

Lab 2

🕒 Created	@April 22, 2023 12:53 AM
📖 Class	Υπολογιστική Γλωσσολογία
📄 Type	Report
📎 Materials	
☑ Reviewed	<input type="checkbox"/>
≡ AM	1072754
≡ Department	Ηλεκτρολογων Μηχανικων και Τεχνολογιας Υπολογιστων
≡ Τομεας	Υπολογιστων
# Ετος Σπουδων	4
≡ Ονοματεπωνυμο	Νικόλας Φιλίππátος

[Question 1](#)

[Question 2](#)

[Question 3](#)

[Question 4](#)

[Question 5](#)

[Question 6](#)

[Question 7](#)

[Question 8](#)

Question 1

Σχεδιάστε στο JFLAP ένα αυτόματο πεπερασμένων καταστάσεων που θα αναγνωρίζει τη γλώσσα: **αλφαριθμητικά από το αλφάβητο $\Sigma=\{a,b,c\}$ που τελειώνουν με c.**

Θετικά παραδείγματα: c, bcc, abcabc, cccc, ababccbac
Αρνητικά παραδείγματα: abcba, cacab, cccccca, ababab

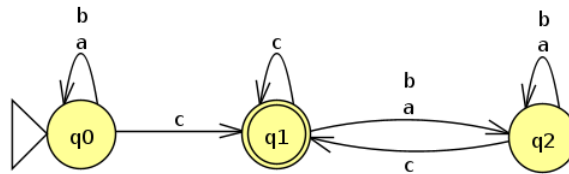
Ελέγξτε το αυτόματο που σχεδιάσατε με τα παραπάνω θετικά και αρνητικά παραδείγματα με Multiple Run και δείξτε screenshot με το αυτόματο και το test.

question 1

Απάντηση :

Ορίζουμε την αρχική κατάσταση (q_0) και την τελική κατάσταση (q_1) .

Με το transition Creator (shortcut T) πατώντας πάνω σε μια κατάσταση θα βγαλει ένα πλαίσιο κειμένου στο οποίο συμπληρώνουμε το γράμμα που θέλουμε . Αυτό θα εμφανισει ένα βέλος το οποίο ξεκινάει και τελειώνει πάνω στην κατάσταση που πατήσαμε. Αν θέλουμε να αρχίζει και να καταληγει σε άλλη κατάσταση , επιλεγουμε την κατάσταση και σερνουμε τον κερσορα μέχρι την κατάσταση που επιθυμούμε. Εάν αφήσουμε κενό το πλαίσιο κειμένου θα εμφανισει το γράμμα λ.



question01.jff

Για να κάνουμε multiple run, πηγαίνουμε στο input και στην λειτουργία multiple run .

Μπορούμε να εισαγάγουμε το input που θέλουμε είτε χειροκίνητα στο πλαίσιο του πίνακα που μας δίνει, είτε με ένα αρχείο και τα input μας χωρισμένα με ένα κενό , και καλούμε το αρχείο από το "Load Inputs" .

Input	Result
c	Accept
bbc	Accept
abcabc	Accept
cccc	Accept
ababccbac	Accept
abcba	Reject
cacab	Reject
cccccca	Reject
ababab	Reject

multiple run

https://s3-us-west-2.amazonaws.com/secure.notion-static.com/1005ccca-676e-4c8d-bc23-7f8f09127cf4/question_01.jff

https://s3-us-west-2.amazonaws.com/secure.notion-static.com/c3253917-976b-4658-8767-b498775c6c26/question_01_input.txt

Question 2

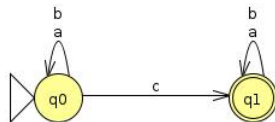
Σχεδιάστε στο JFLAP ένα αυτόματο πεπερασμένων καταστάσεων που θα αναγνωρίζει τη γλώσσα: αλφαριθμητικά από το αλφάβητο $\Sigma=\{a,b,c\}$ με ακριβώς ένα c σε οποιαδήποτε θέση.

