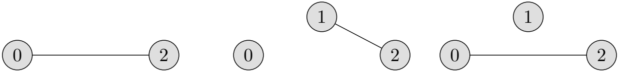
Προβλήματα απόφασης

- Μας απασχολούν κυρίως προβλήματα απόφασης: «Σε γράφο με τρείς κόμβους υπάρχει μονοπάτι από τον κόμβο 0 στον κόμβο 2;»
- Τα στιγμιότυπα που μπορεί να έχουμε είναι τριών ειδών:

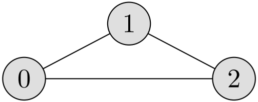
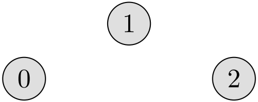
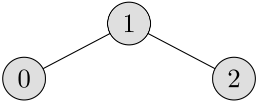
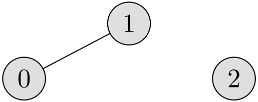
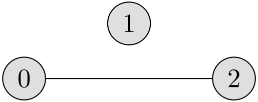
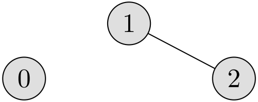


άλλου προβλήματος όχι ναι

- Κάθε πρόβλημα απόφασης Π κωδικοποιείται με φυσικό τρόπο από κάποια τυπική γλώσσα, κάτω από ένα σχήμα κωδικοποίησης.
- Μπορούμε να χρησιμοποιήσουμε το αλφάβητο $\Sigma = \{0, 1\}$ και τον «εναδικό» συμβολισμό σαν σχήμα κωδικοποίησης.

Κωδικοποίηση προβλήματος

Σε γράφο με τρείς κόμβους υπάρχει μονοπάτι από τον κόμβο 0 στον κόμβο 2;

ΝΑΙ	ΟΧΙ
 1011011100110101110011101011	 1001100111
 1011001101011100111011	 101100110100111
 1011100110011101	 10011011100111011

Προβλήματα απόφασης και γλώσσες

- Το πρόβλημα Π και το σχήμα κωδικοποίησης χωρίζει το Σ^* σε τρία ξένα μεταξύ τους υποσύνολα:

$$
\begin{aligned}
&L_{no} = \{x \in \Sigma^* : x \text{ είναι ΟΧΙ στιγμιότυπο του } \Pi\}\\
&L_{yes} = \{x \in \Sigma^* : x \text{ είναι ΝΑΙ στιγμιότυπο του } \Pi\}\\
&\Sigma^* \setminus (L_{no} \cup L_{yes}) = \{x \in \Sigma^* : x \text{ δεν είναι στιγμιότυπο του } \Pi\}
\end{aligned}
$$

- Το πρόβλημα απόφασης μετασχηματίζεται στην ερώτηση $w \in L_{yes}$; (w είναι η κωδικοποίηση του στιγμιοτύπου του προβλήματος)
- Οι γλώσσες λοιπόν περιγράφουν προβλήματα.
- Θέλουμε ένα τυπικό τρόπο περιγραφής των γλωσσών και μια τυπική περιγραφή μιας ιδεατής μηχανής που «αναγνωρίζει» γλώσσες (λύνει προβλήματα απόφασης).

Μηχανές πεπερασμένων καταστάσεων

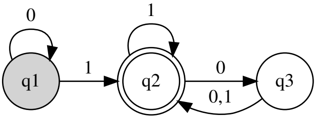
Finite State Machines (FSM)

- Τυπικό μοντέλο υπολογιστή με πεπερασμένη μνήμη (καταστάσεις).
- Η μνήμη της μηχανής είναι το σύνολο των καταστάσεών της.
- Η μηχανή διαβάζει σύμβολα από την ταινία εισόδου (input) και κάνει προκαθορισμένες μεταβάσεις.
- Η είσοδος «καταναλώνεται» από τα αριστερά προς τα δεξιά.
- Η επεξεργασία σταματά όταν καταναλωθούν όλα τα σύμβολα.
- Η μηχανή αποδέχεται ή απορρίπτει την είσοδο ανάλογα με την κατάσταση που βρίσκεται στο τέλος της επεξεργασίας.

