Σχολή Ηλεκτρολόγων Μηχανικών και Μηχανικών Υπολογιστών

Θεμελιώδη Θέματα Επιστήμης Υπολογιστών 2016-2017
1η Σειρά Γραπτών Ασκήσεων

Όνομα: Παναγιώτης Κωστοπαναγιώτης
Εξάμηνο: 3ο
Α.Μ: 03115196

Άσκηση 1.

Θα θέλαμε να μπορούμε να υπολογίσουμε με κάποιο αποδοτικό τρόπο το άθροισμα των ψηφίων του αριθμού 2^29 ώστε έπειτα να μπορούσαμε να βρούμε το ψηφίο που λείπει (έστω α) αφαιρώντας το άθροισμα από το 45 = 0 + 1 + 2 + … + 9.

Έστω $2^{29} = \sum_{i=0}^{8} 10^i \alpha_i$ Έχουμε λοιπόν: α = 45 - $\sum_{i=0}^{8} \alpha_i$.Επιπλέον, από βασική ιδιότητα διαιρετότητας λαμβάνουμε ότι $(\sum_{i=0}^{8} 10^i \alpha_i) mod\, 9 = (\sum_{i=0}^{8} \alpha_i) mod\, 9$.Αντικαθιστώντας στην προηγούμενη σχέση και παίρνοντας το υπόλοιπο με το 9 κατά μέλη, λαμβάνουμε α mod 9 = 45 mod 9 - $(\sum_{i=0}^{8} 10^i \alpha_i) mod\, 9$ = 0 – $2^{29} mod\, 9$ = 9 – $2^{29} mod\, 9$. Όπου το $2^{29} mod\, 9$ υπολογίζεται με modular exponentiation by squaring σε λογαριθμική πολυπλοκότητα. Υπολογίζοντας λαμβάνουμε α = 9 – 5 = 4.

Άσκηση 2.

Χωρίζουμε τους κόμβους μας σε 2 ομάδες μεγέθους $\frac{n}{2}$ η καθεμία ($\lfloor \frac{n}{2} \rfloor$ και $\lfloor \frac{n}{2} \rfloor + 1$ αντίστοιχα αν ο n είναι περιττός). Έπειτα οι κόμβοι στις ομάδες ανταλλάζουν τα μυστικά τους, διαδοχικά, από αριστερά προς τα δεξιά. Έτσι καταλήγουμε με 2 κόμβους σε κάθε ομάδα που γνωρίζουν όλα τα μυστικά εντός της ομάδας μέ n – 2 ανταλλαγές μυστικών συνολικά. Ας ονομάσουμε τους εν λόγω κόμβους A_1 , B_1 και A_2 , B_2 αντίστοιχα.

Έπειτα γίνονται 2 ανταλλαγές μυστικών, του A_1 με τον A_2 και του B_1 με τον B_2 . Έτσι καταλήγουμε με 4 κόμβους που ξέρουν όλα τα μυστικά. Το μόνο που μένει είναι οι κόμβοι αυτοί να πούνε τα μυστικά στην ομάδα τους. Αυτό απαιτεί συνολικά n – 4 ανταλλαγές, δηλαδή την διαδοχική ανταλλαγή που είχαμε και στο αρχικό στάδιο, χωρίς 2 ακμές που είναι αυτές μεταξύ των A_1 , B_1 και A_2 , B_2 αντίστοιχα.

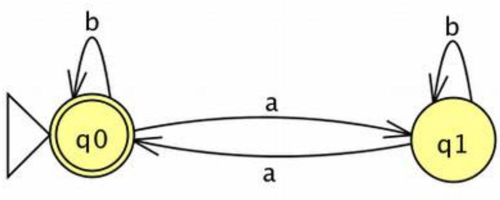
Συνολικά έχουμε n – 2 + 2 + n – 4 = 2n – 4 ανταλλαγές μυστικών που είναι και το βέλτιστο.

Άσκηση 3.

