: S → aSb | bU | U a, U → bU | aU | ε

(α)$G_1 = \{a^k b^{2n} \big| \, k \geq 1, n \geq 0 \ με\ k, n \in \mathbb{N}\}$

Η παραπάνω γραμματική παράγει μια γλώσσα που περιέχει στην αρχή τουλάχιστον ένα ‘a’ και στη συνέχεια είτε καθόλου ‘b’ είτε ένα άρτιο πλήθος ‘b’.

(β) $G_2 = \{a^k (b(a+b)^* + (a+b)^* a) b^k \big| \, k \geq 0\, \}$

Η παραπάνω γραμματική παράγει μια γλώσσα η οποία αρχίζει και τελειώνει με τον ίδιο αριθμό χαρακτήρων ‘a’ και ‘b’ αντίστοιχα (μπορεί να είναι και 0). Στη συνέχεια, έχουμε δύο επιλογές, είτε να αρχίζουμε με τον χαρακτήρα ‘b’ και έπειτα να προσθέσουμε το ‘a’ ή το ‘b’ όσες φορές θέλουμε (μπορούμε και καμία), είτε να προσθέσουμε πρώτα το ‘a’ ή το ‘b’ όσες φορές θέλουμε και μετά να προσθέσουμε στο τέλος τον χαρακτήρα ‘a’.

Άσκηση 6.

Να διατυπώσετε γραμματικές χωρίς συμφραζόμενα για τις παρακάτω γλώσσες:
(α) $L = \{a^n b^k a^m b: n, m, k \in \mathbb{N}^* \text{ τέτοιοι ώστε } n + m \text{ περιττός}\}$
(β) Το συμπλήρωμα της L

(α) $L: V\{S, A, B\}, T\{a, b, \varepsilon\}$

P:

$$S \to aAb$$

$$A \to aBa|Baa|aaA|Aaa|aAa$$

$$B \to bB|b$$

(β) $L' = \{a^n b^k a^m b: n, m, k \in \mathbb{N}^* \text{ τέτοιοι ώστε } n + m \text{ άρτιος}\}$

$$L': V\{S, A, B\}, T(a, b, \varepsilon)$$

$$S \to Ab$$

$$A \to aBa|aaA|Aaa|aAa$$

$$B \to bB|b$$

Άσκηση 7. (Υπολογισιμότητα)

Αποδείξτε ότι το πρόβλημα του ελέγχου αν ένα πρόγραμμα τερματίζει με είσοδο την κενή συμβολοσειρά είναι μη επιλύσιμο.

Υπόδειξη: Δείξτε ότι αν υπάρχει πρόγραμμα Π_ε που να παίρνει σαν είσοδο ένα οποιοδήποτε πρόγραμμα Π' και να αποφαίνεται αν το Π' τερματίζει με είσοδο την κενή συμβολοσειρά, τότε θα μπορούσαμε να λύσουμε το Πρόβλημα Τερματισμού τροφοδοτώντας το Π_ε με κατάλληλη είσοδο Π' . Συγκεκριμένα δείξτε πώς, από μια οποιαδήποτε είσοδο (Π, x) του Προβλήματος Τερματισμού, όπου το ερώτημα είναι αν το Π με είσοδο x τερματίζει, μπορούμε να κατασκευάσουμε κατάλληλο Π'

Λύση:

Έστω πως υπάρχει πρόγραμμα Π_ε τέτοιο ώστε να παίρνει σαν είσοδο ένα οποιοδήποτε πρόγραμμα Π' και να αποφαίνεται αν το πρόγραμμα αυτό τερματίζει με είσοδο την κενή συμβολοσειρά.

Ας πάρουμε τώρα το Πρόβλημα Τερματισμού με είσοδο (Π', ε), όπου ε η κενή συμβολοσειρά. Τότε το Πρόβλημα Τερματισμού ανάγεται στο πρόγραμμα Π_ε ,δηλαδή ελέγχει εάν ένα πρόγραμμα Π' με είσοδο ε τερματίζει ή όχι. Συνεπώς επειδή το Πρόβλημα Τερματισμού είναι μη επιλύσιμο το ίδιο θα ισχύει και για το πρόβλημα της εκφώνησης.

