Σειρά Προβλημάτων 2 – Λύσεις

Άσκηση 1

Να δώσετε κανονικές εκφράσεις που να περιγράφουν τις πιο κάτω γλώσσες.

(α) Όλες οι λέξεις επί του αλφάβητου {0,1} οι οποίες περιέχουν το σύμβολο 0 τουλάχιστον δύο και όχι περισσότερες από τέσσερις φορές.

(β) Όλες οι λέξεις w επί του αλφάβητου {0,1} οι οποίες έχουν μήκος |w|, τέτοιο ώστε |w|mod 5 = 3.

(γ) Όλες οι λέξεις επί του αλφάβητου {0,1} στις οποίες κάθε εμφάνιση της συμβολοσειράς 00 ακολουθείται συνεχόμενα από τη συμβολοσειρά 11.

[Παράδειγμα: Οι λέξεις 010, 0100111 ανήκουν στη γλώσσα, ενώ οι λέξεις 00011 και 001100 δεν ανήκουν στη γλώσσα.]

(δ) Όλες οι λέξεις επί του αλφάβητου {0,1} στις οποίες δεν εμφανίζεται η συμβολοσειρά 010.

Λύση

(α) $1^*01^*01^*(\varepsilon \cup 01^* \cup 01^*01^*)$

(β) $[(0 \cup 1)(0 \cup 1)(0 \cup 1)(0 \cup 1)(0 \cup 1)]^*(0 \cup 1)(0 \cup 1)(0 \cup 1)$

(γ) $(1\cup 0(1\cup 011))^*(\varepsilon\cup 0)$

Εξήγηση: Κάθε εμφάνιση του 0 πρέπει να ακολουθείται είτε από 1, είτε από 011. Επομένως η λέξη μπορεί να φτιαχτεί μέσω επαναλήψεων των δομικών στοιχείων 1, 01, και 0011. Επιπρόσθετα η λέξη μπορεί να τελειώνει σε 0.

(δ) $(1^*\cup\varepsilon)(0\cup 11^+)^*(1^*\cup\varepsilon)$

Εξήγηση: Οι λέξεις μπορούν είτε να ξεκινούν είτε να τελειώνουν με αυθαίρετο αριθμό από 1. Ενδιάμεσα οι λέξεις μπορούν να κτιστούν με δομικά στοιχεία τα 0 και 11^+ (δηλαδή ακολουθίες που περιέχουν τουλάχιστον δύο συνεχόμενα 1).

Άσκηση 2

Θεωρήστε τις γλώσσες

$\Lambda_1 = \{ w \in \{a,b\}^* \mid$ *η w περιέχει τη λέξη aa το πολύ μια φορά* $\}$

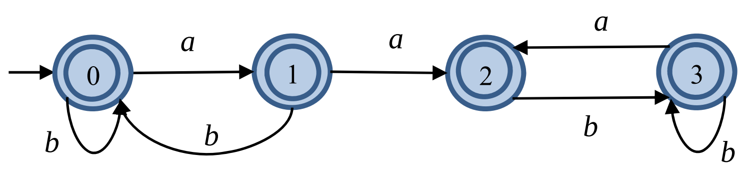
$\Lambda_2 = \{ w \in \{a,b\}^* \mid$ *η w περιέχει τη λέξη bb ακριβώς μία φορά* $\}$

Να υπολογίσετε την κανονική έκφραση που περιγράφει την τομή των δύο γλωσσών ακολουθώντας τα πιο κάτω βήματα.

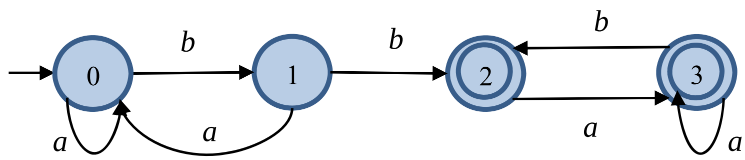
(i) Να περιγράψετε τις γλώσσες Λ_1 και Λ_2 ως NFA N_1 και N_2, αντίστοιχα.

Λύση:

Το πιο κάτω αυτόματο NFA αναγνωρίζει τη γλώσσα Λ_1.

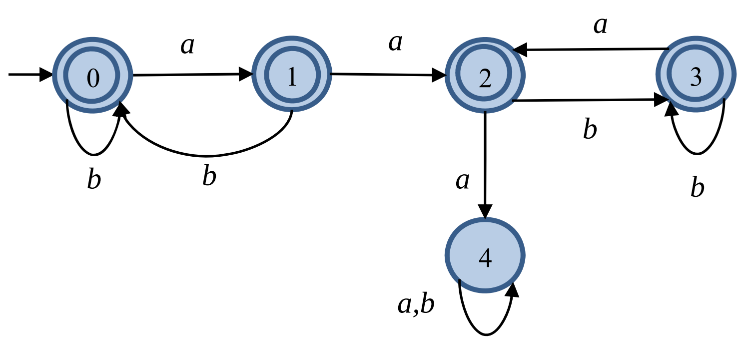


Το πιο κάτω αυτόματο NFA αναγνωρίζει τη γλώσσα $Λ_2$.

