Τα Θεμέλια της Πληροφορικής

Στάθης Ζάχος Άρης Παγουρτζής

Περιεχόμενα

Κεφάλαιο 7

Πεπερασμένα αυτόματα και κανονικές παραστάσεις

7.1 Εισαγωγή

Στο κεφάλαιο αυτό θα ασχοληθούμε με τυπικές γλώσσες που μπορούν να περιγράψουν υπολογιστικά προβλήματα και επίσης χρησιμεύουν στον προγραμματισμό. Επίσης θα επιδιώξουμε να ταξινομήσουμε γλώσσες ανάλογα με το είδος του **αυτομάτου** (*αφηρημένης υπολογιστικής συσκευής*) που μπορεί να τις αναγνωρίσει.

Ένα αυτόματο έχει μερικές εσωτερικές καταστάσεις: $q_0, q_1, q_{16}, q_{23}, \ldots$ και μία συνάρτηση δ που καθορίζει την επόμενη κατάσταση του αυτομάτου.

Στις πιο απλές συσκευές, που ονομάζονται **μηχανισμοί**, η συνάρτηση μετάβασης (*transition function*) δ έχει ως όρισμα μία κατάσταση q_i και ως τιμή μια άλλη κατάσταση q_j. Δεν υπάρχει δηλαδή είσοδος και έξοδος. Υπολογιστική ακολουθία σε αυτήν την περίπτωση είναι μια ακολουθία από καταστάσεις:

$$q_i \to q_j \to q_k \to q_l \cdots$$

Αν επιπλέον η συσκευή διαβάζει μια συμβολοσειρά εισόδου, σύμβολο προς σύμβολο, από αριστερά προς τα δεξιά και ανάλογα αλλάζει καταστάσεις, τότε ονομάζεται **αναγνωριστής πεπερασμένων καταστάσεων** (*FSA: finite state acceptor*). Η συνάρτηση μετάβασης είναι της μορφής

$$\delta\colon (q_i, a) \to q_j,$$

όπου $\alpha \in \Sigma$ και το Σ το αλφάβητο εισόδου. Στο τέλος το FSA αποδέχεται ή απορρίπτει την είσοδό του.

Αν προσθέσουμε έξοδο σε ένα *FSA*, δηλαδή $\delta\colon (q_i, a) \to (q_j, b)$, όπου $b \in \Delta$ και Δ το αλφάβητο εξόδου, τότε έχουμε τη λεγόμενη μηχανή πεπερασμένων καταστάσεων (*FSM: finite state machine*). Οι FSA και FSM εμφανίζονται συχνά ως χρήσιμα εργαλεία στο λογικό, ψηφιακό σχεδιασμό. Προσθέτοντας στο αυτόματό μας μνήμη υπό μορφή στοίβας (*stack*) έχουμε πολύ περισσότερες δυνατότητες: το αυτόματο ονομάζεται τότε **αυτόματο στοίβας** (*PDA: push-down automaton*). Αν αντί στοίβας έχουμε απεριόριστη δυνατότητα μνήμης υπό μορφή ταινίας, τότε έχουμε τη γνωστή **μηχανή Turing** (*TM*). **Γραμμικά περιορισμένο αυτόματο** (*LBA: linearly bounded automaton*) είναι μια μηχανή Turing που όμως μπορεί να χρησιμοποιήσει ταινία που το μήκος της είναι μια γραμμική συνάρτηση του μήκους της εισόδου.

Τα διαφορετικά αυτά αυτόματα αναγνωρίζουν κλάσεις γλωσσών που όμως μπορούν να περιγραφούν και με άλλους τρόπους π.χ. με αλγεβρικό τρόπο, ή με **τυπικές γραμματικές**.

Οι γλώσσες προγραμματισμού είναι προφανώς **τυπικές γλώσσες** (*formal languages*) που έχουν αυστηρό συντακτικό ορισμό. Οι τυπικές γλώσσες μπορούν να ταξινομηθούν ανάλογα με τις τυπικές γραμματικές που τις παράγουν. Στη θεωρία τυπικών γλωσσών οι πρωταρχικές έννοιες είναι τα **σύμβολα** (ως αντικείμενα) και η **παράθεση** (ως πράξη).

Ένα **αλφάβητο** Σ είναι ένα πεπερασμένο σύνολο συμβόλων. Μια λέξη ή πρόταση ή συμβολοσειρά ή string είναι μια πεπερασμένου μήκους ακολουθία συμβόλων. Το μήκος του string w συμβολίζεται με $|w|$. Το κενό string συμβολίζεται με ε. Η παράθεση των strings x και y συμβολίζεται με xy. ’Αλλοι χρήσιμοι όροι είναι **πρόθεμα** (*prefix*), **υποσυμβολοσειρά** (*substring*), **υπακολουθία** (*subsequence*), **κατάληξη** (*suffix*), **αντίστροφη** (*reversal*), **παλινδρομική** ή **καρκινική** (*palindrome*).

Ισχύουν $x\varepsilon = \varepsilon x = x$ για όλα τα strings x και $|\varepsilon| = 0$. Το string x^k μπορεί να οριστεί με πρωταρχική αναδρομή:

$$\begin{cases} x^0 = \varepsilon \\ x^{k+1} = x^k x \end{cases}$$

Ορισμός 7.1. Αν Σ είναι ένα αλφάβητο τότε Σ^* είναι το σύνολο όλων των strings από το Σ. Μια γλώσσα L από το Σ δεν είναι παρά κάποιο υποσύνολο του Σ^*.

Παραδείγματα (με αλφάβητο $\Sigma = \{a, b\}$):

- $L_1 = \{w \in \Sigma^* \mid w \text{ αρχίζει με } a\}$
- $L_2 = \{w \in \Sigma^* \mid w \text{ περιέχει ζυγό αριθμό από } a\}$
- $L_3 = \{w \in \Sigma^* \mid w \text{ είναι παλινδρομική }\}$

7.2 Ντετερμινιστικά πεπερασμένα αυτόματα

Θα περιγράψουμε πρώτα τα *ντετερμινιστικά πεπερασμένα αυτόματα* (deterministic finite automata - DFA). Ένα DFA είναι ένα αυτόματο που *αναγνωρίζει* (δηλαδή *αποδέχεται* ή *απορρίπτει*) συμβολοακολουθίες που εμφανίζονται στην ταινία εισόδου του. Η κεφαλή[1] του DFA βρίσκεται αρχικά στο αριστερότερο σημείο της εισόδου. Μετακινείται δε προς τα δεξιά κατά ένα σύμβολο σε κάθε βήμα μέχρι να σταματήσει στο τέλος της ταινίας εισόδου. Από σύμβαση, η αρχική κατάσταση (initial state) ονομάζεται q_0, κάποιες από τις καταστάσεις αποδέχονται και ονομάζονται τελικές (accepting, final states), και οι υπόλοιπες καταστάσεις απορρίπτουν (rejecting, nonfinal states). Αν το DFA βρίσκεται σε μια κατάσταση που αποδέχεται αφότου η κεφαλή έχει διαβάσει όλη την συμβολοακολουθία είσοδου, λέμε ότι το DFA *αποδέχεται* την είσοδο. Η γλώσσα L, την οποία αναγνωρίζει ένα DFA, είναι το σύνολο των συμβολοακολουθιών που αποδέχεται.