Θετικά παραδείγματα: c, abc, cbbb, babcbba, abcba

Αρνητικά παραδείγματα: cacab, ababab, bcc, abcabca, ababcccba

question 2

Απάντηση:



question_02.jff

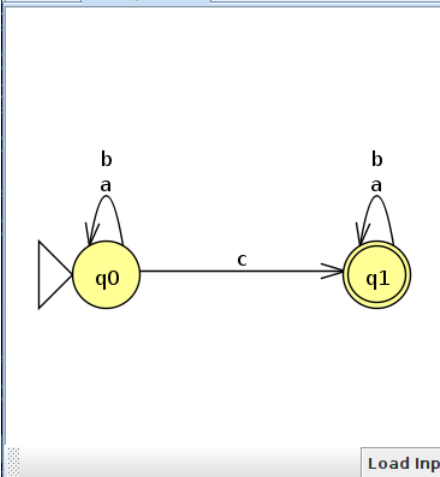


Table Text Size

Input	Result
c	Accept
abc	Accept
cbbb	Accept
babcbba	Accept
abcba	Accept
cacab	Reject
ababab	Reject
bcc	Reject
abcabca	Reject
ababcccba	Reject

Load Inputs Run Inputs Clear Enter Lambda View Trace

question_02.jff - multiple runs

https://s3-us-west-2.amazonaws.com/secure.notion-static.com/7eda7057-d7ab-481a-bd66-9f56d56f4d46/question_02.jff

https://s3-us-west-2.amazonaws.com/secure.notion-static.com/45516113-d8ba-486d-b9d8-7a88f20518e3/question_02_input.txt

Question 3

Σχεδιάστε στο JFLAP ένα αυτόματο πεπερασμένων καταστάσεων που θα αναγνωρίζει τη γλώσσα: **αλφαριθμητικά από το αλφάβητο $\Sigma=\{a,b,c\}$ με το πολύ δύο συνεχόμενα c, το πολύ μία φορά, σε οποιαδήποτε θέση.**

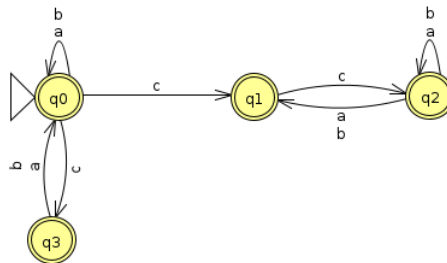
Θετικά παραδείγματα: c, abaa, cacab, babccbba, caacbcbac, cc

Αρνητικά παραδείγματα: abccabccab, bcccca, abcabccca, ccc

Ελέγξτε το αυτόματο που σχεδιάσατε με τα παραπάνω θετικά και αρνητικά παραδείγματα με Multiple Run και δείξτε screenshot με το αυτόματο και το test.

question 3

Απάντηση:



quesion03.jff

Input	Result
c	Accept
abaa	Accept
cacab	Accept
babccbba	Accept
caacbcbac	Accept
cc	Accept
abccabccab	Reject
bcccca	Reject
abcabccca	Reject
ccc	Reject

question 3 multiple run

https://s3-us-west-2.amazonaws.com/secure.notion-static.com/aac690b7-2aaa-4b7f-a401-7c1666f4aa74/question_03.jff

Question 4

Σχεδιάστε στο JFLAP το αυτόματο που αντιστοιχεί στον ακόλουθο πίνακα:

	a	b	c	λ
>q0	q0, q2			q1, q5
q1		q1	q0, q2	
q2		q3	q4	q5
q3	q3		q3	
q4:	q4	q5		
q5:	q4	q5		

και δείξτε το screenshot.

ΣΗΜΕΙΩΣΗ: Το λ δεν είναι σύμβολο του Σ. Δηλώνει λ-transition (ή ε-transition) δηλαδή μετάβαση χωρίς σύμβολο.

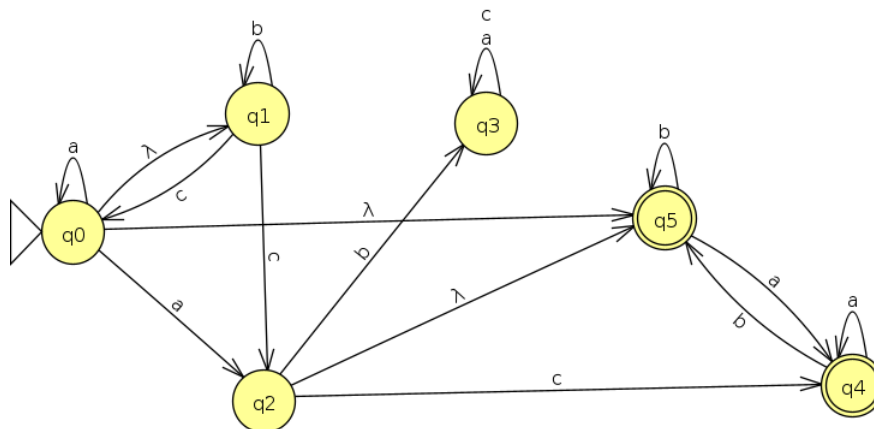
question 4

Απάντηση:

Το > σηματοδοτεί ότι είναι η αρχική κατάσταση και το : ότι είναι μια από τις τελικές καταστάσεις.