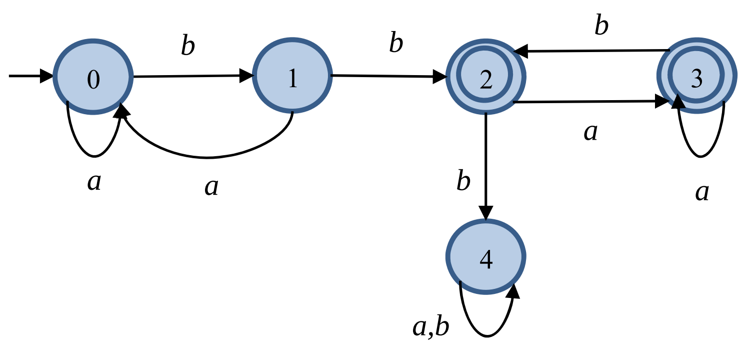


(ii) Να μετατρέψετε τα αυτόματα N_1 και N_2 σε ισοδύναμα DFA D_1 και D_2, αντίστοιχα, χρησιμοποιώντας τον σχετικό αλγόριθμο (Διαφάνεια 2-37 και 2-38).

Ο αλγόριθμος μετατροπής δημιουργεί τα πιο κάτω αυτόματα:



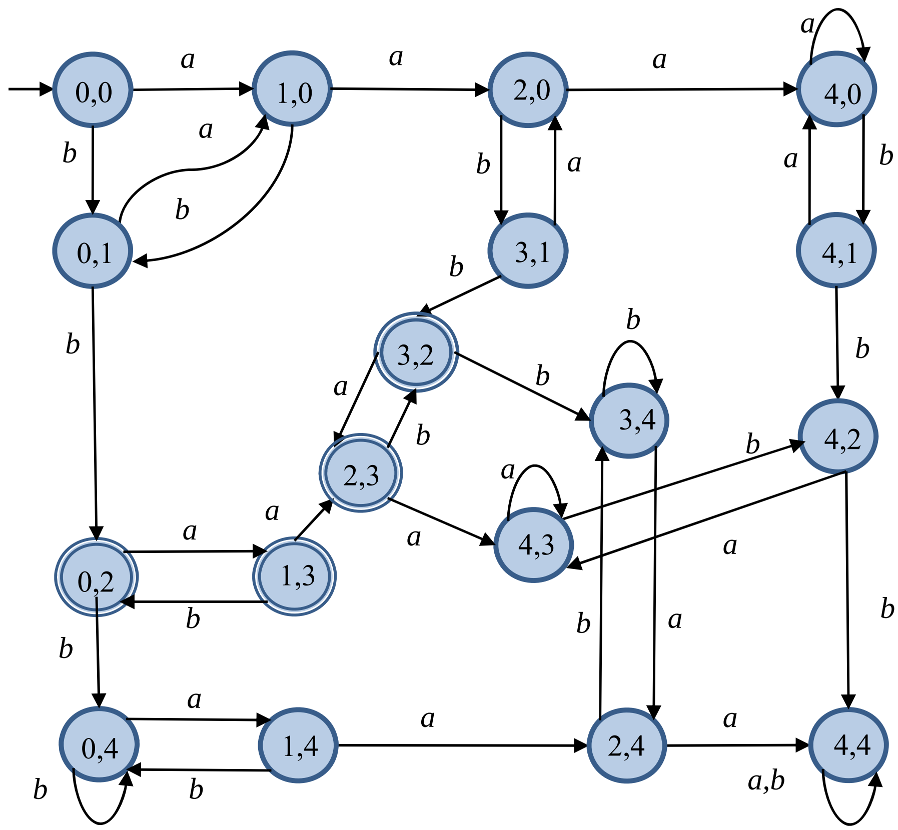
$\boldsymbol{D_1}$



$\boldsymbol{D_2}$

(iii) Να δημιουργήσετε το αυτόματο D που αναγνωρίζει την τομή των γλωσσών των αυτομάτων D_1 και D_2 χρησιμοποιώντας τη μέθοδο που παρουσιάζεται στην Άσκηση 3 του Φροντιστηρίου 2.