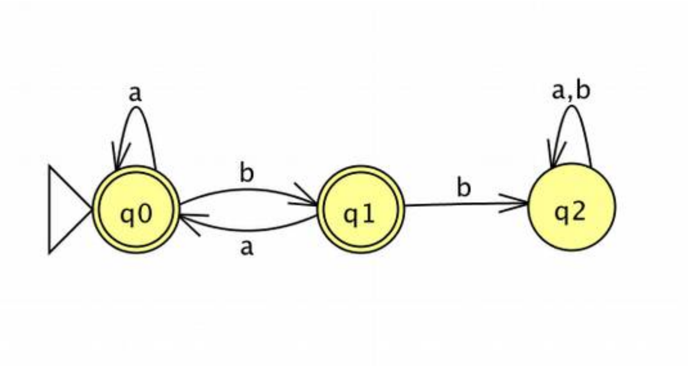
<u>Γλώσσα L1</u>

Κατασκευάζουμε 2 DFA για τα επιμέρους υποπροβλήματα και τα συνθέτουμε σε ένα NFA-ε για να πάρουμε την ένωση των υποπροβλημάτων που μας δίνει αυτόματο που αναγνωρίζει την αρχική μας γλώσσα.

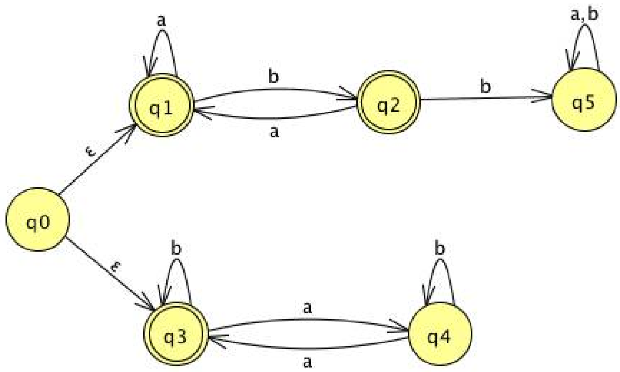
DFA για άρτιο αριθμό 'a':



DFA για συμβολοσειρές που δεν περιέχουν το "bb":

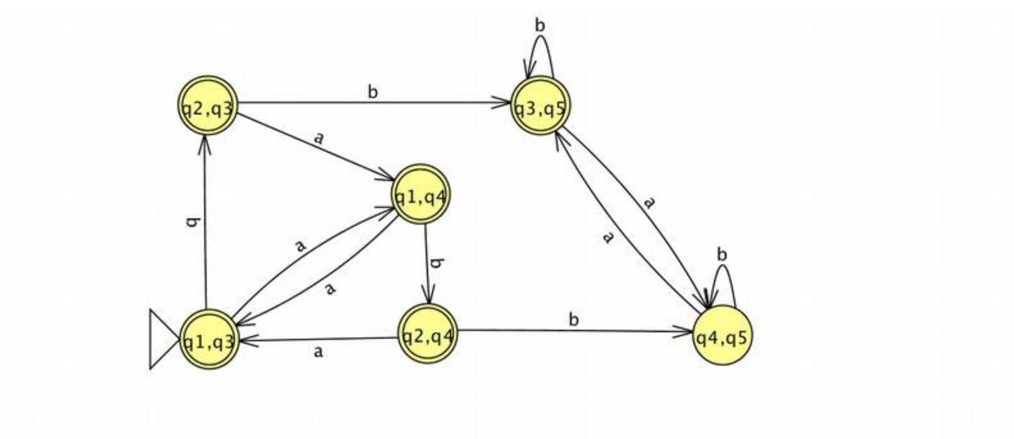


Συνδυάζοντας τα 2 αυτόματα λαμβάνουμε το ακόλουθο NFA-ε:



Κάνουμε την μετατροπή σε DFA:

State	a	b	ε
q0	-	-	{q1,q3}
{q1,q3}	{q1,q4}	{q2,q3}	-
{q1,q4}	{q1,q3}	{q2,q4}	-
{q2,q4}	{q1,q3}	{q4,q5}	-
{q4,q5}	{q3,q5}	{q4,q5}	-
{q3,q5}	{q4,q5}	{q3,q5}	-
{q2,q3}	{q1,q4}	{q3,q5}	-



Εφαρμόζουμε την μέθοδο της ελαχιστοποίησης για να δούμε αν το εν λόγω DFA είναι το ελάχιστο δυνατό:

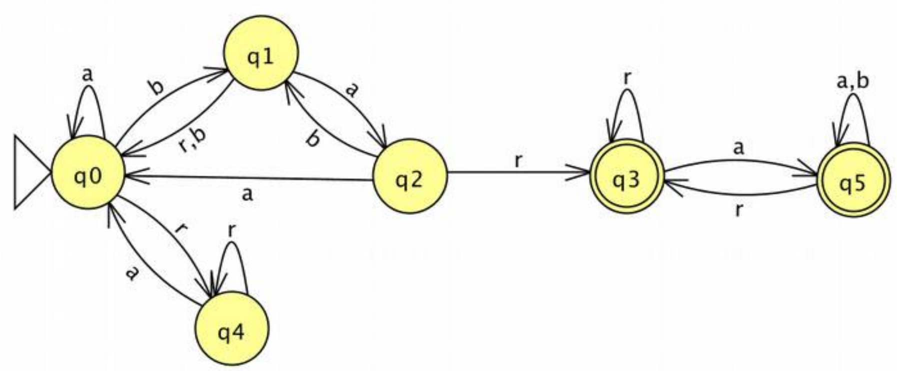
{q2,q3}	X2	-	-	-	-
{q1,q4}	X2	X2	-	-	-
{q2,q4}	X1	X1	X1	-	-
{q4,q5}	X0	X0	X0	X0	-
{q3,q5}	X1	X1	X1	X1	X0
	{q1,q3}	{q2,q3}	{q1,q4}	{q2,q4}	{q4,q5}

Όλες οι καταστάσεις του DFA είναι διακρίσιμες ανα 2, συνεπώς το DFA που κατασκευάσαμε είναι το ελάχιστο δυνατό που να αναγνωρίζει την γλώσσα L1.

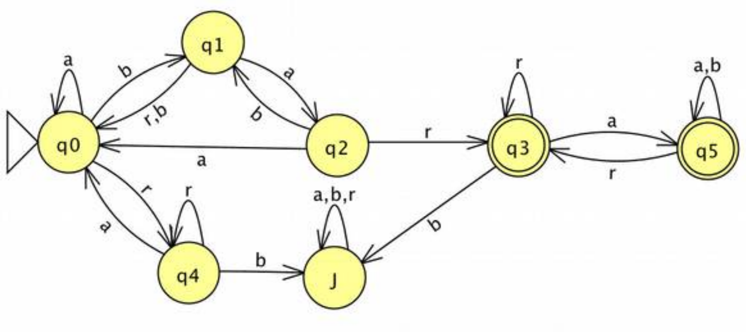
Μια κανονική έκφραση για την γλώσσα L1 είναι η: (aba + b(aa)*) *

Γλώσσα L2

Αρχικά κατασκευάζουμε NFA το οποίο αναγνωρίζει την γλώσσα L2.



Για την κατασκευή ισοδύναμου DFA, επειδή ο μη-ντετερμινισμός εντοπίζεται απλά στην παράλειψη κάποιων μεταβάσεων, δημιουργούμε ένα junk-state και προσθέτουμε τις εν λόγω μεταβάσεις.



Για να διαπιστώσουμε αν το DFA που κατασκευάσαμε είναι το ελάχιστο δυνατό, εφαρμόζουμε την μέθοδο της ελαχιστοποίησης.

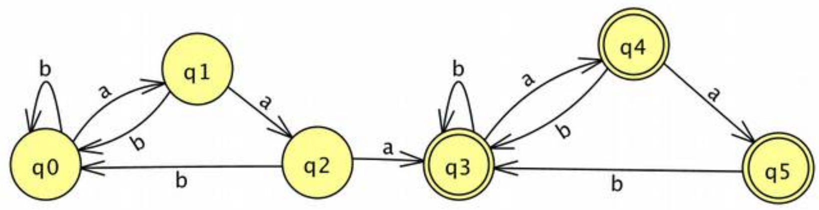
q1	X2	-	-	-	-	-
q2	X1	X1	-	-	-	-
q3	X0	X0	X0	-	-	-
q4	X3	X2	X1	X0	-	-
q5	X0	X0	X0	X1	X0	-
J	X3	X2	X1	X0	X4	X0
	q0	q1	q2	q3	q4	q5

Από τον πίνακα προκύπτει πως όλες οι καταστάσεις είναι ανά 2 διακρίσιμες. Συνεπώς το DFA που κατασκευάσαμε είναι το ελάχιστο που αναγνωρίζει την L2.