Λέμε ότι η μηχανή αναγνωρίζει τη γλώσσα των συμβολοσειρών που αποδέχεται.

Ντετερμινιστικό πεπερασμένο αυτόματο

Deterministic Finite Automaton (DFA)



- Έχει τρεις καταστάσεις με ονόματα: $q1, q2, q3$.
- Η αρχική κατάσταση σημειώνεται με σκούρο χρώμα (για τεχνικούς λόγους, συνήθως σημειώνεται με βέλος από το πουθενά).
- Η τελική κατάσταση σημειώνεται με διπλό κύκλο.
- Οι μεταβάσεις σημειώνονται με βέλη ανάμεσα στις καταστάσεις.

state/input	0	1
$q1$	$q1$	$q2$
$q2$	$q3$	$q2$
$q3$	$q2$	$q2$

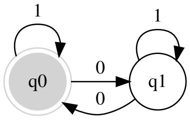
Η διαδικασία της επεξεργασίας

- Η επεξεργασία ξεκινά με το αυτόματο στην αρχική κατάσταση.
- Η είσοδος είναι μια συμβολοσειρά που «καταναλώνεται» από αριστερά προς τα δεξιά.
- Η λήψη ενός συμβόλου προκαλεί μετάβαση σε νέα κατάσταση, εκεί που δείχνει το βέλος με την αντίστοιχη ετικέτα.
- Μετά τη λήψη του τελευταίου συμβόλου, το αυτόματο δέχεται την είσοδο ανν βρίσκεται στην τελική κατάσταση, αλλιώς η είσοδος απορρίπτεται από το αυτόματο.

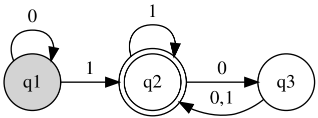
Παραδείγματα επεξεργασίας

Είσοδος: 1101

$q0$	1101	$q0$	101
$q0$	101	$q0$	01
$q0$	01	$q1$	1
$q1$	1	$q1$	ϵ
Απορρίπτει την 1101			



$q1$	1101	$q2$	101
$q2$	101	$q2$	01
$q2$	01	$q3$	1
$q3$	1	$q2$	ϵ
Δέχεται την 1101			



Ένα πεπερασμένο αυτόματο M είναι μια πεντάδα $M(K, \Sigma, \delta, s, F)$ όπου:

- K είναι ένα πεπερασμένο σύνολο καταστάσεων.
- Σ είναι ένα πεπερασμένο αλφάβητο.
- $s \in K$ είναι η αρχική κατάσταση.
- $F \subseteq K$ είναι το σύνολο των τελικών καταστάσεων.
- $\delta : K \times \Sigma \to K$ είναι η συνάρτηση μετάβασης.

Η συνάρτηση μετάβασης δ καθορίζει ουσιαστικά τη λειτουργία του αυτομάτου. Αν q είναι η κατάστασή του και σ το επόμενο σύμβολο της εισόδου, τότε το αυτόματο μεταβαίνει στην κατάσταση $\delta(q, \sigma)$.

Συμβολοσειρές και πεπερασμένα αυτόματα

DFA $M(K, \Sigma, \delta, s, F)$ δέχεται την συμβολοσειρά w:

Το M ξεκινά από την κατάσταση s, «καταναλώνει» την είσοδο και καταλήγει σε κάποια κατάσταση του F.

Παράγει σε ένα βήμα (συνάρτηση $\vdash_M$):

$(q, w) \vdash_M (q', w')$ ανν

- $w = \sigma w'$, για κάποιο $\sigma \in \Sigma$
- $q' = \delta(q, \sigma)$

Έχουμε δηλαδή τη συνάρτηση $\vdash_M$: $K \times \Sigma^+ \to K \times \Sigma^*$. Τα στοιχεία του $K \times \Sigma^*$ ονομάζονται συνολικές καταστάσεις.