Άσκηση 8. (Λογική και Αλγόριθμοι)

Διατυπώστε αποδοτικό αλγόριθμο που να δέχεται σαν είσοδο οποιονδήποτε τύπο της προτασιακής λογικής σε μορφή Horn και να αποφαίνεται αν είναι ικανοποιήσιμος. Σε περίπτωση που είναι θα πρέπει να επιστρέφει μία ανάθεση αληθοτιμών που ικανοποιεί τον τύπο. Π.χ. με είσοδο $(x_1 \vee \neg x_2 \vee \neg x_3) \wedge (\neg x_2 \vee x_3) \wedge (\neg x_4 \vee x_5 \vee \neg x_6) \wedge (x_6)$ θα πρέπει να επιστρέφει 'Ναι' και μία από τις αναθέσεις αληθοτιμών στις (x1, . . . , x6) που ικανοποιούν τον τύπο, π.χ. την ανάθεση (True, False,True, False,True,True) ενώ με είσοδο $(\neg x_1 \vee \neg x_2) \wedge (x_1 \vee \neg x_3) \wedge (\neg x_2 \vee x_3) \wedge (x_2)$ θα πρέπει να επιστρέφει ‘Όχι’. Θεωρήστε ότι όλες οι μεταβλητές ενός τύπου δίνονται στη μορφή x_n, όπου n ένας φυσικός αριθμός.

Λύση:

Αρχικά παρατηρούμε πως μια πρόταση Horn αποτελεί τη σύζευξη φράσεων με το πολύ ένα θετικό λέκτημα. Επομένως, για να είναι satisfiable μια πρόταση Horn πρέπει κάθε φράση να είναι True. Τα είδη των φράσεων είναι τα εξής:

$(x_1 \vee \neg x_2 \vee \dots \vee \neg x_n) \equiv (x_2 \wedge \dots \wedge x_n \to x_1)$ (1)

$x_0 \equiv (True \to x_0)$ (2)

$(\neg x_1 \vee \dots \vee \neg x_n) \equiv (x_1 \wedge \dots \wedge x_n) \to False$ (3)

Βήμα 1: Αρχικά, θέτουμε όλα τα λεκτήματα ίσα με False.

Βήμα 2: Οποιαδήποτε φράση που περιέχει μόνο ένα θετικό λέκτημα και τίποτα άλλο, δηλαδή είναι της μορφής (2) (έστω x_0) τότε αυτό το λέκτημα πρέπει να το αναθέσουμε ίσο με True.

Βήμα 3: Εάν υπάρχει στη πρότασή μας φράση με μόνο το αρνητικό λέκτημα που θέσαμε True, δηλαδή το $\neg x_0$, τότε η πρόταση είναι unsatisfiable και εκτυπώνουμε «Όχι».

Βήμα 4: Αλλιώς, αν υπάρχει φράση που περιέχει το x_0, τότε διαγράφουμε την φράση από την πρότασή μας, αφού θα είναι της μορφής $(x_1 \wedge \dots \wedge x_n \to True)$ που θα είναι πάντα True ανεξαρτήτως από τις τιμές των αρνητικών λεκτημάτων.

Βήμα 5: Επίσης, αν υπάρχει φράση που περιέχει το $\neg x_0$, τότε μπορούμε να διαγράψουμε το $\neg x_0$ από την φράση, διότι το $\neg x_0$ θα έχει τιμή False και ισχύει: $A \vee False \equiv A$.

Ωστόσο, με την διαγραφή λεκτημάτων μπορεί να προκύψουν και άλλες φράσεις που περιέχουν μόνο ένα θετικό λέκτημα. Επομένως, επαναλαμβάνουμε την παραπάνω διαδικασία από το 2^{o} Βήμα μέχρι να μην υπάρχουν φράσεις με μόνο ένα θετικό λέκτημα.

Εάν μετά από τις επαναλήψεις δεν υπάρχει φράση με μόνο ένα θετικό λέκτημα τότε η πρόταση είναι satisfiable, εκτυπώνουμε «Ναι» και επιστρέφουμε τις μεταβλητές μας.