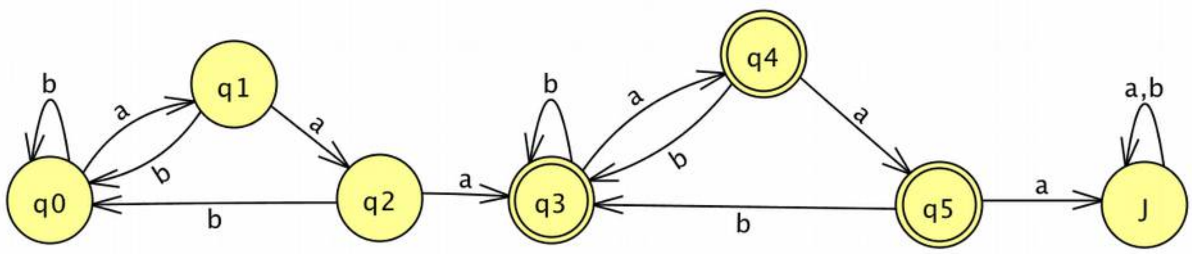
Κανονική έκφραση για την L2: ((a+b)*r*a)*bar((a+b)*r*a)*

Γλώσσα L3

Κατασκευάζουμε NFA που να αναγνωρίζει την γλώσσα L3.



Το ισοδύναμο DFA είναι το παρακάτω(προσθήκη junk-state στην κατάσταση q5).



Για να διαπιστώσουμε αν το εν λόγω DFA είναι το ελάχιστο δυνατό εφαρμόζουμε την μέθοδο της ελαχιστοποίησης.

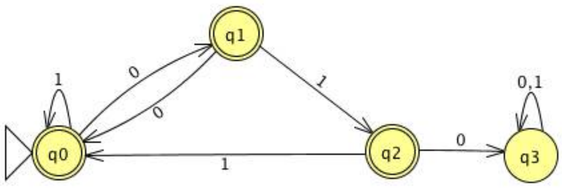
q1	X2	-	-	-	-	-
q2	X1	X1	-	-	-	-
q3	X0	X0	X0	-	-	-
q4	X0	X0	X0	X2	-	-
q5	X0	X0	X0	X1	X1	-
J	X3	X2	X1	X0	X0	X0
	q0	q1	q2	q3	q4	q5

Από τον παραπάνω πίνακα, συμπαιρένουμε πως όλες οι καταστάσεις είναι διακρίσιμες ανά 2 και άρα το εν λόγω DFA είναι το ελάχιστο δυνατό.

Μια κανονική έκφραση για την γλώσσα L3 είναι η: ((a + aa)b)*aaa((a+aa)b)*

Άσκηση 4.

α) Για την κατασκευή της εν λόγω κανονικής έκφρασης, αρχικά κατασκευάζουμε αυτόματο(DFA) που αναγνωρίζει την γλώσσα και έπειτα το μετατρέπουμε σε κανονική έκφραση.



Μετατρέποντας το σε κανονική έκφραση λαμβάνουμε ((1+00)*11)*.Η ορθότητα της κανονικής έκφρασης προκύπτει απευθείας από την ορθότητα του αυτομάτου.

β) Το σύνολο των strings με την δεδομένη ιδιότητα είναι όσα δεν περιέχουν πάνω από 2 συνεχόμενα μηδενικά, ή 2 συνεχόμενους άσσους, διότι αν περιείχαν, κάποιο σύμβολο θα ξεπέρναγε σε εμφανίσεις το άλλο κατά 2.

Για πλήρη απόδειξη του εν λόγω ισχυρισμού μπορούμε να εφαρμόσουμε την μέθοδο της μαθηματικής επαγωγής στο μέγεθος του string.

Συνεπώς η κανονική έκφραση που περιγράφει την δεδομένη γλώσσα είναι η (01 + 10)*.

Άσκηση 5.

(i) α) - <u>Κανονικότητα</u>:

Θα αποδείξουμε πως το εν λόγω σύνολο δεν είναι κανονικό. Χρησιμοποιούμε το pumping lemma. Έστω πως το σύνολο που μας δίνεται είναι κανονικό, τότε για κάθε αυτόματο μεγέθους $n, n \in Z$ που το αναγνωρίζει επιλέγουμε λέξη $z = 0^{n+1} 1^n$ με μήκος $L(z) = 2n+1$. Θεωρούμε σπάσιμο της εν λόγω λέξης z=uvw όπου $L(uv) \leq n$ Από το παραπάνω συμπεραίνουμε πως $v = 0^k$ για κάποιο k. Εφ' όσον η λέξη ανήκει στην γλώσσα τότε και κάθε λέξη της μορφής $z' = uv^i w \forall i$ θα ανήκει στην γλώσσα. Επιλέγοντας i=0 λαμβάνουμε z'=uw. Όμως, η λέξη που παράχθηκε περιέχει το πολύ n μηδενικά και ακριβώς n άσσους, συνεπώς δεν ανήκει στην γλώσσα το οποίο είναι άτοπο. Άρα το σύνολο δεν είναι κανονικό.