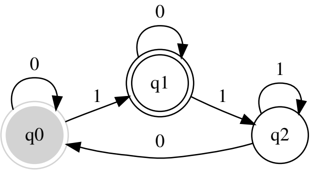
Σχέση $\vdash_M^*$

Παράγει σε μηδέν ή περισσότερα βήματα:

$(q, w) \vdash_M^* (q', w')$ ανν υπάρχουν $q1, \ldots, q_n \in K$ και $\sigma_1, \ldots, \sigma_n \in \Sigma$:

- $w = \sigma_1\sigma_2 \ldots \sigma_n w'$
- $(q, \sigma_1\sigma_2 \cdots \sigma_n w') \vdash_M (q_1, \sigma_2 \ldots \sigma_n w') \vdash_M \cdots \vdash_M (q_n, w')$
- $q_n = q'$

Η $\vdash_M^*$ είναι σχέση και όχι συνάρτηση



$$(q0, 0110) \vdash_M^* (q0, 0110)$$
$$(q0, 0110) \vdash_M^* (q2, 0)$$
$$(q0, 0110) \vdash_M^* (q0, \epsilon)$$

Γλώσσες και πεπερασμένα αυτόματα

DFA M δέχεται τη συμβολοσειρά w:

Η συμβολοσειρά w γίνεται αποδεκτή από το ΠΑ M, ανν

$$(s, w) \vdash_M^* (q, \epsilon), \text{ για κάποιο } q \in F$$

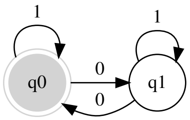
Γλώσσα ενός DFA $M(K, \Sigma, \delta, s, F)$:

Είναι το σύνολο των συμβολοσειρών που γίνονται αποδεκτές από το M:

$$L(M) = \{w : w \text{ είναι αποδεκτό από το } M\}$$

Γλώσσα ενός πεπερασμένου αυτομάτου

Παράδειγμα



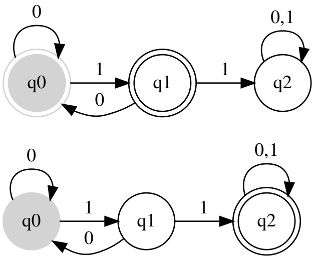
$$L(M) = \{w : w \text{ περιέχει άρτιο αριθμό από } 0\}$$

Με μαθηματική επαγωγή στο μήκος του w. Η επαγωγική υπόθεση είναι:

- $(q0, w) \vdash_M^* (q0, \epsilon)$ ανν το w περιέχει άρτιο αριθμό από 0,
- $(q0, w) \vdash_M^* (q1, \epsilon)$ ανν το w περιέχει περιττό αριθμό από 0.

Γιατί είναι απαραίτητη η δεύτερη υπόθεση;

DFA που δέχονται συμπληρωματικές γλώσσες



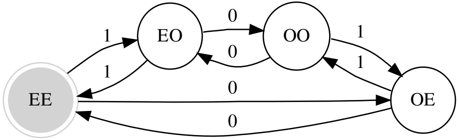
$L = \{w : w$ δεν περιέχει $11\}$

$L = \{w : w$ περιέχει $11\}$

Σχεδόν τα ίδια αυτόματα: οι τελικές καταστάσεις του ενός είναι οι μη τελικές καταστάσεις του άλλου.

Παράδειγμα

DFA για $L = \{w : w$ περιέχει ζυγό αριθμό 0 και ζυγό αριθμό 1$\}$:

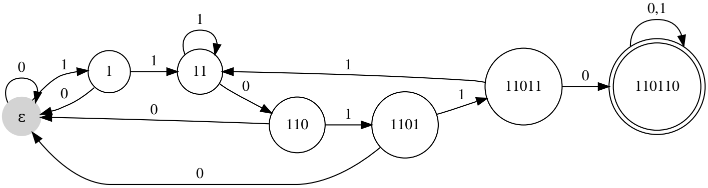


(EE:ζυγός # 0, ζυγός # 1, EO: ζυγός # 0, περιττός # 1, κτλ)

- Γιατί αποδέχεται τις συμβολοσειρές EE;
- Γιατί απορρίπτει τις συμβολοσειρές EO, OE, OO;

Παράδειγμα

DFA για $L = \{w : w \text{ περιέχει } 110110\}$:



DFA για $L = \{w : w \text{ δεν περιέχει } 110110\}$: το ίδιο αλλάζοντας τις καταστάσεις (μη τελικές $\longrightarrow$ τελικές και τελικές $\longrightarrow$ μη τελικές).