Το ζητούμενο αυτόματο D φαίνεται πιο κάτω:

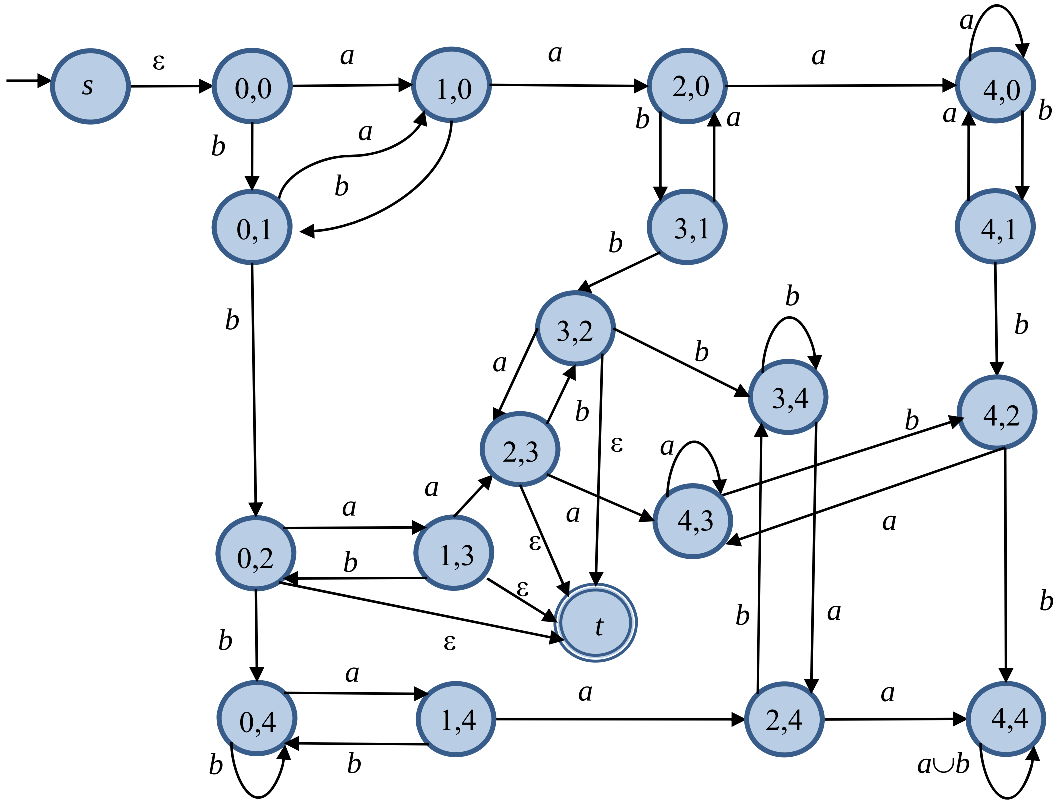


(iv) Να μετατρέψετε το αυτόματο D στην ισοδύναμη κανονική έκφραση χρησιμοποιώντας τον σχετικό αλγόριθμο (Διαφάνεια 3-20).

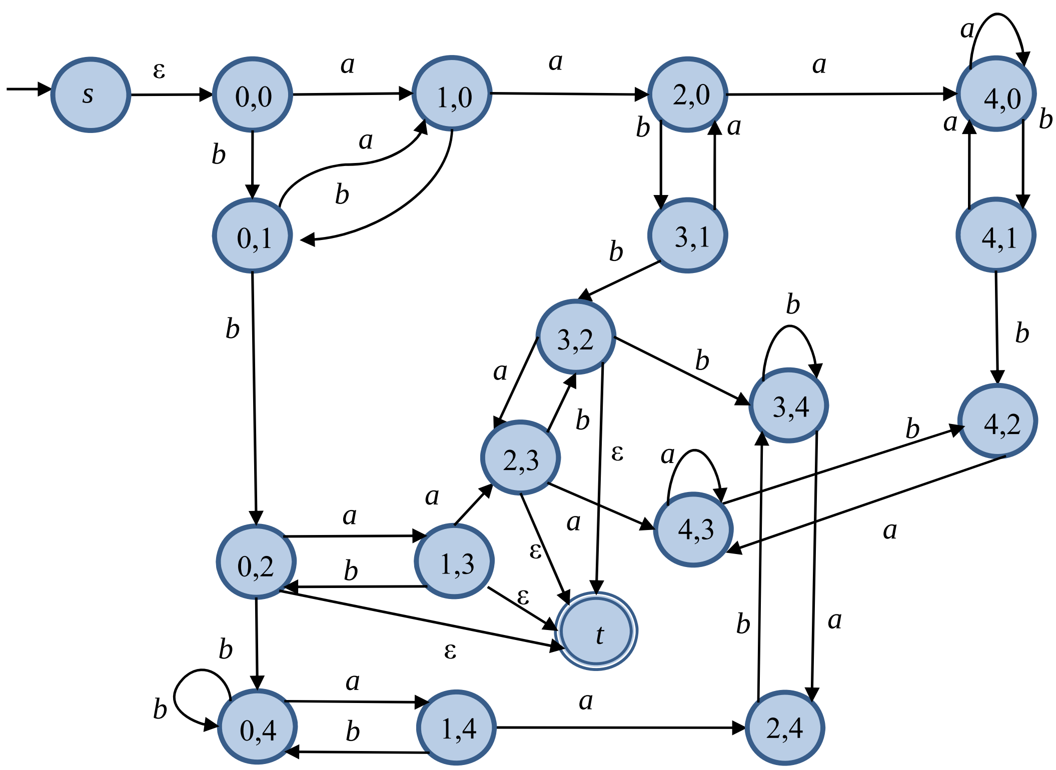
[Σημείωση: Επειδή το αυτόματο D είναι πολύ μεγάλο σε μέγεθος, στην άσκηση αυτή να εφαρμόσετε μόνο 5 επαναλήψεις του αλγόριθμου, αφαιρώντας 5 κορυφές που μπορείτε να επιλέξετε όπως επιθυμείτε.]