Ορισμός 7.2. Τυπικά ένα DFA είναι μία πεντάδα $M = (Q, \Sigma, \delta, q_0, F)$, όπου:

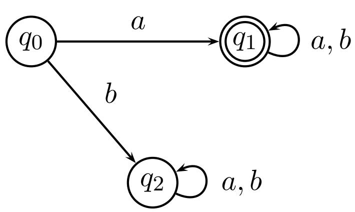
- Q: ένα πεπερασμένο σύνολο από καταστάσεις,
- Σ: ένα αλφάβητο εισόδου ($\Sigma \cap Q = \emptyset$),
- $\delta\colon Q \times \Sigma \to Q$: η συνάρτηση μετάβασης,
- $q_0 \in Q$: η αρχική κατάσταση,
- $F \subseteq Q$: το σύνολο των τελικών καταστάσεων.

Ενα DFA μπορεί επίσης να περιγραφεί (όπως και μια TM) από ένα *πίνακα καταστάσεων* ή από ένα *διάγραμμα καταστάσεων*

Στα παρακάτω παραδείγματα θεωρούμε αλφάβητο $\Sigma = \{a, b\}$.

Παράδειγμα 7.1. $L_1 = \{w \in \Sigma^* \mid w \text{ αρχίζει από } a\}$

Διάγραμμα:



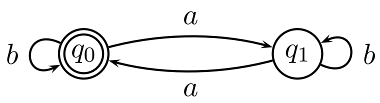
Πίνακας:

	a	b
q_0	q_1	q_2
q_1	q_1	q_1
q_2	q_2	q_2

Χρησιμοποιούμε έναν επιπλέον κύκλο για να δείξουμε τις τελικές καταστάσεις.

[1]ώς *κεφαλή* ορίζουμε εκείνο το στοιχείο της ταινίας εισόδου που βρίσκεται την παρούσα στιγμή υπό εξέταση

Παράδειγμα 7.2. $L_2 = \{w \in \Sigma^* \mid w$ περιέχει άρτιο πλήθος από $a\}$



	a	b
q_0	q_1	q_0
q_1	q_0	q_1

Παράδειγμα 7.3. $L_3' = \{w \in \Sigma^* \mid w$ παλίνδρομη αρτίου μήκους$\}$, δηλαδή $L_3' = \{ww^R \mid w \in \Sigma^*, w^R =$ αντίστροφη της $w\}$. Δεν υπάρχει DFA που αποδέχεται την L_3'.

Για να συντομεύσουμε την περιγραφή της συμπεριφοράς του DFA, επεκτείνουμε την συνάρτηση δ ώστε να δέχεται ως όρισμα συμβολοακολουθίες αντί για απλά σύμβολα. Η δ επεκτείνεται λοιπόν ως εξής:

Ορισμός 7.3. $\delta : Q \times \Sigma^* \to Q$ όπου

$$\begin{cases} \delta(q, \varepsilon) = q \\ \delta(q, wa) = \delta(\delta(q, w), a) \end{cases}$$

Ο πιο πάνω ορισμός είναι αναδρομικός, ή, πιο συγκεκριμένα, είναι ορισμός σύμφωνα με το σχήμα της *πρωταρχικής αναδρομής*.

Ορισμός 7.4. Έχουμε:

- Ένα DFA αποδέχεται το string $w \in \Sigma^*$ ανν $\delta(q_0, w) \in F$
- Ένα DFA αποδέχεται τη γλώσσα $L(M) = \{w \mid \delta(q_0, w) \in F\}$
- Η γλώσσα L λέγεται *κανονική* (*regular*) ανν $\exists$ FA $M : L = L(M)$

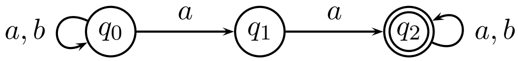
Άσκηση: Δείξτε ότι $\delta(q, uv) = \delta(\delta(q, u), v)$, όπου $u, v \in \Sigma^*$.

Υπάρχουν διάφορες γενικεύσεις και διαφοροποιήσεις ως προς τα DFA:

- NFA: *μη ντετερμινιστικό πεπερασμένο αυτόματο*. Σε κάθε μετάβαση υπάρχει επιλογή της επόμενης κατάστασης από ένα σύνολο δυνατών νομίμων καταστάσεων.
- NFA$_\varepsilon$: *μη ντετερμινιστικό πεπερασμένο αυτόματο με ε-κινήσεις*. Το πεπερασμένο αυτόματο ενδέχεται να αλλάζει την κατάστασή του χωρίς να μετακινείται η κεφαλή στην ταινία εισόδου.

7.3 Μη ντετερμινιστικά πεπερασμένα αυτόματα

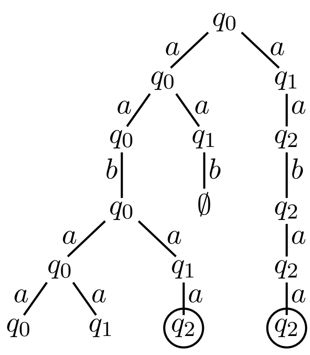
Παράδειγμα 7.4. NFA για $L_4 := \{w \in \Sigma^* \mid w$ περιέχει δύο συνεχόμενα $a\}$



	a	b
q_0	$\{q_0, q_1\}$	$\{q_0\}$
q_1	$\{q_2\}$	$\emptyset$
q_2	$\{q_2\}$	$\{q_2\}$

Ένας υπολογισμός σε ένα NFA δεν είναι απλώς μία γραμμική (νόμιμη) ακολουθία καταστάσεων, αλλά ένα *υπολογιστικό δένδρο* (κάθε κλάδος είναι μία νόμιμη ακολουθία καταστάσεων).

Το δένδρο υπολογισμού για το παραπάνω παράδειγμα για είσοδο $aabaa$:



Η συμβολοσειρά $aabaa$ γίνεται αποδεκτή, επειδή υπάρχει τουλάχιστον ένα νόμιμο μονοπάτι που την αποδέχεται.

Στα μη ντετερμινιστικά αυτόματα, για κάθε είσοδο και κατάσταση, μπορεί να υπάρχει καμμία, μία ή πολλές νόμιμες επόμενες καταστάσεις. Αυτό εκφράζεται στον ορισμό ενός NFA από το γεγονός ότι η συνάρτηση μετάβασης δ έχει ως πεδίο τιμών το *δυναμοσύνολο* του Q ($\mathrm{Pow}(Q)$).

Ένα NFA λοιπόν αποτελείται από μια πεντάδα $M = (Q, \Sigma, \delta, q_0, F)$, όπου:

- Q: ένα πεπερασμένο σύνολο από καταστάσεις,
- Σ: ένα πεπερασμένο αλφάβητο εισόδου,
- $\delta : Q \times \Sigma \to \mathrm{Pow}(Q)$: η συνάρτηση μετάβασης
- $q_0 \in Q$: η αρχική κατάσταση και
- $F \subseteq Q$: το σύνολο των τελικών καταστάσεων.