Για το λ, στο Transition Creator θα αφήσουμε κενό το πλαίσιο.

Κοιτάμε την κατάσταση στην γραμμή και την συνδεουμε με την κατάσταση που αντιστοιχεί στην ίδια σειρά κάτω από το γράμμα που κοιτάζουμε.



question04.jff

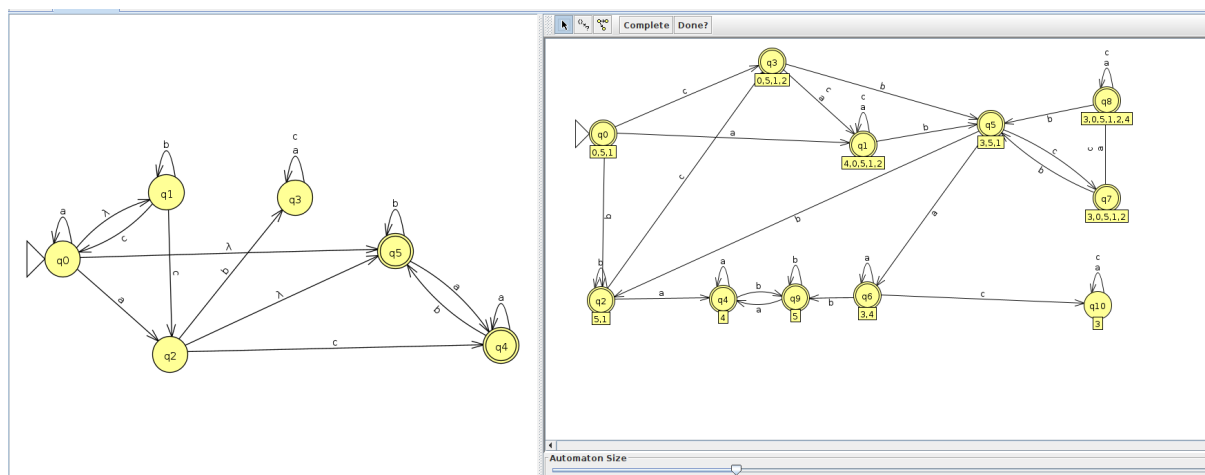
Question 5

Μετατρέψτε το αυτόματο του προηγούμενου ερωτήματος σε DFA και δείξτε το αποτέλεσμα (screenshot).

Απάντηση:

Απο το Convert > Convert to DFA .

Επιλεγουμε την τριτη επιλογη State Expander και διαλεγουμε την κατασταση που θελουμε να επεκτεινουμε. Συνεχιζουμε την διαδικασια για ολες τις καταστασεις που εμφανιζονται και εχουμε το παρακατω αποτελεσμα .