Λύση

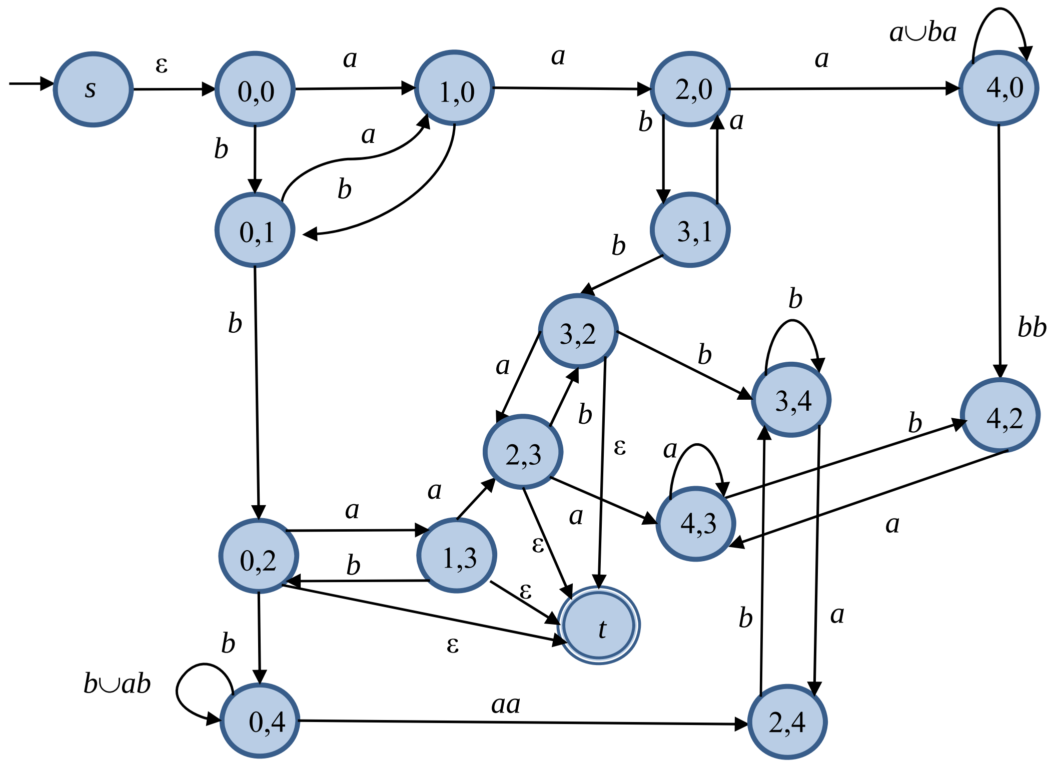
Εισάγουμε καινούρια αρχική κατάσταση και καινούρια τελική κατάσταση δημιουργώντας τις κατάλληλες συνδέσεις και μετατρέποντας το αυτόματο σε GNFA.