Όπως και πριν, επεκτείνουμε την συνάρτηση μετάβασης δ για να συμπεριλάβουμε στο πεδίου ορισμού της τις συμβολοακολουθίες. Η επεκτεταμένη συνάρτηση μετάβασης $\delta(q, w)$ θα περιέχει το σύνολο όλων των καταστάσεων που μπορούμε να φτάσουμε από το q με είσοδο το w.

Ορισμός 7.5. $\delta : Q \times \Sigma^* \to \text{Pow}(Q)$ όπου

$$\begin{cases} \delta(q, \varepsilon) = \{q\} \\ \delta(q, wa) = \{p \in \delta(r, a) | r \in \delta(q, w)\} = \bigcup\limits_{r \in \delta(q,w)} \delta(r, a) \end{cases}$$

Επεκτείνουμε περαιτέρω την δ, συμπεριλαμβάνοντας στο πεδίο ορισμού το δυναμοσύνολο του Q. Η $\delta(P, w)$ θα περιέχει όλες τις καταστάσεις που μπορούμε να φτάσουμε από οποιαδήποτε κατάσταση στο P με είσοδο w.

Ορισμός 7.6. $\delta : \text{Pow}(Q) \times \Sigma^* \to \text{Pow}(Q)$ όπου

$$\delta(P, w) = \{p \in \delta(q, w) | q \in P\} = \bigcup_{q \in P} \delta(q, w)$$

Παρατήρηση 7.1. Ισχύει $\delta(q, wa) = \delta(\delta(q, w), a)$

Παρατήρηση 7.2. Ισχύει $\delta(P, wa) = \delta(\delta(P, w), a)$

Ορισμός 7.7. Έχουμε:

- Ένα NFA αποδέχεται το string $w \in \Sigma^*$ ανν $\delta(q_0, w) \cap F \neq \emptyset$
- Ένα NFA M αποδέχεται τη γλώσσα $L(M) = \{w | \delta(q_0, w) \cap F \neq \emptyset\}$

Ισοδυναμία DFA και NFA. Όπως φαίνεται από τον ορισμό της δ ενός NFA, ένα DFA είναι μια "υποπερίπτωση" ενός NFA. Παρ' όλα αυτά, τα NFA δεν μας παρέχουν περισσότερες δυνατότητες υπολογισμού από ότι τα DFA. Αυτό αποδεικνύει το παρακάτω θεώρημα:

Θεώρημα 7.5. *(Rabin - Scott)*

Έστω M ένα NFA. Τότε $\exists$ DFA $M' : L(M) = L(M')$

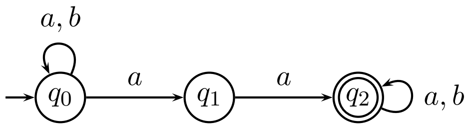
Απόδειξη. Ορίζουμε το DFA $M' = (Q', \Sigma', q_0', F', \delta')$ όπου

- $Q' = \text{Pow}(Q)$,

- $\Sigma' = \Sigma$, $q_0' = \{q_0\}$,
- $F' = \{R \in Q' | R \cap F \neq \emptyset\} = \bigcup_{R \cap F \neq \emptyset} (R \in Q')$ και τέλος
- $\delta'(R, a) = \delta(R, a)$, όπου $R \in Q'$.

Μένει να αποδειχθεί ότι $L(M) = L(M')$ με την βοήθεια του ισχυρισμού: $\delta'(q_0', w) = \delta(q_0, w)$. (Αφήνεται ως άσκηση· χρησιμοποιήστε επαγωγή.) $\square$

Παράδειγμα 7.6.

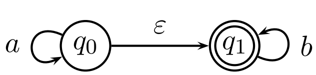


	a	b
$\emptyset$	$\emptyset$	$\emptyset$
$\{q_0\}$	$\{q_0, q_1\}$	$\{q_0\}$
$\{q_1\}$	$\{q_2\}$	$\emptyset$
$\{q_2\}$	$\{q_2\}$	$\{q_2\}$
$\{q_0, q_1\}$	$\{q_0, q_1, q_2\}$	$\{q_0\}$
$\{q_0, q_2\}$	$\{q_0, q_1, q_2\}$	$\{q_0, q_2\}$
$\{q_1, q_2\}$	$\{q_2\}$	$\{q_2\}$
$\{q_0, q_1, q_2\}$	$\{q_0, q_1, q_2\}$	$\{q_0, q_2\}$

Το DFA φαίνεται στο παράδειγμα 7.9.

Μη ντετερμινιστικά αυτόματα με ε-κινήσεις - NFA$_\varepsilon$

Παράδειγμα 7.7. NFA$_\varepsilon$ για $L_5 := \{a^*b^*\} = \{a^n b^m \mid n, m \in \mathbb{N}\}$



	a	b	ε
q_0	$\{q_0\}$	$\emptyset$	$\{q_0, q_1\}$
q_1	$\emptyset$	$\{q_1\}$	$\{q_1\}$

Μπορούμε να γενικεύσουμε το μη ντετερμινιστικό αυτόματο που ορίσαμε ώστε να συμπεριλάβουμε στο πεδίο ορισμού της δ και το ε. Με άλλα λόγια, το αυτόματο μπορεί να αλλάζει κατάσταση χωρίς να διαβάζει κάποια είσοδο, με μια ε-κίνηση. Ένα NFA$_\varepsilon$ λοιπόν είναι μια πεντάδα $M = (Q, \Sigma, \delta, q_0, F)$ όπου

- Q: ένα πεπερασμένο σύνολο από καταστάσεις,
- Σ: ένα πεπερασμένο αλφάβητο εισόδου,
- $\delta : Q \times (\Sigma \cup \{\varepsilon\}) \to \mathrm{Pow}(Q)$: η συνάρτηση μετάβασης
- $q_0 \in Q$: η αρχική κατάσταση και
- $F \subseteq Q$: το σύνολο των τελικών καταστάσεων.

Για την ανάλυσή μας, είναι χρήσιμο να ορίσουμε το *ε-κλείσιμο* μιας κατάστασης.

Ορισμός 7.8. Ως ε-κλείσιμο: $Q \to \mathrm{Pow}(Q)$ ορίζουμε το

$$\varepsilon\text{-κλείσιμο}(q) = \{p| \text{ τα } p \text{ προσβάσιμα από το } q \text{ μόνο με } \varepsilon\text{-κινήσεις}\}$$

Παρατηρούμε ότι πάντα $q \in \varepsilon$-κλείσιμο(q). Επεκτείνουμε τον ορισμό αυτό:

Ορισμός 7.9. Ως ε-κλείσιμο: $\mathrm{Pow}(Q) \to \mathrm{Pow}(Q)$ ορίζουμε το

$$\varepsilon\text{-κλείσιμο}(P) = \bigcup_{q \in P} \varepsilon\text{-κλείσιμο}(q)$$

Παρατηρούμε επιπλέον ότι ε-κλείσιμο(ε-κλείσιμο(P)) $=$ ε-κλείσιμο(P). Η συνάρτηση μετάβασης δ μπορεί να επεκταθεί. Η επεκτεταμένη συνάρτηση μετάβασης $\delta(q, w)$ θα περιέχει το σύνολο όλων των καταστάσεων που μπορούμε να φτάσουμε από το q με είσοδο το w, καθώς και με οσεσδήποτε αρχικές, ενδιάμεσες και τελικές ε-κινήσεις.