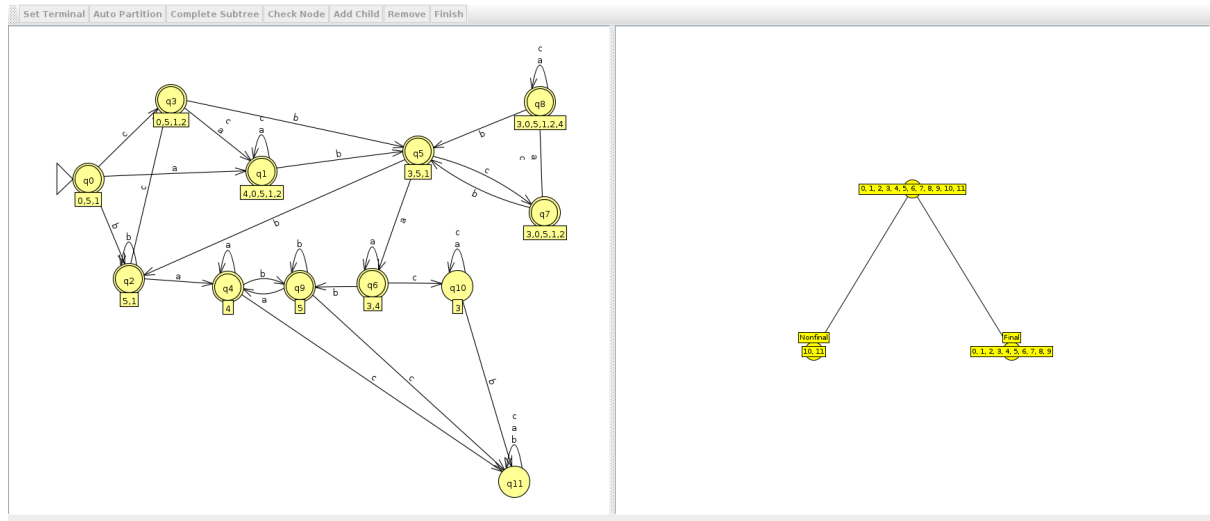
Question 6

Μετατρέψτε το αυτόματο του προηγούμενου ερωτήματος σε minimal DFA και δείξτε το αποτέλεσμα (screenshot).

Απάντηση:

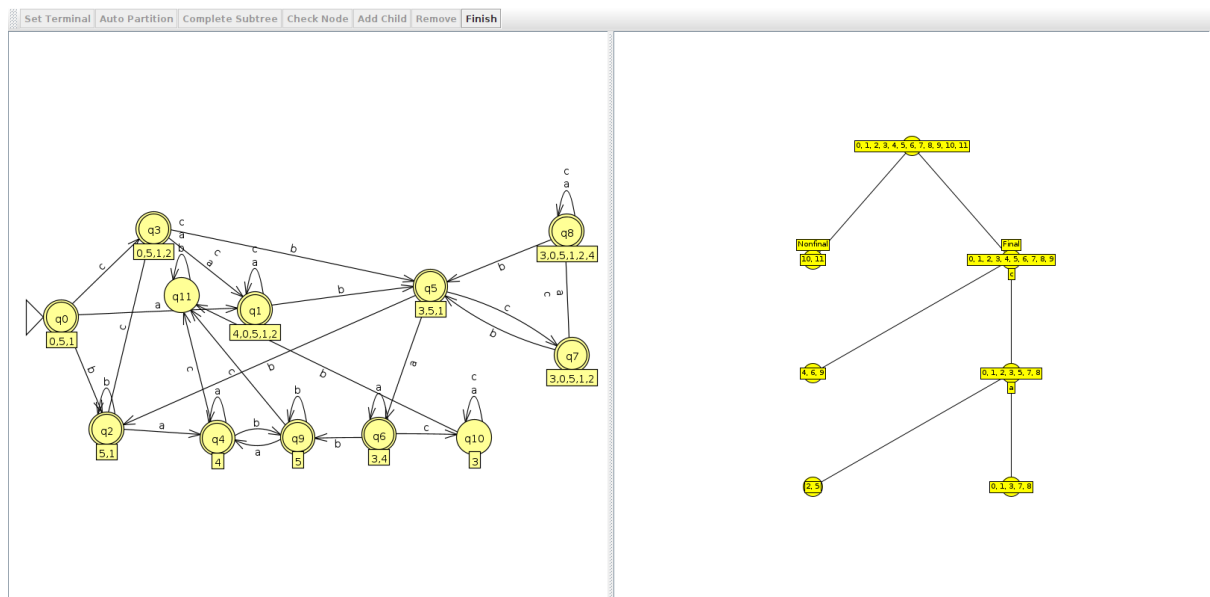
Convert > Minimize DFA

Επιλεγουμε την ριζα .