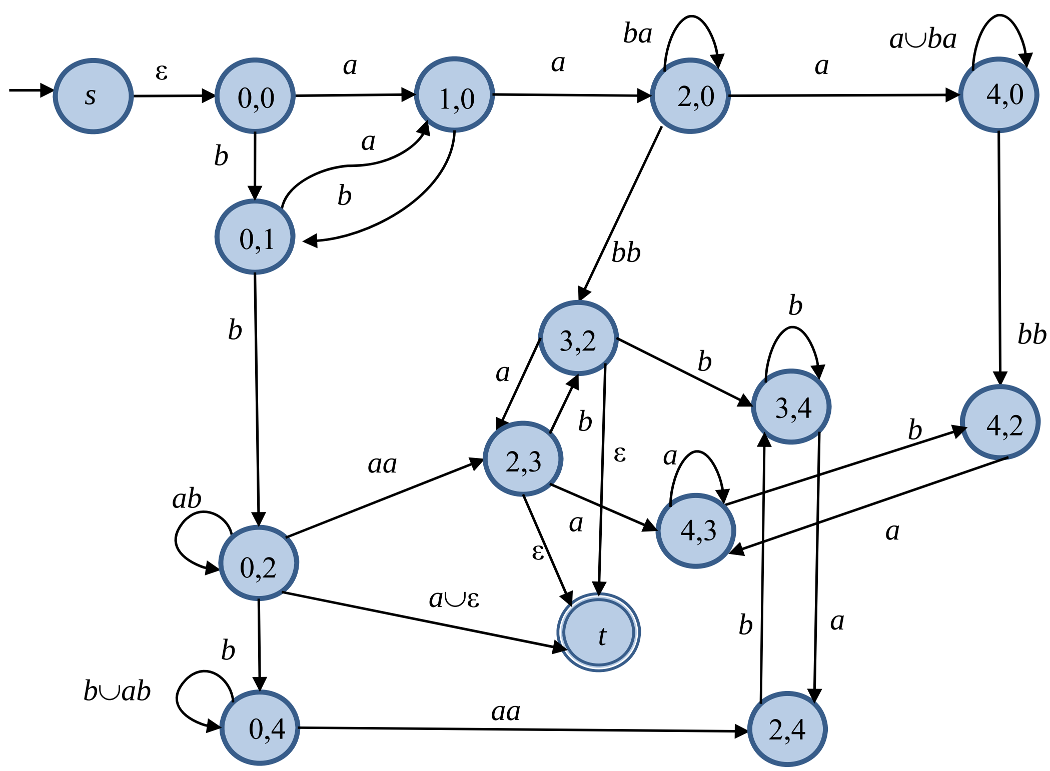
Βήμα 1: Αφαίρεση της κορυφής (4,4)



<u>Βήμα 2 και 3: Αφαίρεση των κορυφων (1,4) και (4,1)</u>



<u>Βήμα 4 και 5: Αφαίρεση των κορυφων (1,3) και (3,1)</u>



Άσκηση 3

Να αποφασίσετε κατά πόσο οι πιο κάτω γλώσσες είναι κανονικές αιτιολογώντας με ακρίβεια τις απαντήσεις σας.

(α) $\{ x \mid x = 10^m + 10^n + 1, m > n \geq 1\}$

(β) $\{ x \mid x = 10^{2n} + 10^n + 1, n \geq 1\}$

(γ) $\{ uv^n w \mid u,v,w \in \{0,1,2\}, u \neq v, v \neq w, u \neq w, n \geq 2 \}$

(δ) $\{ uv^n w \mid u,v,w \in \{0,1,2\}^*, u \neq v, v \neq w, u = w, n \geq 2 \}$

(ε) $\{ a^{n^2+2n} \mid n \geq 0 \}$

Λύση

(α) Η γλώσσα είναι κανονική αφού μπορεί να τύχει περιγραφής μέσω της κανονικής έκφραση 10*10*1.

(β) Υποθέτουμε για να φτάσουμε σε αντίφαση ότι η $L_2 = \{ x \mid x = 10^{2n} + 10^n + 1, n \geq 1\}$ είναι κανονική. Τότε, σύμφωνα με το Λήμμα της Άντλησης, υπάρχει *p*, το μήκος άντλησης της γλώσσας, τέτοιο ώστε κάθε λέξη της γλώσσας με μήκος μεγαλύτερο από *p* να ικανοποιεί την ιδιότητα που περιγράφεται στο Λήμμα.
Ας επιλέξουμε τη λέξη $10^{2p} + 10^p + 1$ που μας δίνει την έκφραση $10^{p-1}10^{p-1}1$. Τότε, σύμφωνα με το λήμμα, $s = xyz$ έτσι ώστε η υπολέξη xy να βρίσκεται μέσα στις *p* πρώτες θέσεις της *w* ($|xy| \leq p$) η *y* να είναι μη κενή ($|y| \neq 0$) και οποιαδήποτε επανάληψη της υπολέξης *y* να διατηρεί την προκύπτουσα λέξη εντός της γλώσσας ($xy^iz \in L_2$).

Αφού $|xy| \leq p$, τότε πρέπει να ισχύει ότι $x = 1^\kappa 0^\lambda$, $y = 1^\mu 0^\nu$, $0^{p-1-\lambda-\nu}10^{p-1}1$ όπου $\mu = 1$ αν x=ε διαφορετικά μ = 0.

Επίσης, από το λήμμα, πρέπει να ισχύει ότι $xy^2z \in L_2$.