Ορισμός 7.10. $\delta : Q \times \Sigma^* \to \mathrm{Pow}(Q)$ όπου

$$\begin{cases} \delta(q,\varepsilon) = \varepsilon\text{-κλείσιμο}(q) \\ \delta(q,wa) = \varepsilon\text{-κλείσιμο}(\{p \in \delta(r,a) \mid r \in \delta(q,w)\}) = \varepsilon\text{-κλείσιμο}(\bigcup_{r \in \delta(q,w)} \delta(r,a)) \end{cases}$$

Επιπλέον, επεκτείνουμε την δ, συμπεριλαμβάνοντας στο πεδίο ορισμού το δυναμοσύνολο του Q.

Ορισμός 7.11. $\delta : \mathrm{Pow}(Q) \times \Sigma^* \to \mathrm{Pow}(Q)$ όπου

$$\delta(P,w) = \{p \in \delta(q,w) | q \in P\} = \bigcup_{q \in P} \delta(q,w)$$

Ορισμός 7.12. Έχουμε:

- Ένα NFA_ε αποδέχεται το string $w \in \Sigma^*$ ανν $\delta(q_0, w) \cap F \neq \emptyset$
- Ένα NFA_ε αποδέχεται τη γλώσσα $L(M) = \{w | \delta(q_0, w) \cap F \neq \emptyset\}$

Ισοδυναμία NFA και NFA$_\varepsilon$. Και πάλι, μπορεί να θεωρήσει κανείς ότι τα NFA είναι μια “υποπερίπτωση” των NFA$_\varepsilon$. Όμως, όπως και πριν, τα NFA$_\varepsilon$ δεν έχουν περισσότερες δυνατότητες υπολογισμού από τα NFA, όπως αποδεικνύει το επόμενο θεώρημα, και άρα και από τα DFA.

Θεώρημα 7.8. *Έστω M ένα NFA$_\varepsilon$ τότε* $\exists$ NFA M' : $L(M) = L(M')$

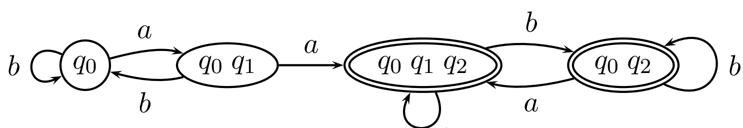
Απόδειξη. Ορίζουμε το NFA $M' = (Q, \Sigma, q_0, F', \delta')$ όπου

$$F' = \begin{cases} F \cup \{q_0\}, & \text{αν } \varepsilon\text{-κλείσιμο}(q_0) \cap F \neq \emptyset \\ F, & \text{ειδάλλως} \end{cases} \quad , \qquad \delta'(q,a) = \delta(q,a)$$

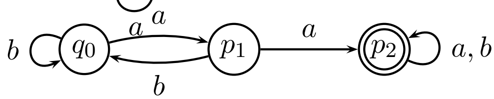
Πλέον, προκειμένου να ισχύει $L(M) = L(M')$, αρκεί να αποδειχθεί ο ισχυρισμός: $\forall w \in \Sigma^* - \{\varepsilon\}$: $\delta'(q_0, w) = \delta(q_0, w)$. (’Ασκηση.) $\square$

Ακολουθούν μερικά παραδείγματα με μη ντετερμινιστικά αυτόματα και ισοδύναμά τους ντετερμινιστικά αυτόματα.

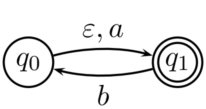
Παράδειγμα 7.9. DFA για L_4 (βλ. παραδ. 7.4):



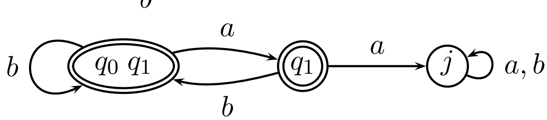
Ελάχιστο DFA για L_4:



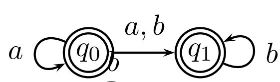
Παράδειγμα 7.10. NFA$_\varepsilon$ για $\overline{L_4}$ (δηλαδή “όχι δύο συνεχόμενα a”):



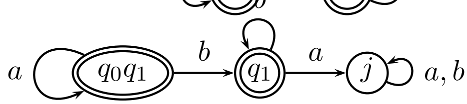
DFA για $\overline{L_4}$:



Παράδειγμα 7.11. NFA για L_5 (βλ. παράδ. 7.7):



Ελάχιστο DFA για L_5:



7.4 Κανονικές παραστάσεις

Παρακάτω, ορίζουμε κάποιες πράξεις επί των γλωσσών.

Ορισμός 7.13. Έστω L, L_1, L_2 γλώσσες επί του ίδιου αλφαβήτου Σ.

- $L_1 L_2 := \{uv \mid u \in L_1 \wedge v \in L_2\}$: παράθεση
- $L_1 \cup L_2 := \{w \mid w \in L_1 \vee w \in L_2\}$: ένωση
- $L_1 \cap L_2 := \{w \mid w \in L_1 \wedge w \in L_2\}$: τομή
- $L^0 := \{\varepsilon\}$, $L^{n+1} := LL^n$
- $L^* := \bigcup_{n=0}^{\infty} L^n$: άστρο του Kleene
- $L^+ := \bigcup_{n=1}^{\infty} L^n$

Τώρα πλέον μπορούμε να εισάγουμε επαγωγικά έναν συμβολισμό για τις κανονικές γλώσσες, τις λεγόμενες *κανονικές παραστάσεις*.

Ορισμός 7.14. Κανονική παράσταση είναι:

$\emptyset$: παριστάνει το κενό σύνολο·

ε: παριστάνει το $\{\varepsilon\}$·

a: παριστάνει το $\{a\}$, όπου $a \in \Sigma$·

$(r + s)$: παριστάνει το $R \cup S$, όπου r, s κανονικές παραστάσεις που παριστάνουν τα R, S αντιστοίχως·

(rs): παριστάνει το RS, όπου r, s κανονικές παραστάσεις που παριστάνουν τα R, S αντιστοίχως·

(r^*): παριστάνει το R^*, όπου r κανονική παράσταση που παριστάνει το R.

Σύμβαση: Μπορούμε να περιορίσουμε τις παρενθέσεις αν χρησιμοποιήσουμε την ακόλουθη προτεραιότητα των τελεστών: $*$, παράθεση, ένωση.

Παράδειγμα 7.12.