- <u>Γραμματική</u>

G_1 : V={S}, T={0,1}, S→0S | 00S1 | ε

- <u>Αυτόματο</u>

Η γραμματική που περιγράφει την γλώσσα μας είναι context-free, συνεπώς σχεδιάζουμε PDA.

push(a): $\forall 0 \in input$
pop: $\forall 1 \in input$
Αποδοχή με μη κενή στοίβα

β) - <u>Κανονικότητα</u>

Κάνουμε χρήση του pumping lemma. Έστω πως το σύνολο είναι κανονικό και έστω αυτόματο μεγέθους n in Z που το αναγνωρίζει. Κατασκευάζουμε λέξη z της μορφής $z = 0^{2n} 1^n 0^n$ με $L(z) = 4n$. Προφανώς η παραπάνω λέξη ανήκει στην γλώσσα. Θεωρούμε τυχαίο σπάσιμο της $z = uvw$ με $L(uv) \leq n$, τότε θα είναι $v = 0^k$ για κάποιο $k \in Z$.Εφ' όσον η εν λόγω λέξη ανήκει στην γλώσσα τότε κάθε λέξη της μορφής $z' = uv^i w$ θα ανήκει στην γλώσσα, για κάποιο $i > 0$.Επιλέγουμε i=2, τότε, η λέξη θα είναι της μορφής $z' = 0^{2n+L(v)} 1^n 0^n$, η οποία όμως δεν ανήκει στην γλώσσα, άτοπο. Άρα το σύνολό μας δεν είναι κανονικό.

- <u>Γραμματική</u> G_2 : V = {S,A}, T={0,1}, S→00A0 | 00S0, A→1A | 1

- <u>Αυτόματο</u>

Η γραμματική που περιγράφει την γλώσσα μας είναι context-free και άρα σχεδιάζουμε PDA που να την αναγνωρίζει.

push(a): $\forall 0 \in input$
συνέχισε μέχρι να διαβαστεί 1
pop, pop: $\forall 0 \in input$
αποδοχή με κενή στοίβα

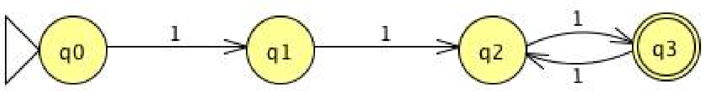
γ) – Κανονικότητα

Το σύνολο αυτό είναι κανονικό καθότι μπορούμε να φτιάξουμε μια κανονική έκφραση που να το περιγράφει. Η εν λόγω έκφραση είναι η 111(11)*

- Γραμματική

G_3 : V={S}, T={1}, S→11S | 111

- Αυτόματο



δ) - Κανονικότητα

Για την απόδειξη της μη-κανονικότητας του συνόλου χρησιμοποιούμε ξανά το pumping lemma. Έστω πως το σύνολο είναι κανονικό και αναγνωρίζεται από αυτόματο μεγέθους n. Κατασκευάζουμε την λέξη $z=0^{n+1}1^n0^n$, η οποία ανήκει στην γλώσσα. Θεωρούμε τυχαίο σπάσιμο της λέξης z=uvw, τέτοιο ώστε $L(uv)\leq n$. Τότε θα είναι $v=0^k$ για κάποιο $k>0$. Εφ' όσον η παραπάνω λέξη ανήκει στην γλώσσα, τότε και κάθε λέξη της μορφής $z'=uv^iw$,θα ανήκει στην γλώσσα, όμως για i=0 λαμβάνουμε $z'=uw=0^{n+1-k}1^n0^n$ η οποία δεν ανήκει στην γλώσσα, άτοπο. Άρα το σύνολό μας δεν είναι κανονικό.