Αλλά, αν το y περιέχει 1, τότε η λέξη xy^2z θα ξεκινά με 11 ενώ αν περιέχει 0, τότε η λέξη θα έχει τη μορφή $10^{p-1+\nu}10^{p-1}1$. Και στις δύο περιπτώσεις, από τον ορισμό της γλώσσας, $xy^2z \notin L_2$.

Αυτό μας οδηγεί σε αντίφαση και επομένως η υπόθεσή μας ότι η γλώσσα L_2 είναι κανονική ήταν εσφαλμένη.

(γ) Η γλώσσα είναι κανονική: μπορούμε να την αναπαραστήσουμε με την κανονική έκφραση

$$011^+2 \cup 211^+0 \cup 100^+2 \cup 200^+1 \cup 022^+1 \cup 122^+0$$

(δ) Υποθέτουμε για να φτάσουμε σε αντίφαση ότι η $L_4 = \{ uv^n w \mid u,v,w \in \{0,1,2\}^*, u \neq v, v \neq w, u = w, n \geq 2 \}$ είναι κανονική.

Τότε, σύμφωνα με το Λήμμα της Άντλησης, υπάρχει *p*, το μήκος άντλησης της γλώσσας, τέτοιο ώστε κάθε λέξη της γλώσσας με μήκος μεγαλύτερο από *p* να ικανοποιεί την ιδιότητα που περιγράφεται στο Λήμμα.

Ας επιλέξουμε τη λέξη $0^p1^p0^p$. Τότε, σύμφωνα με το λήμμα, $s = xyz$ έτσι ώστε η υπολέξη *xy* να βρίσκεται μέσα στις *p* πρώτες θέσεις της *w* ($|xy| \leq p$) η *y* να είναι μη κενή ($|y| \neq 0$) και οποιαδήποτε επανάληψη της υπολέξης *y* να διατηρεί την προκύπτουσα λέξη εντός της γλώσσας ($xy^iz \in L_4$).
Αφού $|xy| \leq p$, τότε πρέπει να ισχύει ότι $x = 0^\lambda$, $y = 0^\mu$, $z = 0^{p-\lambda-\mu}1^p0^p$.

Επίσης, από το λήμμα, πρέπει να ισχύει ότι $xy^2z \in L_4$. Αλλά $xy^2z = 0^{\lambda}0^{2\mu}0^{p-\lambda-\mu}1^p0^p = 0^{p+\mu}1^p0^p$. Από τον ορισμό της γλώσσας, $xy^2z \notin L_4$.

Αυτό μας οδηγεί σε αντίφαση και επομένως η υπόθεσή μας ότι η γλώσσα L_4 είναι κανονική ήταν εσφαλμένη.

(ε) Υποθέτουμε για να φτάσουμε σε αντίφαση ότι η $L_5 = \{\, a^{n^2+2n} \mid n \geq 0 \,\}$ είναι κανονική.

Τότε, σύμφωνα με το Λήμμα της Άντλησης, υπάρχει *p*, το μήκος άντλησης της γλώσσας, τέτοιο ώστε κάθε λέξη της γλώσσας με μήκος μεγαλύτερο από *p* να ικανοποιεί την ιδιότητα που περιγράφεται στο Λήμμα.

Ας επιλέξουμε τη λέξη a^{p^2+2p}. Τότε, σύμφωνα με το λήμμα, $s = xyz$ έτσι ώστε η υπολέξη xy να βρίσκεται μέσα στις p πρώτες θέσεις της w ($|xy| \leq p$) η y να είναι μη κενή ($|y| \neq 0$) και οποιαδήποτε επανάληψη της υπολέξης y να διατηρεί την προκύπτουσα λέξη εντός της γλώσσας ($xy^iz \in L_5$).
Αφού $|xy| \leq p$, τότε πρέπει να ισχύει ότι $x = a^{\lambda}$, $y = a^{\mu}$, $z = a^{p^2+2p-\lambda-\mu}$ και $\lambda+\mu \leq p$.

Επίσης, από το λήμμα, πρέπει να ισχύει ότι $xy^2z \in L_5$.

Έχουμε ότι $xy^2z = a^{p^2+2p+\mu}$.

Για να ανήκει η λέξη στη γλώσσα πρέπει $p^2 + 2p + \mu = q^2 + 2q$ για κάποιο ακέραιο q. Αλλά

$p^2 + 2p + \mu > p^2 + 2p$ και

$p^2 + 2p + \mu < (p+1)^2 + 2(p+1) = p^2 + 4p + 3$ για τον λόγο ότι $\mu \leq p$.