$L_1 = a(a + b)^*$

$L_2 = (b^* ab^* a)^* b^* = (b + ab^* a)^*$

L_3 δεν είναι δυνατόν να παρασταθεί με κανονική παράσταση

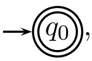
$$L_4 = (a+b)^*aa(a+b)^* \qquad \text{(τουλάχιστον δύο συνεχόμενα } a\text{)}$$
$$\overline{L_4} = (a+\varepsilon)(ba+b)^* \qquad \text{(όχι συνεχόμενα } a\text{)}$$
$$L_5 = a^*b^*$$

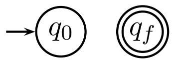
Θεώρημα 7.13. *Μία γλώσσα μπορεί να παρασταθεί με κανονική παράσταση ανν $L = L(M)$ για κάποιο πεπερασμένο αυτόματο M.*

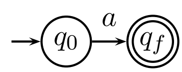
Ιδέα απόδειξης.

$\Rightarrow$ Επαγωγή στην δομή της κανονικής παράστασης. Έστω r η κανονική παράσταση.

1. Επαγωγική Βάση:

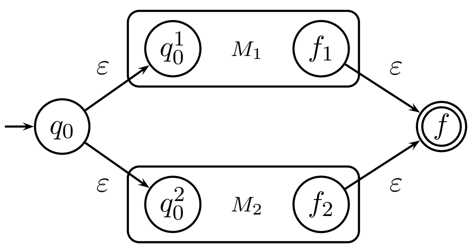
   $r = \varepsilon$: 

, $r = \emptyset$: 

, $r = a \in \Sigma$: 

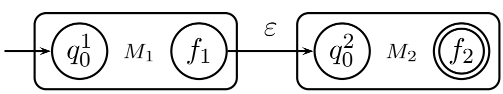
2. Επαγωγικό Βήμα: Υποθέτουμε ότι οι $L(r_1)$ και $L(r_2)$ αναγνωρίζονται από NFA$_\varepsilon$ M_1, M_2 αντιστοίχως με τελική κατάσταση f_1, f_2 αντιστοίχως.

   Περίπτωση α: $r = r_1 + r_2$



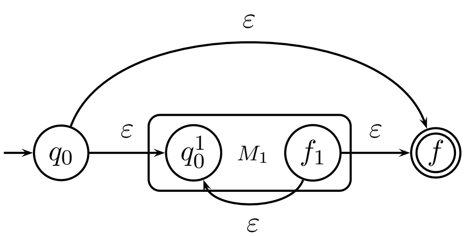
$L(M) = L(M_1) \cup L(M_2)$

   Περίπτωση β: $r = r_1 r_2$



$L(M) = L(M_1)L(M_2)$

   Περίπτωση γ: $r = r_1^*$



$L(M) = L(M_1)^*$

$\Leftarrow$ Θεωρούμε ότι το DFA είναι της μορφής $= (Q, \Sigma, \delta, q_1, F)$, όπου τα στοιχεία του Q είναι αριθμημένα κατά αύξουσα σειρά και η αρχική κατάσταση είναι η q_1, δηλαδή $Q = \{q_1, \ldots, q_n\}$, όπου $n = |Q|$. Ορίζουμε:

$$R_{ij}^k = \{w \mid \tilde{\delta}(q_i, w) = q_j \text{ και } \forall x \text{ πρόθεμα του } w \text{ με } x \neq w, \varepsilon\colon \tilde{\delta}(q_i, x) = q_l \Rightarrow l \leq k\}$$

Δηλαδή, R_{ij}^k είναι το σύνολο των συμβολοακολουθιών που οδηγούν από την κατάσταση q_i στην q_j χωρίς να περνούν από οποιαδήποτε κατάσταση με δείκτη μεγαλύτερο από k. Προφανώς, αφού δεν υπάρχει κατάσταση με δείκτη μεγαλύτερο από n, το R_{ij}^n περιέχει όλες τις συμβολοακολουθίες από την q_i στην q_j. Μπορούμε να υπολογίσουμε το R_{ij}^k αναδρομικά, σύμφωνα με μια ιδέα των Floyd και Warshall, ως εξής:

Αλγόριθμος Floyd-Warshall

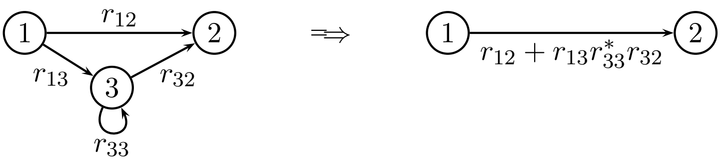
$$R_{ij}^0 = \begin{cases} \{a | \delta(q_i, a) = q_j\}, & \text{αν } i \neq j \\ \{a | \delta(q_i, a) = q_j\} \cup \{\varepsilon\}, & \text{αν } i = j \end{cases}$$
$$R_{ij}^k = R_{ik}^{k-1} (R_{kk}^{k-1})^* R_{kj}^{k-1} \cup R_{ij}^{k-1}$$

Τέλος, αρκεί να παρατηρήσουμε ότι $L(M) = \bigcup_{q_j \in F} R_{1j}^n$. Η ιδέα των Floyd και Warshall είναι ότι το R_{ij}^k αποτελείται από συμβολοακολουθίες που οδηγούν από την κατάσταση q_i στην q_j είτε χωρίς να περνούν από την κατάσταση q_k (οπότε περιέχονται και στο R_{ij}^{k-1}, είτε περνούν από την κατάσταση q_k μία ή περισσότερες φορές (οι λεπτομέρειες της απόδειξης αφήνονται ως άσκηση.) $\square$

Κατασκευή κανονικής παράστασης από FA Το 2ο σκέλος της πιο πάνω απόδειξης δίνει έναν πλήρη, συστηματικό, αλλά συχνά χρονοβόρο, τρόπο κατασκευής της κανονικής παράστασης που αντιστοιχεί σε ένα FA. Παρακάτω δίνεται ένας πιο γρήγορος τρόπος που βασίζεται στην έννοια του GNFA: γενικευμένο αυτόματο, του οποίου η συνάρτηση μετάβασης δέχεται και κανονικές παραστάσεις (στα βέλη του μπορούμε να γράφουμε κανονικές παραστάσεις αντί για σκέτα σύμβολα).

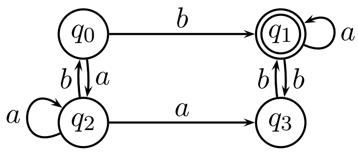
Κατ' αρχάς, θεωρούμε ότι το FA έχει μια αρχική και μια τελική κατάσταση, διαφορετικές μεταξύ τους. Αν υπάρχουν πάνω από μια τελικές καταστάσεις, προσθέτουμε μια νέα κατάσταση στην οποία βαίνουν όλες με ε-κινήσεις και στη συνέχεια τις κάνουμε μη τελικές. Αντίστοιχα μετατρέπουμε και την αρχική κατάσταση σε μη τελική, αν χρειάζεται.