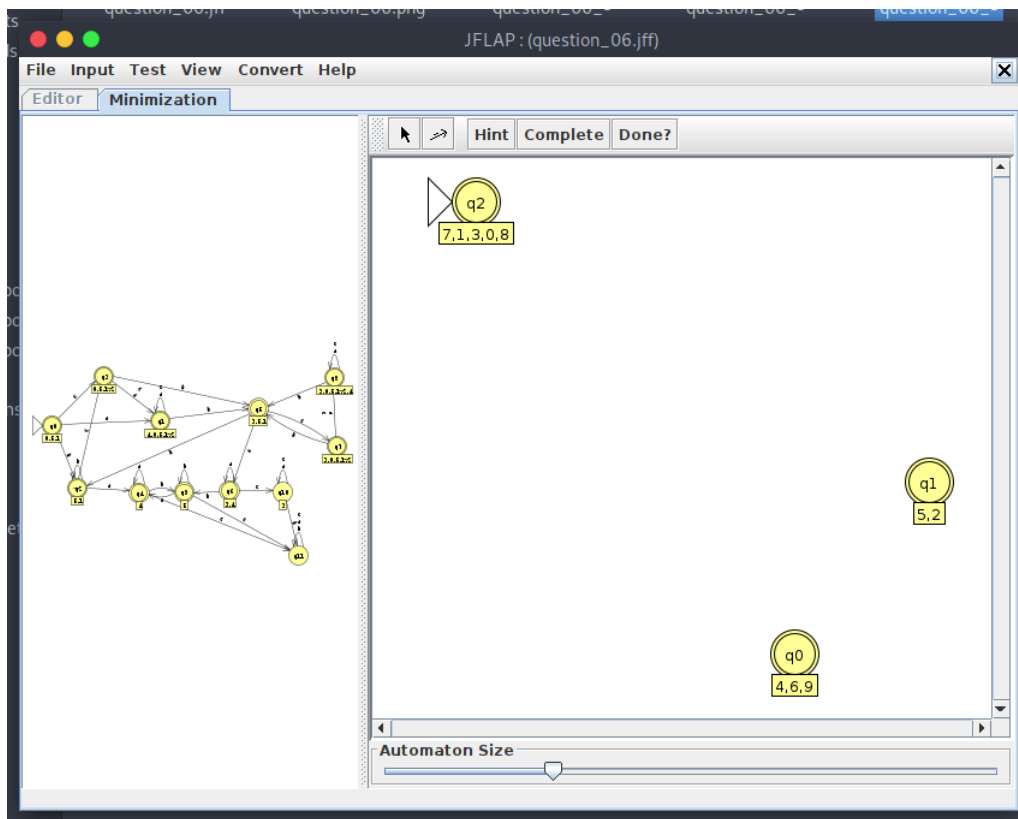
first step

Αφου επιλεξουμε την ριζα , μας ανοιγει η επιλογη Complete Subtree .