Αφού η ποσότητα $p^2 + 2p + \mu$ βρίσκεται ανάμεσα στις δύο τιμές, και δεν ισούται με καμιά από αυτές, είναι αδύνατο να υπάρχει q τέτοιο ώστε $p^2 + 2p + \mu = q^2 + 2q$. Αυτό μας οδηγεί στο συμπέρασμα ότι $xy^2z \notin L_5$ και επομένως η υπόθεσή μας ότι η γλώσσα L_5 είναι κανονική ήταν εσφαλμένη.

<u>Άσκηση 4</u>

Η έννοια των ανταιτιοκρατικών αυτομάτων (NFA) μπορείς να γενικευθεί ώστε, αντί να υπάρχει μια μόνο αρχική κατάσταση, να υπάρχει ένα μη-κενό σύνολο από αρχικές καταστάσεις. Σε αυτή την περίπτωση θεωρούμε ότι το γενικευμένο NFA αποδέχεται μια λέξη αν υπάρχει μονοπάτι που ξεκινά από κάποια αρχική κατάσταση και καταλήγει σε κάποια τελική κατάσταση.

Να αποδείξετε ότι η κλάση των γλωσσών που αποδέχονται τα γενικευμένα NFA είναι η ίδια με την κλάση των γλωσσών που αποδέχονται τα αυθεντικά NFA.

(Υπόδειξη: Αρκεί να δείξετε ότι για κάθε γενικευμένο NFA (με περισσότερες από μια αρχική καταστάσεις) υπάρχει ισοδύναμο αυτόματο με μια μοναδική αρχική κατάσταση.

<u>Λύση</u>

Τυπικά, η γενίκευση στην οποία αναφέρεται η άσκηση είναι αυτόματα της μορφής $(Q, \Sigma, \delta, I, F)$, όπου

1. Q είναι το σύνολο με τις *καταστάσεις* του αυτόματου,
2. Σ είναι το *αλφάβητο*,

3. $\delta: Q \times \Sigma_\varepsilon \to \mathcal{P}(Q)$, είναι η *συνάρτηση μεταβάσεων*,
4. $\emptyset \subset I \subseteq Q$ είναι ένα μη-κενό σύνολο από *εναρκτήριες καταστάσεις*
5. $F \subseteq Q$ είναι το *σύνολο των καταστάσεων αποδοχής* (τελικές καταστάσεις).

και κάποιο αυτόματο $M = (Q,\ \Sigma,\ \delta,\ I, F)$ ***αποδέχεται*** μια λέξη w αν αυτή μπορεί να γραφτεί στη μορφή $w = y_1 y_2 \ldots y_m$ όπου $y_i \in \Sigma_\varepsilon^n$ και υπάρχει ακολουθία καταστάσεων $r_0, r_1, \ldots, r_n$ του Q που να ικανοποιεί τις συνθήκες:

- $r_0 \in I$
- $r_{i+1} \in \delta(r_i, w_{i+1})$ για $i = 0,\ldots,n-1$, και
- $r_n \in F$

Για να δείξουμε ότι η κλάση των γλωσσών που αποδέχονται τα συγκεκριμένα γενικευμένα αυτόματα είναι η ίδια με την κλάση των γλωσσών που αποδέχονται τα αυθεντικά πρέπει να δείξουμε ότι για κάθε γενικευμένο αυτόματο υπάρχει ισοδύναμο αυθεντικό, και αντίστροφα.

Κατ' αρχάς, παρατηρούμε ότι κάθε αυθεντικό αυτόματο είναι και ένα γενικευμένο αυτόματο, όπου το σύνολο I περιέχει μια κατάσταση. Για την αντίθετη κατεύθυνση, ας θεωρήσουμε κάποιο γενικευμένο αυτόματο $M = (Q, \Sigma,\ \delta,\ I, F)$, θα κτίσουμε ένα αυθεντικό αυτόματο ως $N = (Q \cup \{q_0\},\ \Sigma,\ \delta',\ q_0, F)$ όπου

- q_0 , μια καινούρια κατάσταση η οποία παίζει το ρόλο της μοναδικής αρχικής στο αυθεντικό αυτόματο N, και
- $\delta'(s,a) = \begin{cases} \{q \mid q \in I\}, & \alpha\nu\ s = q_0\ \kappa\alpha\iota\ a = \varepsilon \\ \delta(s,a), & \alpha\nu\ s \neq q_0 \text{ ή } a = \varepsilon \\ \varepsilon, & \alpha\nu\ s = q_0\ \kappa\alpha\iota\ a \neq \varepsilon \end{cases}$
  Με λόγια το καινούριο αυτόματο συμπεριφέρεται ακριβώς όπως το αρχικό με μόνη διαφορά ότι από την αρχική κατάσταση μπορεί να μεταβεί στις αρχικές καταστάσεις του Μ μέσω της λέξης ε.