Φέρνοντας το FA μας σε αυτή τη μορφή, μπορούμε να απαλείψουμε μία μία τις ενδιάμεσες καταστάσεις σύμφωνα με το πιο κάτω σχήμα:

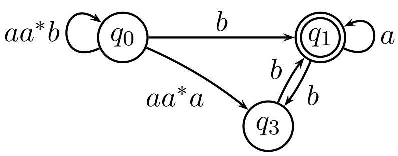


Με διαδοχικές απαλοιφές οδηγούμαστε τελικά στο επιθυμητό GNFA (γενικευμένο αυτόματο, του οποίου η συνάρτηση μετάβασης δέχεται και κανονικές παραστάσεις). Στη συνέχεια θα δώσουμε ένα παράδειγμα με όλα τα ενδιάμεσα βήματα υπολογισμού.

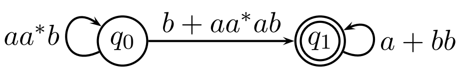
Παράδειγμα 7.14. ’Εστω πως έχουμε το ακόλουθο FA:



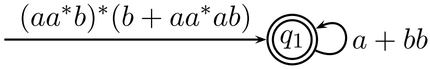
Επιλέγουμε να διαγράψουμε τη κατάσταση q_2, οπότε ενημερώνουμε τις μεταβάσεις των άλλων καταστάσεων:



Επιλέγουμε να διαγράψουμε τη κατάσταση q_3, οπότε ενημερώνουμε τα υπόλοιπα:



Επιλέγουμε να διαγράψουμε τη κατάσταση q_0:



Τελικά, η κανονική έκφραση που προκύπτει είναι:

$$(aa^*b)^*(b + aa^*ab)(a + bb)^*$$

7.5 Ελαχιστοποίηση DFA

Σε πολλές περιπτώσεις μπορούμε να μειώσουμε τον αριθμό καταστάσεων ενός DFA ως εξής: Δύο καταστάσεις μπορούν να συγχωνευτούν σε μία αν είναι και οι δύο τελικές ή και οι δύο μη τελικές και αν ξεκινώντας από οποιαδήποτε από αυτές το αυτόματο θα έχει το ίδιο αποτέλεσμα (ως προς την αποδοχή ή μη) για οποιαδήποτε συμβολοσειρά διαβαστεί στη συνέχεια. Για παράδειγμα, δύο καταστάσεις που απορρίπτουν και με κάθε σύμβολο πηγαίνουν στον εαυτό τους ('junk' states) μπορούν να συγχωνευτούν σε μία.

Πιο αυστηρά, οι καταστάσεις q_i, q_j μπορούν να συγχωνευτούν αν:

$$\forall w \in \Sigma^* : \delta(q_i, w) \in F \Leftrightarrow \delta(q_j, w) \in F$$

Προσέξτε ότι ο παραπάνω ορισμός περιλαμβάνει και την περίπτωση $w = \varepsilon$: αυτό ισοδυναμεί με την απαίτηση οι δύο καταστάσεις να είναι του ίδιου είδους ως προς την αποδοχή (δηλ. τελικές ή μη τελικές).

Παρατηρήστε ακόμη ότι δεν έχει καμμία σημασία με ποιες συμβολοσειρές μπορεί το αυτόματο να φτάσει στις δύο καταστάσεις. Αυτό που έχει σημασία είναι τι κάνει *από εκεί και μετά*.

Αποδεικνύεται (συνέπεια του Θεωρήματος Myhill-Nerode) ότι σε κάθε κανονικό σύνολο αντιστοιχεί ένα μοναδικό DFA (εκτός ισομορφισμού, δηλαδή εκτός μετονομασίας των καταστάσεων) με ελάχιστο αριθμό καταστάσεων και επιπλέον υπάρχει αλγόριθμος που κατασκευάζει το μοναδικό αυτό DFA. Ο αλγόριθμος αυτός βρίσκει συστηματικά όλα τα ζεύγη διακρίσιμων καταστάσεων όπως περιγράφεται παρακάτω:

Αλγόριθμος ελαχιστοποίησης DFA

1. Εξαλείφουμε όλες τις απρόσιτες καταστάσεις.

2. Φτιάχνουμε ένα πίνακα για να συγκρίνουμε κάθε ζεύγος καταστάσεων. Αρχικά σημειώνουμε X_0 σε όλα τα ζεύγη όπου η μία κατάσταση είναι τελική ενώ η άλλη δεν είναι.

3. Επαναλαμβάνουμε το παρακάτω για $i := 1, 2, \ldots$ μέχρι που σε κάποια επανάληψη να μην σημειωθεί κανένα νέο ζεύγος στον πίνακα:

   για κάθε ζεύγος q_k, q_j που δεν έχει ήδη σημειωθεί:

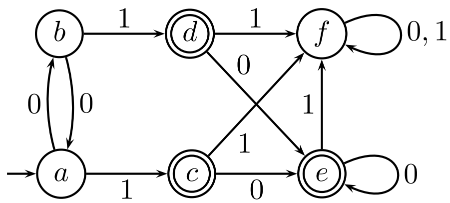
   για κάθε σύμβολο $\sigma \in \Sigma$:

   αν το ζεύγος $(\delta(q_k, \sigma), \delta(q_j, \sigma))$ είναι σημειωμένο με X_{i-1} τότε σημειώνουμε το ζεύγος (q_k, q_j) με X_i.

4. Ζεύγη που δεν έχουν σημειωθεί συγχωνεύονται.

Αποδεικνύεται (η απόδειξη παραλείπεται) ότι ο παραπάνω αλγόριθμος δεν οδηγεί σε αντιφάσεις, δηλαδή αν τα ζεύγη (q_k, q_j) και (q_k, q_m) δεν έχουν σημειωθεί (είναι συγχωνεύσιμα) τότε ούτε το ζεύγος (q_m, q_j) έχει σημειωθεί (επομένως και οι τρεις καταστάσεις συγχωνεύονται σε μία.

Παράδειγμα 7.15. Έστω το αυτόματο M που φαίνεται στο σχήμα, το οποίο αποδέχεται την γλώσσα $L = 0^*10^*$.

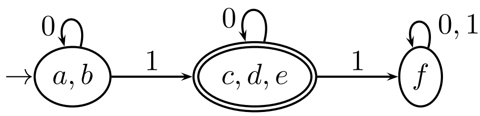


Στον πίνακα οι δείκτες του X δείχνουν σε ποια επανάληψη εγγράφουμε το (οι δείκτες υποδηλώνουν επίσης με πόσα σύμβολα διακρίνονται οι καταστάσεις).

b					
ⓒ	X_0	X_0			
ⓓ	X_0	X_0			
ⓔ	X_0	X_0			
f	X_1	X_1	X_0	X_0	X_0
	a	b	ⓒ	ⓓ	ⓔ

Τελικά οι ισοδύναμες καταστάσεις είναι $a \equiv b$, $c \equiv d \equiv e$.

Το ελάχιστο αυτόματο φαίνεται στο παρακάτω σχήμα.