- Γραμματική

G_4 : C={S, A, B, P}, T={0,1}, S → 1P, P → 1A1, A → 0 | 1A0B, B0 → 0B, B1 → 11

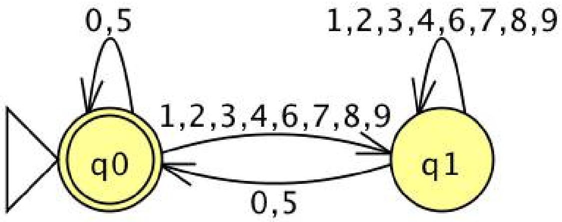
- Αυτόματο

Η γραμματική που περιγράφει την γλώσσα μας είναι context-sensitive, συνεπώς σχεδιάζουμε LBA που να την αναγνωρίζει. Προσωμοιάζουμε την λειτουργία του LBA με ένα 2-PDA.

Όσο διαβάζεις 0 κάνε push 0 στην 1η στοίβα
Όσο διαβάζεις 1 κάνε push 1 στην 2η στοίβα.
Όσο διαβάζεις 0 κάνε pop από την πρώτη στοίβα ένα 0 και από την δεύτερη ένα 1. Αν κάποια στιγμή κάποια από τις στοίβες είναι κενή, απέρριψε.
Αφού διαβαστεί η είσοδος, αν η πρώτη στοίβα είναι κενή απέρριψε.
Κάνε pop ένα 0 από την πρώτη στοίβα.
Αποδοχή αν και οι 2 στοίβες είναι κενές.

Άσκηση 6.

(i) Κατακευάζουμε αυτόματο με 2 καταστάσεις. Το αυτόματο μεταβαίνει σε κατάσταση αποδοχής μόνο αν δεχτεί 5 ή 0 σαν ψηφίο(αφού η διαιρετότητα με το 5 καθορίζεται από το τελευταίο ψηφίο). Το αυτόματο φαίνεται παρακάτω:



Άσκηση 8.

(α) Έστω αυτόματο A, το οποίο αναγνωρίζει την γλώσσα L, το αυτόματο αυτό αποτελείται από μία αρχική κατάσταση q_0 και ένα σύνολο καταστάσεων αποδοχής
F = { $q_{a_1}, q_{a_2}, \ldots, q_{a_n}$ }.

Λήμμα 1: Κάθε πεπερασμένο αυτόματο με σύνολο καταστάσεων F, μπορεί να μετασχηματιστεί σε ένα ισοδύναμο NFA-ε με μία κατάσταση αποδοχής

Απόδειξη: Κάνουμε την ακόλουθη κατασκευή, για κάθε κατάσταση $q \in F$, βάζουμε μια αμφίδρομη ε-ακμή σε μια καινούργια κατάσταση αποδοχής Q. Πλέον, η μόνη κατάσταση αποδοχής είναι η Q και το αυτόματο που παράχθηκε είναι ισοδύναμο με αυτό.

Κάνοντας χρήση του λήμματος 1 μετασχηματίζουμε το αρχικό μας αυτόματο. Έτσι, καταλήγουμε σε ένα NFA-ε το οποίο αποτελείται από 1 κατάσταση αποδοχής και μια αρχική κατάσταση που αναγνωρίζει την γλώσσα L.

Έστω Α το παραπάνω αυτόματο και A^T το ίδιο αυτόματο αλλά με ανεστραμμένες τις ακμές μετάβασης και του οποίο η αρχική κατάσταση είναι η τελική κατάσταση του Α. Τότε, το αυτόματο A^T αναγνωρίζει την γλώσσα L^R .

Τυπικά, για ένα αυτόματο Α, συμβολίζουμε ως δ_A την συνάρτηση μετάβασής του, ως Q^A το σύνολο καταστάσεών του, καθώς και ως Q^A_{AC} την κατάσταση αποδοχής του.

Θα λέμε επίσης ότι ένα μονοπάτι $q_1^A, q_2^A, \ldots, q_n^A$ ενός αυτομάτου Α είναι έγκυρο για μια λέξη w μεγέθους n αν ξεκινάει από την αρχική κατάσταση, δηλαδή $q_s = q_1$, καταλήγει στην τελική, δηλαδή $q_n = Q^A_{AC}$, και $\delta_A(q_i, w_{i-1}) = q_{i+1} \forall i \leq n$