Pseudocode:

$x_i = False\ \forall i \in [1, n]$

$while\ (exists\ a\ clause\ with\ only\ one\ positive\ literal)$:

{

$find\ all\ clauses\ that\ contain\ only\ one\ positive\ literal\ x_k$;

$if\,(\ there\ exists\ a\ clause\ with\ only\ the\ literal \neg \mathrm{x}_k)$

{

$print($"Όχι"$)$;

$return\ null$;

}

$delete\ any\ clause\ that\ includes\ the\ literal\ x_k$;

$delete\ the\ literal\ \neg \mathrm{x}_k\ in\ any\ clause\ that\ contains\ it$;

$x_k = True$;

}

$print($"Ναι"$)$;

$return\ x_i$;

Άσκηση 9. (Πολυπλοκότητα: αναγωγές προς απόδειξη δυσκολίας)

Έστω ένα μη κατευθυνόμενο γράφημα G = (V, E). Κάλυμμα κορυφών (Vertex Cover) λέγεται ένα υποσύνολο των κορυφών V ′ ⊆ V που περιέχει τουλάχιστον ένα άκρο κάθε ακμής του γραφήματος (λέμε ότι το άκρο αυτό καλύπτει την ακμή.

Ένα σύνολο κορυφών του G λέγεται ανεξάρτητο σύνολο αν κάθε δύο κορυφές του δε συνδέονται με ακμή. 2 Θεωρήστε τα προβλήματα απόφασης:

(i) Independent Set (IS), στο οποίο, δεδομένης εισόδου (G, k) αποδεχόμαστε την είσοδο αν και μόνο αν το γράφημα G περιέχει κάποιο ανεξάρτητο σύνολο μεγέθους τουλάχιστον k, και
(ii) Vertex Cover (VC), στο οποίο, δεδομένης εισόδου (G, r) αποδεχόμαστε την είσοδο αν και μόνο αν το γράφημα G περιέχει κάλυμμα κορυφών μεγέθους το πολύ r.

(α) Δώστε τουλάχιστον 3 παραδείγματα εισόδων για το κάθε πρόβλημα και αποφανθείτε αν η απάντηση για κάθε μία από αυτές είναι θετική ή αρνητική.

(β) Δημιουργήστε τουλάχιστον 4 ζεύγη εισόδων της μορφής (G, k), (G, |V|− k), όπου η πρώτη είσοδος κάθε ζεύγους θα αφορά στο πρόβλημα IS και η δεύτερη στο πρόβλημα VC (δηλαδή σε κάθε ζεύγος εισόδων το γράφημα είναι ίδιο, ενώ τα μεγέθη των αναζητούμενων συνόλων έχουν άθροισμα όσο το πλήθος των κορυφών). Τι παρατηρείτε;

(γ) Δείξτε ότι αν είναι γνωστό ότι το πρόβλημα VC είναι NP-πλήρες, τότε και το πρόβλημα IS είναι NP-πλήρες.

(δ) Μπορείτε να τροποποιήσετε την αναγωγή του (α) για να αποδείξετε και το αντίστροφο, δηλαδή ότι αν IS είναι NP-πλήρες, τότε και το VC είναι NP-πλήρες;

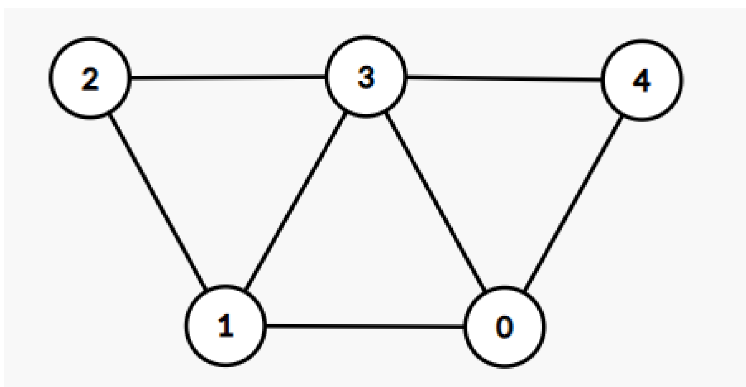
(α)