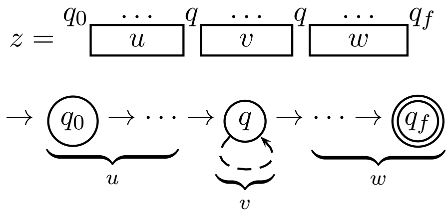
Επεξηγήσεις για τον παραπάνω αλγόριθμο. Ο παραπάνω αλγόριθμος βασίζεται στην εξής παρατήρηση: αν δύο καταστάσεις q_i, q_j διακρίνονται με k σύμβολα, δηλαδή υπάρχει συμβολοσειρά w μήκους k που τις οδηγεί σε διαφορετικό αποτέλεσμα, τότε αν δύο καταστάσεις έστω q_m, q_n οδηγούν με κάποιο σύμβολο στις q_i, q_j, τότε οι q_m, q_n διακρίνονται με $k + 1$ σύμβολα. Αυτό ισχύει και αντίστροφα, και επομένως αρκεί ο αλγόριθμος να εξετάζει ένα σύμβολο κάθε φορά: αν υπάρχουν δύο καταστάσεις που διακρίνονται με $k + 1$ σύμβολα, τότε υπάρχει ένα σύμβολο που τις οδηγεί σε k-διακρίσιμες καταστάσεις.

7.6 Pumping Lemma για κανονικά σύνολα

Ένα σημαντικό αποτέλεσμα είναι το Pumping Lemma που χρησιμοποιείται κυρίως για να αποδεικνύουμε ότι συγκεκριμένες γλώσσες δεν είναι κανονικές αλλά και σε αλγορίθμους για να απαντάμε ερωτήσεις σχετικά με πεπερασμένα αυτόματα, όπως αν μια γλώσσα που γίνεται αποδεκτή από ένα πεπερασμένο αυτόματο είναι πεπερασμένη ή άπειρη.

Pumping Lemma. Αν μία γλώσσα είναι κανονική τότε την αποδέχεται ένα DFA $M = \{Q, \Sigma, \delta, q_0, F\}$ με κάποιο συγκεκριμένο αριθμό από καταστάσεις, έστω n, δηλαδή $|Q| = n$. Έστω μία λέξη z που γίνεται αποδεκτή από το αυτόματο M και η οποία έχει μήκος μεγαλύτερο από n.

Καθώς επεξεργαζόμαστε το z, το αυτόματο μας M πρέπει να περάσει ξανά από μία κατάσταση, γιατί δεν υπάρχουν περισσότερες από n καταστάσεις (αρχή του περιστερώνα, pigeonhole principle). Έχουμε ότι ένα μονοπάτι που αποδέχεται το z είναι το ακόλουθο:



Το uvw γίνεται αποδεκτό, όπως επίσης και το uw, ή το $uvvw$ ή γενικά το uv^iw για οποιοδήποτε $i \in \mathbb{N}$. Δηλαδή το v μπορεί να επαναληφθεί όσες φορές θέλουμε.

Λήμμα 7.16 (Pumping Lemma)**.** *Εάν L είναι regular τότε:*
$\exists n \in \mathbb{N}, \forall z \in L$ *με* $|z| \geq n, \exists u, v, w \in \Sigma^*$:
$[z = uvw \ \wedge |uv| \leq n \ \wedge |v| \geq 1 \wedge \forall i \in \mathbb{N}\ (uv^iw \in L)]$

Χρησιμοποιώντας το λήμμα δείχνουμε ότι ένα δοσμένο σύνολο δεν είναι regular. Η μέθοδος είναι η ακόλουθη:

1. Διαλέγεις τη γλώσσα που θέλεις να αποδείξεις πως δεν είναι regular.

2. Ο αντίπαλος (PL) επιλέγει ένα n. Θα πρέπει να μπορείς για οποιοδήποτε πεπερασμένο ακέραιο n διαλέξει, να αποδείξεις ότι η L δεν είναι regular, αλλά από τη στιγμή που ο αντίπαλος έχει διαλέξει ένα n αυτό είναι σταθερό στην απόδειξη.

3. Διαλέγεις ένα string z της L έτσι ώστε $\mid z \mid \geq n$.

4. Ο αντίπαλος (PL) σπάει το z σε u, v και w που ικανοποιούν τους περιορισμούς $|uv| \leq n$ και $|v| \geq 1$.

5. Φτάνεις σε αντίφαση δείχνοντας ότι για κάθε u, v, w που καθορίζονται από τον αντίπαλο, υπάρχει ένα i για το οποίο $uv^i w$ δεν ανήκει στην L. Τότε μπορούμε να συμπεράνουμε ότι η L δεν είναι regular. Η επιλογή του i μπορεί να εξαρτάται από τα n, u, v, w.

Παράδειγμα 7.17. $L = \{a^k b^k \mid k \in \mathbb{N}\} = \{\varepsilon, ab, aabb, aaabbb, \ldots\}$ δεν είναι regular.

1. Υποθέτουμε ότι L είναι regular και χρησιμοποιούμε το Pumping lemma.

2. PL: $\exists n \in \mathbb{N}$

3. Διαλέγουμε $z = a^n b^n$. Εντάξει επιλογή, διότι $z \in L$, $|z| = 2n \geq n$.

4. PL: z μπορεί να γραφεί: $z = uvw$ με $|uv| \leq n \wedge |v| \geq 1$, οπότε αναγκαστικά $v = a^l$ με $l \geq 1$.

5. Διαλέγουμε $i = 2$: $uvvw = a^{n+l} b^n \in L$.

’Ατοπο

Εναλλακτικά:

1. Υποθέτουμε πάλι ότι η L είναι regular.

2. Συνεπώς, αναγνωρίζεται από ένα DFA M. Αυτό θα έχει κάποιον πεπερασμένο αριθμό καταστάσεων, έστω n.

3. Θέτουμε $z = a^n b^n \in L$.

4. Το DFA M αποδέχεται τη λέξη z. Καθώς τη διατρέχει, μέχρι να φτάσει στο $n-$οστό a, αναγκαστικά τουλάχιστον μία κατάστασή του επαναλαμβάνεται, έστω στο $i-$οστό και στο $j-$οστό a, $i < j$.

5. Έστω $l = j - i$. Τότε, για κάθε $k \geq -1$, το DFA M θα αποδέχεται το string $a^{n+k \cdot l} b^n$. Το substring $a^{k \cdot l}$ θα αντιστοιχεί σε k επαναλήψεις της ίδιας κυκλικής ακολουθίας καταστάσεων του M.

’Ατοπο

7.7 Ασκήσεις

1. Αποδείξτε ή ανταποδείξτε (δίνοντας αντιπαράδειγμα) ότι για δύο γλώσσες $L_1, L_2 \subseteq \Sigma^*$ ισχύει $(L_1 \cup L_2)^* = L_1^* \cup L_2^*$

2. Δώστε διάγραμμα καταστάσεων για DFA που αποδέχεται την γλώσσα

$$L = \left\{ w \in \{a,b\}^* \mid \text{κάθε } a \left\langle \begin{array}{l} \text{ακολουθείται αμέσως από ένα } b \\ \text{και έπεται αμέσως ενός } b \end{array} \right. \right\}$$

Για παράδειγμα, $bababb \in L$, $bbb \in L$, αλλά $ab \notin L$.
Το DFA σας δεν θα πρέπει να έχει πάνω από τέσσερις καταστάσεις.