Απόδειξη: Έστω πως το αυτόματο A^T δεν αναγνωρίζει την γλώσσα L^R , δηλαδή υπάρχει λέξη w με μήκος l για την οποία ισχύει $w \in L \wedge w^R \notin L^R$. Δηλαδή η λέξη w αναγνωρίζεται από το αυτόματο A, ενώ η w^R δεν αναγνωρίζεται από το A^T . Αυτό συνεπάγεται ότι για την λέξη w υπάρχει μονοπάτι καταστάσεων $q_0^A, q_1^A, \ldots, q_{l-1}^A, Q_{AC}^A$ τέτοιο ώστε $\delta_A(q_i^A, w_i) = q_{i+1}^A \forall i \leq l-2 \wedge \delta_A(q_{l-1}^A, w_{l-1}) = Q_{AC}^A$ αλλά δεν υπάρχει μονοπάτι καταστάσεων $q_0^{A^T}, q_1^{A^T}, \ldots, q_{l-1}^{A^T}, Q_{AC}^{A^T}$ στο αυτόματο A^T με $\delta_{A^T}(q_i^{A^T}, w_i^R) = q_{i+1}^{A^T} \forall i \leq 2 \wedge \delta_A(q_{l-1}^{A^T}, w_{l-1}^R) = Q_{AC}^{A^T}$.Όμως, το μονοπάτι $Q_{AC}, q_{l-1}, \ldots, q_{1,} q_0$ είναι έγκυρο μονοπάτι του αυτομάτου A^T για την λέξη w^R . Συνεπώς η λέξη w^R αναγνωρίζεται από το αυτόματο A^T εφ' όσον η w αναγνωρίζεται από το αυτόματο A. Άτοπο.

Συνεπώς, για κάθε αυτόματο A το οποίο αποδέχεται μια κανονική γλώσσα L, μπορούμε να φτιάξουμε αυτόματο A' το οποίο αποδέχεται την γλώσσα L^R . Συνεπώς αποδυκνείεται η κλειστότητα των κανονικών γλωσσών ως προς την πράξη της αναστροφής.

β) <u>Κλειστότητα context-free γλωσσών ως προς την πράξη ένωση:</u>

Έστω G_1 και G_2 οι γραμματικές των αντίστοιχων γλωσσών $L_{1,} L_2$. Οι γραμματικές αυτές περιέχουν ένα σύνολο από τερματικά, μη-τερματικά σύμβολα $T_{1,} T_{2,}$ V_1, V_2 αντίστοιχα, καθώς και ένα σύνολο από κανόνες R_1 και R_2 αντίστοιχα.

Έχουμε για την $G_1 \cup G_2$: $T = T_1 \cup T_2$, $V = V_1 \cup V_2 \cup (S \notin V_{1,} V_2)$ και $R = R_1 \cup R_2 \cup (S \rightarrow V_1 \vee V_2)$. Εφ΄όσον κάθε κανόνας των 2 γλωσσών είναι χωρίς συμφραζόμενα, τότε το σύνολο της ένωσης τους δεν θα περιέχει κανόνες με συμφραζόμενα. Συνεπώς γραμματική που παράγεται, συνεπώς και η γλώσσα της ένωσης, θα είναι context-free.

<u>Κλειστότητα context-free γλωσσών ως προς την πράξη άστρο του Kleene:</u>

Έστω G η context-free γραμματική που παράγει την γλώσσα L με σύνολο τερματικών T και σύνολο μη-τερματικών V. Θεωρούμε ένα νέο μη-τερματικό σύμβολο $S \notin V$ και ορίζουμε νέο κανόνα της μορφής S → SV | ε. Αυτή η νέα γραμματική μας παράγει την L* και είναι επίσης context-free.

<u>Κλειστότητα context-free γλωσσών ως προς την αντιστροφή:</u>

Έστω G η context-free γραμματική που παράγει την γλώσσα L. Αν στο σύνολο κανόνων της G αντιστρέψουμε το δεξιό μέλος κάθε κανόνα, τότε παράγεται μια νέα γραμματική G', η οποία είναι σαφώς context-free, που παράγει την γλώσσα L^R .Συνεπώς όλες οι context-free γλώσσες είναι κλειστές ως προς την αντιστροφή.

<u>Οι context-free γλώσσες δεν είναι κλειστές ως προς το συμπλήρωμα:</u>

Θεωρούμε ως παράδειγμα μια κανονική γλωσσά L, για ευκολία μπορούμε να θεωρήσουμε και πως η L είναι πεπερασμένη. Εφ' όσον η L είναι κανονική είναι προφανώς και context-free. Προφανώς η L^C θα είναι γλώσσα η οποία περιέχει λέξεις context-sensitive γλωσσών, οπότε είναι τουλάχιστον context-sensitive. Συνεπώς δεν μπορεί να είναι context-free.