Για Independent Set (IS) έχουμε:

$$(G, 1) \rightarrow True \mid (G, 2) \rightarrow True \mid (G, 3) \rightarrow False \mid (G, 4) \rightarrow False$$

Για Vertex Cover (VC) έχουμε:

$$(G, 4) \rightarrow True \mid (G, 3) \rightarrow True \mid (G, 3) \rightarrow False \mid (G, 1) \rightarrow False$$



(β) Αν πάρουμε το παραπάνω παράδειγμα, όπου έχουμε 4 ζεύγη εισόδων (G, k), (G, |V|− k) με τη πρώτη είσοδο κάθε ζεύγους να αφορά στο πρόβλημα IS και τη δεύτερη στο πρόβλημα VC, παρατηρούμε πως κάθε ζεύγος έχει ίδια τιμή (True or False).

(γ) Γνωρίζουμε πως το VC πρόβλημα είναι NP-πλήρες. Θα ανάγουμε το πρόβλημα IS στο πρόβλημα VC αποδεικνύοντας έτσι πως το IS είναι NP-πλήρες.

Έστω γράφος $G = \{V, E\}$ και το IS πρόβλημα με είσοδο (G, k). Έστω ένα ανεξάρτητο σύνολο κορυφών S και μια τυχαία ακμή $(u, v) \in E$. Αφού το S είναι ανεξάρτητο τότε τουλάχιστον μια κορυφή είτε u είτε v δεν θα ανήκει στο σύνολο S, διότι αν $u, v \in S$, τότε η ακμή $(u, v) \notin E$. Άρα $\forall (u, v) \in E$ υπάρχει τουλάχιστον μία κορυφή που ανήκει στο σύνολο $V \setminus S$ που σημαίνει πως το σύνολο αυτό είναι Vertex Cover, καθώς εξ ορισμού ένα σύνολο U είναι Vertex Cover όταν

$\forall \{u, v\} \in E: u \in U \vee v \in U.$

Στον γράφο μας έχουμε ανεξάρτητα σύνολα κορυφών μεγέθους τουλάχιστον κ. Επομένως

$$|S| \geq k \Leftrightarrow |V \backslash S| \leq |V| - k$$

Άρα για να λύσουμε το πρόβλημα $G = \{V, E\}\ \mu\varepsilon\ IS = (G, k)$ είναι το ίδιο με το λύσουμε το πρόβλημα $G = \{V, E\}\ \mu\varepsilon\ VC = (G, |V| - k)$, συμπέρασμα που επιβεβαιώνεται και από την παρατήρηση μας στο ερώτημα (β). Έτσι, καταλήγουμε πως το πρόβλημα IS είναι NP-πλήρες.

(δ) Γνωρίζουμε πως το IS πρόβλημα είναι NP-πλήρες. Θα ανάγουμε το πρόβλημα VC στο πρόβλημα IS αποδεικνύοντας έτσι πως το VC είναι NP-πλήρες.

Έστω γράφος $G = \{V, E\}$ και το VC πρόβλημα με είσοδο $(G, |V| - k)$. Έστω ένα κάλυμμα κορυφών C και ένα τυχαίο ζεύγος κορυφών $(u, v) \in V \backslash C$. Επειδή το C είναι κάλυμμα κορυφών, δηλαδή καλύπτει όλες τις ακμές του γράφου, δεν υπάρχει ακμή που να ενώνει τις κορυφές (u,v). Επομένως το σύνολο $V \backslash C$ είναι ανεξάρτητο σύνολο κορυφών.

Στον γράφο μας έχουμε καλύμματα κορυφών μεγέθους το πολύ $|V| - k$. Επομένως

$$|C| \leq |V| - k \Leftrightarrow |V \backslash C| \geq k$$

Άρα για να λύσουμε το πρόβλημα $G = \{V, E\}\ \mu\varepsilon\ VC = (G, |V| - k)$ είναι το ίδιο με το λύσουμε το πρόβλημα $G = \{V, E\}\ \mu\varepsilon\ IS = (G, k)$, συμπέρασμα που επιβεβαιώνεται και από την παρατήρηση μας στο ερώτημα (β). Έτσι, καταλήγουμε πως το πρόβλημα VC είναι NP-πλήρες.