Κλειστότητα ως προς το συμπλήρωμα

- Τα DFA αποδέχονται (αναγνωρίζουν) τις κανονικές γλώσσες (επόμενη διάλεξη).
- Υπάρχει αλγόριθμος που μετατρέπει κάθε DFA που αποδέχεται μια L σε ένα DFA που αποδέχεται την $\overline{L}$ (απλά άλλαξε τις μη τελικές καταστάσεις σε τελικές και τις τελικές σε μη τελικές).
- Άρα αν η L είναι κανονική τότε είναι κανονική και η $\overline{L}$.
- Λέμε ότι η κλάση των κανονικών γλωσσών είναι κλειστή ώς προς το συμπλήρωμα.

Μη ντετερμινιστικά πεπερασμένα αυτόματα

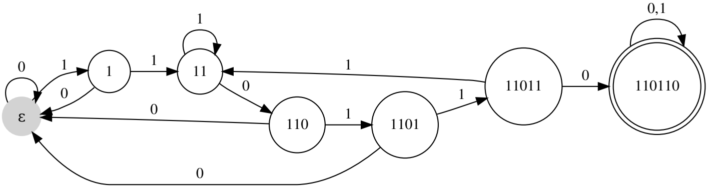
Non-deterministic finite automata (NFA)

- Γενίκευση των ντετερμινιστικών πεπερασμένων αυτομάτων.
- Αντί συνάρτηση μετάβασης δ έχουν σχέση μετάβασης Δ που επιτρέπει $0, 1, 2, \ldots$ επόμενες καταστάσεις.
- Καθώς το NFA υπολογίζει υπάρχουν πολλές διαφορετικές «ενσαρκώσεις» του.
- Γιατί τα μελετάμε;
  - Διευκολύνουν τις αποδείξεις.
  - Πιο φυσικός και σύντομος τρόπος για να περιγραφούν κάποιες γλώσσες.

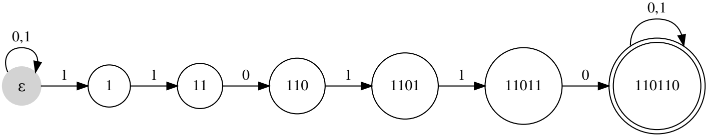
Δύο αυτόματα για την ίδια γλώσσα

$L = \{w : w \text{ περιέχει } 110110\}$

Ντετερμινιστικό:

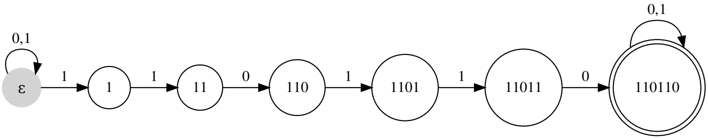


Μη ντετερμινιστικό:



Μη ντετερμινιστικό αυτόματο

$L = \{w : w \text{ περιέχει } 110110\}$



- Το αυτόματο αρχίζει να καταναλώνει την συμβολοσειρά εισόδου.
- Κάποια στιγμή «μαντεύει» ότι ήρθε η ώρα να διαβάσει 110110.
- Επαληθεύει την μαντεψιά.

Παρατήρηση: Υπάρχει μία κατάσταση (γενικά μπορεί να υπάρχουν πολλές) που τα εξερχόμενα βέλη είναι περισσότερα από τα σύμβολα του αλφαβήτου.