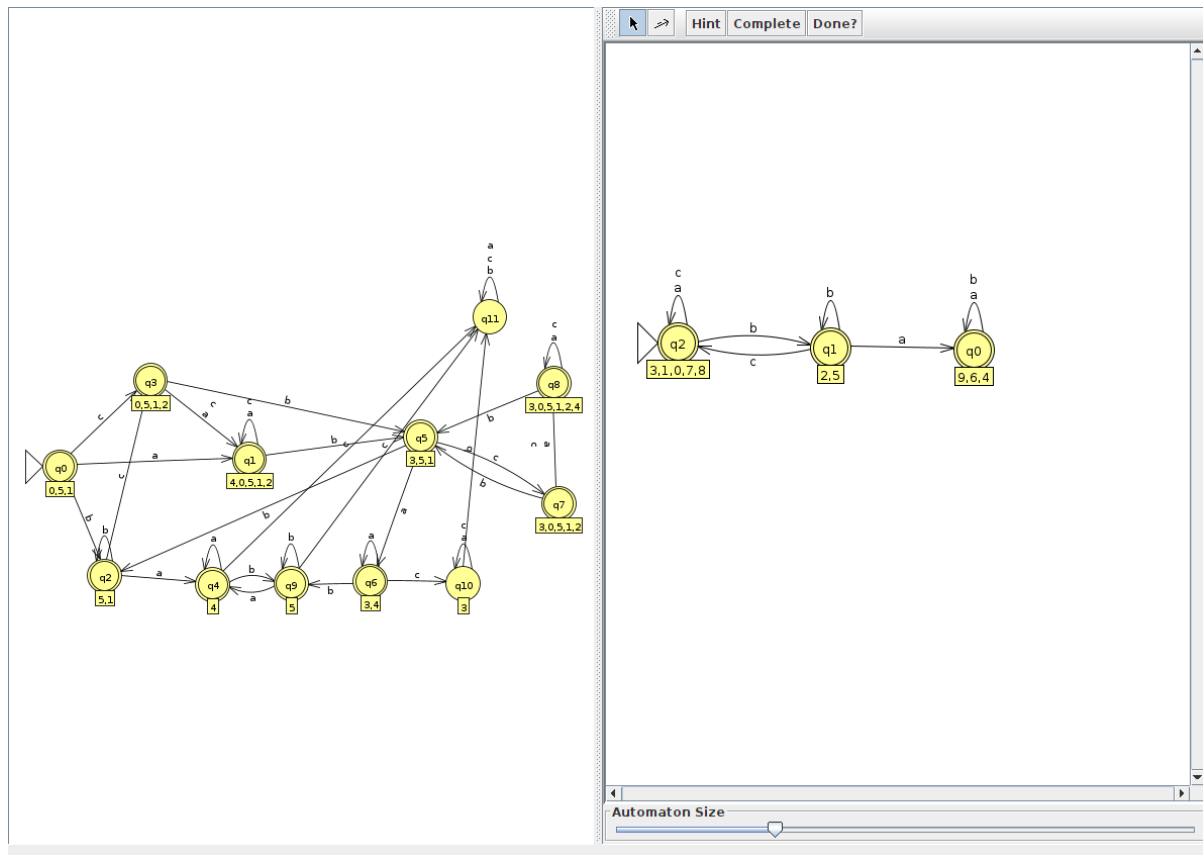
second step - subtrees

Προχωραμε στην επιλογη Finish

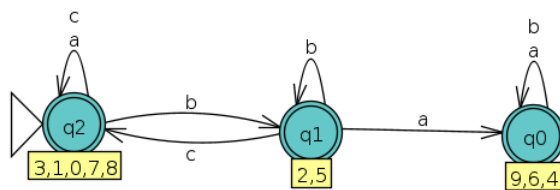


finish

Και μετα με την επιλογη Complete εχουμε το τελικο αποτελεσμα .



Minimized DFA :



Question 7

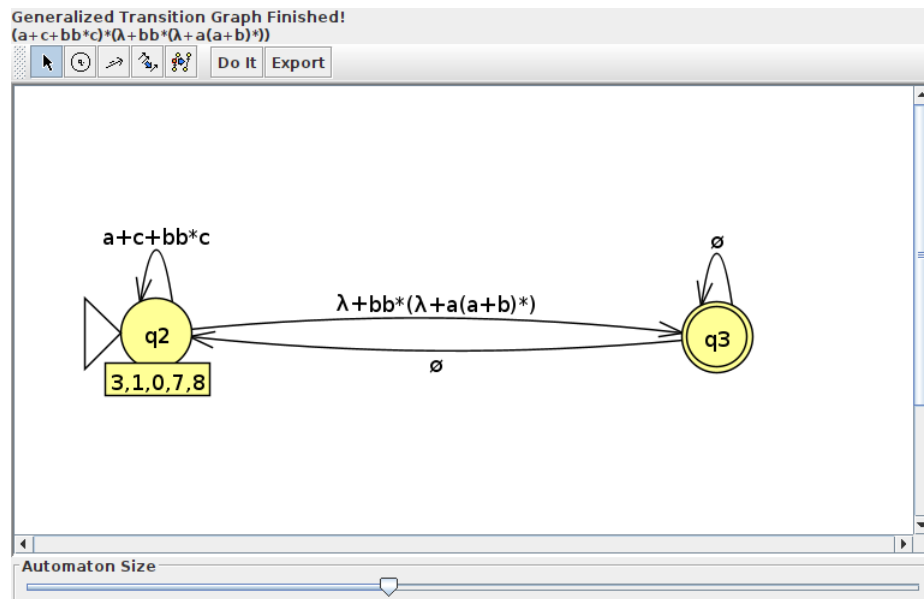
Μετατρέψτε το minimal DFA του προηγούμενου ερωτήματος σε regular expression και δείξτε το αποτέλεσμα (screenshot).

Απάντηση:

Convert > Convert FA to RE

Make a new state as the prompt message says.

And then spamming the do it button solves it until the message "You're Done . Go away" Pops up .



Regex :

$(a+c+bb^*c)^*(\lambda+bb^*(\lambda+a(a+b)^*))$

Question 8

Μετατρέψτε το minimal DFA του ερωτήματος 6 σε κανονική γραμματική και δείξτε το αποτέλεσμα (screenshot).

Απάντηση:

Convert > Convert Grammar

Παιρνουμε το minimized DFA . Και παταμε Show ALL .