3. Κατασκευάστε κανονικές εκφράσεις για κάθε μία από τις γλώσσες:
(α) Σύνολο συμβολοσειρών του $\{a,b\}^*$ των οποίων το πλήθος των a δεν είναι πολλαπλάσιο του 3.
(β) Σύνολο συμβολοσειρών του $\{a,b\}^*$ που έχουν άρτιο πλήθος a και τελειώνουν σε b.
(γ) Σύνολο συμβολοσειρών του $\{a,b\}^*$ που δεν έχουν δύο συνεχόμενα a.

4. Δύο κανονικές εκφράσεις ονομάζονται ισοδύναμες όταν παράγουν την ίδια γλώσσα. Χρησιμοποιούμε το σύμβολο της ισότητας για να δείξουμε την ισοδυναμία. Έστω r και s κανονικές εκφράσεις. Θεωρήστε την 'εξίσωση' $X = rX + s$. Με την προϋπόθεση ότι η γλώσσα που παράγεται από την r, δηλαδή η $L(r)$, δεν περιέχει την κενή συμβολοσειρά ε, βρείτε την λύση για το X και αποδείξτε ότι είναι μοναδική (εκτός ισοδυναμίας). Ποια είναι η λύση αν η $L(r)$ περιέχει την ε; *Υπόδειξη*: Στην τελευταία περίπτωση η λύση δεν είναι μοναδική. Για παράδειγμα, αμφότερες οι $X = 10^*$ και $X = (1+11)0^*$, που αναπαριστούν διαφορετικές γλώσσες, είναι λύσεις της $X = 1 + X0^*$.

5. Δείξτε ότι $\delta(q, xy) = \delta(\delta(q, x), y)$ για κάθε x, y.

6. Δώστε DFA που αποδέχονται τις ακόλουθες γλώσσες επί του αλφαβήτου $\{0,1\}$:

   - $\{w \mid$ κάθε υποακολουθία από 5 διαδοχικά σύμβολα περιέχει τουλάχιστον δύο «0»$\}$
   - $\{w \mid w$: δυαδική αναπαράσταση φυσικού αριθμού $\overline{w}$, w αρχίζει με 1, $\overline{w} \equiv 0 \pmod 5$ $\}$
   - $\{w \mid$ το 3ο σύμβολο πριν το τελευταίο της συμβολοσειράς είναι «1»$\}$

- $\{w \mid |w|$ διαιρείται από το 2 ή το 3 ή από αμφότερα$\}$
- $\{w \in \{1,2,3\}^* \mid$ άθροισμα ψηφίων της w είναι $\equiv 0 \pmod 4)\}$

7. Δώστε NFA για την $\{w \in \{0,1\}^* \mid$ υπάρχουν δύο «0» που χωρίζονται από μία συμβολοσειρά μήκους $4i$ για κάποιο $i \geq 0$ $\}$.

8. Κατασκευάστε DFA ισοδύναμα με τα NFA:

α) $(\{p,q,r,s\},\{0,1\},\delta_1,p,\{s\})$

δ_1	0	1
p	p,q	p
q	r	r
r	s	$-$
s	s	s

β) $(\{p,q,r,s\},\{0,1\},\delta_2,p,\{q,s\})$

δ_2	0	1
p	q,s	q
q	r	q,r
r	s	p
s	$-$	p

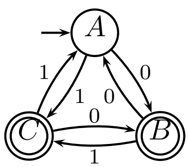
9. Κατασκευάστε FA ισοδύναμα με:

α) $10 + (0 + 11)0^*1$

β) $01[((10)^* + 111)^* + 0]^*1$

γ) $((0+1)(0+1))^* + ((0+1)(0+1)(0+1))^*$

10. Κατασκευάστε κανονική έκφραση για το:



11. Γράψτε κανονικές εκφράσεις για τις παρακάτω γλώσσες επί του $\{0,1\}$. Δικαιολογήστε την ορθότητα των κανονικών σας εκφράσεων.

α) Το σύνολο των strings που δεν περιέχουν το 101 ως substring.

β) Το σύνολο των strings με ίσο πλήθος εμφανίσεων «0» και «1», έτσι ώστε δεν υπάρχει πρόθεμα που να περιέχει δύο περισσότερα «0» από ότι «1», ούτε δύο περισσότερα «1» από ότι «0».

12. Δείξτε τις ακόλουθες ταυτότητες για κανονικές εκφράσεις. Εδώ $r = s$ σημαίνει $L(r) = L(s)$.

- $r + s = s + r$
- $(r+s)+t = r+(s+t)$
- $(rs)t = r(st)$
- $r(s+t) = rs + rt$

- $(r+s)t = rt + st$
- $\emptyset^* = \varepsilon$,
- $(r^*)^* = r^*$
- $(\varepsilon + r)^* = r^*$
- $(r^*s^*)^* = (r+s)^*$

13. Αποδείξτε ή ανταποδείξτε:

    - $(rs+r)^*r = r(sr+r)^*$,
    - $s(rs+s)^*r = rr^*s(rr^*s)^*$,
    - $(r+s)^* = r^* + s^*$.

14. Ποια από τα παρακάτω σύνολα είναι κανονικά; Αποδείξτε!

    - $\{a^{3k} \mid k \in \mathbb{N}\}$
    - σύνολο εξισορροπημένων παρενθέσεων
    - $\{a^i b^j \mid 1 \le i < j\}$
    - $\{w \mid$ η w περιέχει λιγότερα a από $b\}$
    - $\{a^i b^j a^j \mid i, j \ge 1\}$
    - $\{w \in \{1\}^* \mid |w|$ είναι πρώτος αριθμός$\}$
    - $\{uvw \in \{a,b\}^* \mid$ η uv είναι παλίνδρομη αρτίου μήκους$\}$
    - $\{w \in \{a,b\}^* \mid$ η w δεν έχει 4 συνεχόμενα $b\}$
    - $\{uvw \in \{a,b\}^* \mid$ η uw είναι παλίνδρομη αρτίου μήκους$\}$
    - $\{w \in \{a,b\}^* \mid$ η w είναι παλίνδρομη$\}$

15. Έστω κανονικό σύνολο L. Ποια από τα παρακάτω σύνολα είναι κανονικά; Δικαιολογήστε τις απαντήσεις σας.

    - $\{a_1 a_3 a_5 \ldots a_{2n-1} \mid a_1 a_2 a_3 \ldots a_{2n} \in L\}$
    - $\{a_2 a_1 a_4 a_3 \ldots a_{2n} a_{2n-1} \mid a_1 a_2 a_3 \ldots a_{2n} \in L\}$
    - Cycle$(L) = \{w_1 w_2 \mid w_1 w_2 \in L,\ w_1, w_2$: strings$\}$
    - max$(L) = \{x \in L \mid$ δεν υπάρχει $y \neq \varepsilon$ με $xy \in L\}$
    - min$(L) = \{x \in L \mid$ δεν υπάρχει πρόθεμα του x που να είναι στην $L\}$
    - init$(L) = \{x \in L \mid$ για κάποιο y, $xy \in L\}$
    - $L^R = \{x \mid x^R \in L\}$
    - $\{x \mid xx^R \in L\}$