Μη ντετερμινιστικά πεπερασμένα αυτόματα

Τυπικός ορισμός

Ένα μη ντετερμινιστικό πεπεραμένο αυτόματο M είναι μια πεντάδα $M(K, \Sigma, \Delta, s, F)$ όπου:

- K είναι ένα πεπερασμένο σύνολο καταστάσεων.
- Σ είναι ένα πεπερασμένο αλφάβητο.
- $s \in K$ είναι η αρχική κατάσταση.
- $F \subseteq K$ είναι το σύνολο των τελικών καταστάσεων.
- $\Delta : K \times \Sigma \cup \{\epsilon\} \times K$ είναι η σχέση μετάβασης.

Μη ντετερμινιστικά πεπεραμένα αυτόματα

Επέκταση των ορισμών από τα ντετερμινιστικά πεπεραμένα αυτόματα

Σχέση $\vdash_M$ (παράγει σε ένα βήμα):

$(q, w) \vdash_M (q', w')$ ανν υπάρχει $u \in \Sigma^*$ τέτοιο ώστε:

- $w = uw'$,
- $(q, u, q') \in \Delta$.

Ομοίως επεκτείνεται και η σχέση $\vdash_M^*$ (ακολουθία μεταβάσεων).

NFA δέχεται το w:

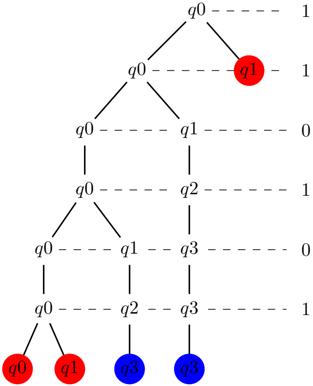
Ανν υπάρχει κάποια ακολουθία μεταβάσεων που καταλήγει σε τελική κατάσταση: $\exists q \in F : (s, w) \vdash_M^* (q, \epsilon)$

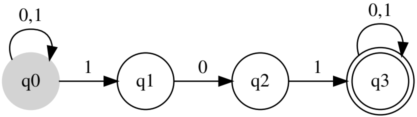
Γλώσσα που δέχεται το NFA M:

$L(M) = \{w : M \text{ δέχεται το } w\}$

Μη ντετερμινιστικός υπολογισμός

Είσοδος 110101





Το δέντρο των επιλογών. Η ρίζα είναι η αρχή των υπολογισμών και κάθε κλαδί δημιουργείται όταν το NFA έχει πολλές επιλογές. Το NFA δέχεται την είσοδο ανν τουλάχιστο ένα φύλλο είναι τελική κατάσταση.

Μη ντετερμινιστικά αυτόματα

Παρατηρήσεις

- Οι παρακάτω εκφράσεις είναι ισοδύναμες:
  - Το NFA κάποια στιγμή «μαντεύει» ότι διαβάζει ένα στοιχείο της L και στη συνέχεια επαληθεύει τη μαντεψιά.
  - Το NFA εκτελεί παράλληλα όλες τις πιθανές μεταβάσεις και ελέγχει αν με κάποιο από όλους τους τρόπους μεταβάσεων γίνεται αποδεκτή η είσοδος.
- Ένα NFA αποδέχεται μια L αν υπάρχει τρόπος να αποδεχτεί όλα τα στοιχεία της L και μόνο αυτά.
- Προσοχή: Αν ένα NFA αποδέχεται τα στοιχεία μιας L, τότε ίσως να υπάρχουν πολλοί τρόποι (ακολουθίες μεταβάσεων) για να μην γίνει αποδεκτό ένα $w \in L$. Όμως κάποια ακολουθία μεταβάσεων θα κάνει το w αποδεκτό.

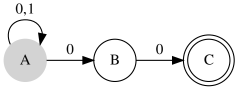
Ισοδυναμία NFA με DFA

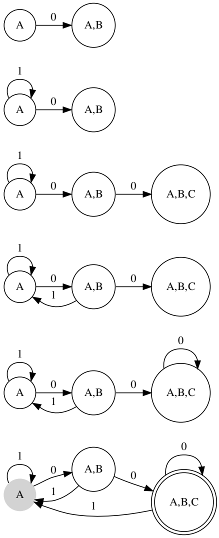
Υπάρχει ένας φυσικός τρόπος να μετατραπεί το NFA σε DFA:

- Θεωρούμε ότι ένα DFA «προσομοιώνει» τη λειτουργία του NFA.
- Το DFA καταγράφει όλες τις πιθανές μεταβάσεις του NFA.
- Οι καταστάσεις του DFA είναι υποσύνολα του συνόλου των καταστάσεων του NFA.
- Τελικές καταστάσεις του DFA είναι όλες εκείνες που περιέχουν κάποια τελική κατάσταση του NFA.
- Το DFA αγνοεί τις μεταβάσεις του NFA που είναι απροσδιόριστες.