Μπορούμε να αποδείξουμε ότι τα Μ και Ν αποδέχονται ακριβώς τις ίδιες λέξεις:

Ας υποθέσουμε ότι $w = w_1 w_2 \ldots w_n \in L(M)$. Τότε, υπάρχει ακολουθία καταστάσεων $r_0, r_1, \ldots, r_n$ του Q που ικανοποιεί τις συνθήκες:

1. $r_0 \in I$
2. $r_{i+1} \in \delta(r_i, w_{i+1})$, για $i = 0,\ldots,n-1$, και
3. $r_n \in F$

Αν θεωρήσουμε την ακολουθία καταστάσεων $q_0, r_0, r_1, \ldots, r_n$ στο αυτόματο Ν, τότε έχουμε ότι για $w = \varepsilon\, w_1 w_2 \ldots w_n$ η ακολουθία αυτή ικανοποιεί τις συνθήκες:

1. $r_0 \in \delta'(q_0, \varepsilon)\ \kappa\alpha\iota\ r_{i+1} \in \delta'(s_i, w_{i+1})$, για $i = 0,\ldots,n-1$, και
2. $r_n \in F$.

Αυτό συνεπάγεται ότι $w \in L(N)$. Επομένως, αν $w \in L(M)$ τότε $w \in L(N)$.
Αντιστρέφοντας τα πιο πάνω επιχειρήματα, λαμβάνουμε ότι αν $w \in L(N)$ τότε $w \in L(M)$, και το ζητούμενο έπεται.

<u>Άσκηση 5</u>

(α) Να προτείνετε αλγόριθμο ο οποίος με δεδομένο εισόδου μια κανονική έκφραση να αποφασίζει κατά πόσο η γλώσσα που περιγράφεται από την κανονική έκφραση είναι μη πεπερασμένη.

(β) Να προτείνετε αλγόριθμο ο οποίος με δεδομένο εισόδου μια κανονική έκφραση να αποφασίζει κατά πόσο η γλώσσα που περιγράφεται από την κανονική έκφραση είναι η $\emptyset$.

<u>Λύση</u>

(α) Για να είναι η γλώσσα της πεπερασμένης έκφρασης μη πεπερασμένη, πρέπει η έκφραση να περιέχει τουλάχιστον μια εφαρμογή της πράξης της σώρευσης σε σύνολο διάφορο των $\emptyset$ και ε όπως επίσης και εκφράσεων οι οποίες αντιστοιχούν σε αυτά τα δύο σύνολα: $\emptyset^* = \{\varepsilon\}$ που είναι πεπερασμένη γλώσσα και $\varepsilon^* = \{\varepsilon\}$, που είναι πεπερασμένη γλώσσα. (Και παρόμοια $[(\emptyset\cup\varepsilon)^*]^* = \{\varepsilon\}$, που είναι πεπερασμένο σύνολο, και ούτω καθεξής.)

Για να βεβαιωθούμε ότι μια γλώσσα είναι διάφορη των $\emptyset$ και ε ορίζουμε τη βοηθητική συνάντηση όπου γράφουμε regex για το «τύπο» *κανονική έκφραση*:

```
bool nottrivial (regex R){
      case of R
            ∅:          return False
            ε:          return False
            a:          return True
            R1∪R2:      return nottrivial(R1) ∨ nottrivial(R2)
            R1R2:       return [nottrivial(R1) ∧ notempty(R2)
                                  ∨  notempty(R1) ∧ nottrivial(R2)
            R*:         return nottrivial(R)
}
```

Η διαδικασία αυτή απαιτεί να γνωρίζουμε ότι μια γλώσσα είναι διάφορη της $\emptyset$. Αυτό χρειάζεται για να μπορέσουμε να αναγνωρίσουμε ότι η συναρμογή δύο κανονικών εκφράσεων δεν είναι κενή. Η διαδικασία ορίζεται ως:

```
bool notempty (regex R){
      case of R
            ∅:          return False
            ε:          return True
            a:          return True
            R1∪R2:      return [notempty(R1) ∨ notempty(R2)]
            R1R2:       return [notempty(R1) ∧ notempty(R2)]
            R*:         return True
}
```

Επομένως ο ζητούμενος αλγόριθμος περιγράφεται από την πιο κάτω αναδρομική διαδικασία:

```
bool infinite(regex R){
      case of R
            ∅:          return False
            ε:          return False
            a:          return False
            R1∪R2:      return infinite(R1)∨ infinite(R2)
```

```
            R1R2:       return infinite(R1)∨ infinite(R2)
            R*:         return nottrivial(R)
}
```

(β) Ο ζητούμενο αλγόριθμος χρησιμοποιεί τη διαδικασία notempty που ορίζεται στο σκέλος (α) και έχει ως εξής:

```
bool empty (regex R){
     return !notempty(R)
}
```