Κατασκευή ισοδύναμου DFA

NFA για $L = \{w : w \text{ τελειώνει με } 00\}$





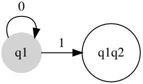
Κατασκευή ισοδύναμου DFA

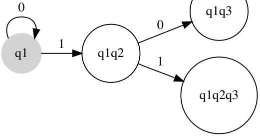
Παρατηρήσεις

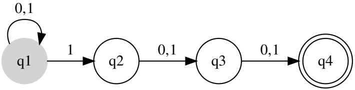
- Οι καταστάσεις του DFA είναι στη χειρότερη περίπτωση όλα τα υποσύνολα των καταστάσεων του NFA.
- Στη χειρότερη περίπτωση οι καταστάσεις του DFA είναι 2^k, όπου k ο αριθμός των καταστάσεων του NFA (εκθετική αύξηση του αριθμού των καταστάσεων).
- Τελικά το NFA είναι «βολικότερο» και όχι ισχυρότερο μοντέλο υπολογισμού.

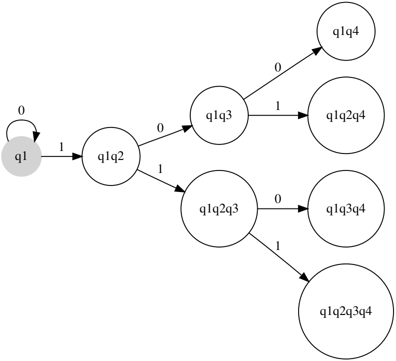
Κατασκευή ισοδύναμου DFA

$L = \{w : w$ στην τρίτη θέση από το τέλος έχει $1\}$



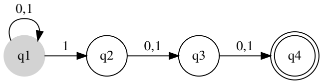


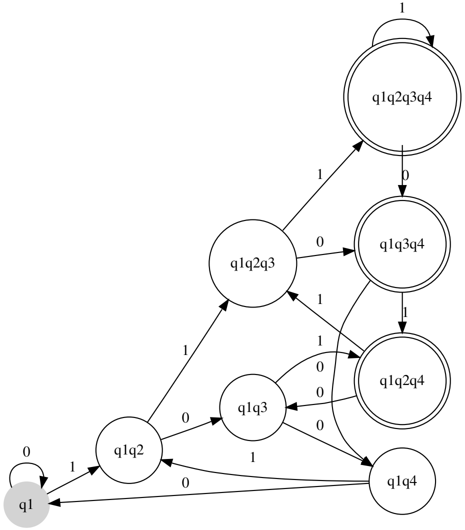




Κατασκευή ισοδύναμου DFA

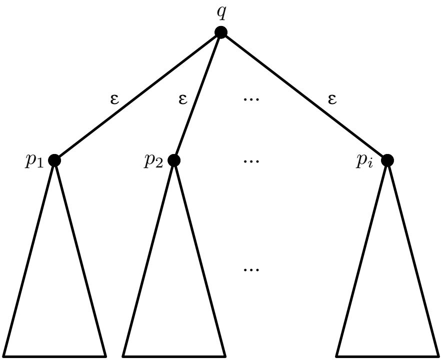
$L = \{w : w$ στην τρίτη θέση από το τέλος έχει 1$\}$





NFA με ϵ-κινήσεις

Από τον ορισμό της σχέσης μετάβασης $\Delta : K \times \Sigma \cup \{\epsilon\} \times K$, υπάρχουν ϵ-κινήσεις για τα NFA: μπορούν να αλλάζουν καταστάσεις χωρίς να διαβάζουν την είσοδο.

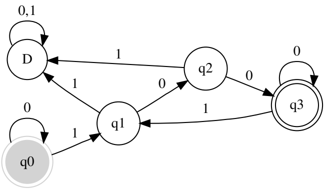


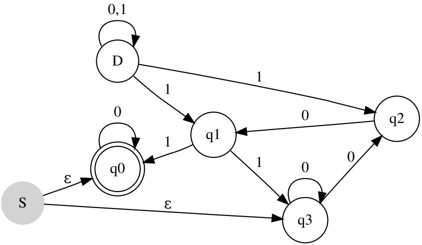
Το NFA, χωρίς να διαβάσει την είσοδο, «μαντεύει» ότι πρέπει ακολουθήσει κάποια μετάβαση $q \longrightarrow p_j$. Στη συνέχεια επαληθεύει τη μαντεψιά.

NFA για την αντίστροφη μιας L

$L = \{w : w$ κάθε 1 το ακολουθεί $(00)^+\}$

- Κατασκευή NFA για $L = \{w : w$ από κάθε 1 προηγείται $(00)^+\}$
- Χρησιμοποιούμε ϵ-κινήσεις για να ξεκινήσει το NFA από τις καταστάσεις που το DFA μπορεί να τελείωσε.
- Αντιστρέφουμε όλα τα βέλη του DFA.
- Τελική κατάσταση του NFA είναι η αρχική κατάσταση του DFA.
- Εξαφανίζονται οι καταβόθρες του DFA!

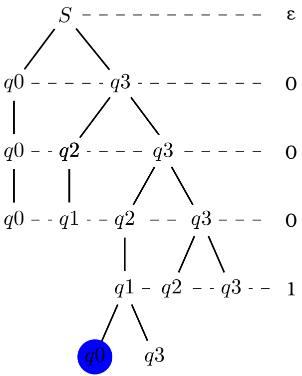


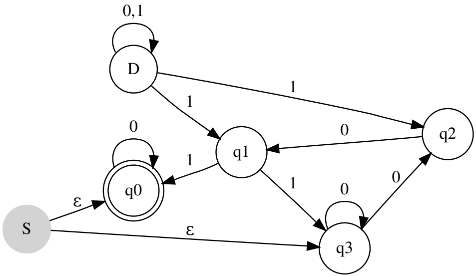


Παράδειγμα ϵ-κίνησης

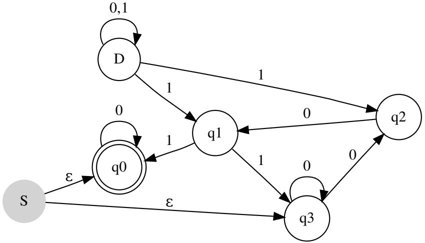
NFA για $L = \{w : w$ από κάθε 1 προηγείται $(00)^+\}$

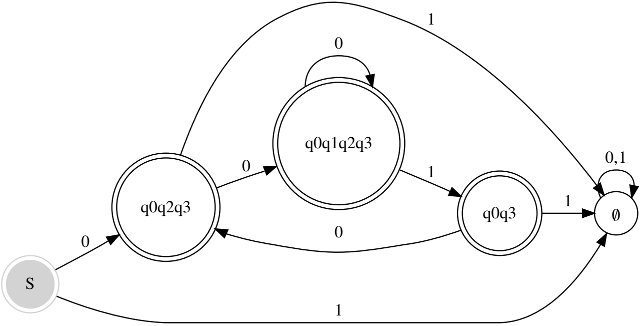
Στην είσοδο 0001 το NFA «μαντεύει» ότι πρέπει να ακολουθήσει το μονοπάτι που ξεκινά από την κατάσταση $q3$.





Κατασκευή ισοδύναμου DFA





Κλειστότητα ως προς την αντιστροφή

- Υπάρχει αλγόριθμος μετατροπής ενός DFA που αποδέχεται μια L σε NFA που αποδέχεται την L^R.
- Το NFA που δέχεται την L^R έχει ένα ισοδύναμο DFA.
- Άρα αν η L είναι κανονική τότε είναι κανονική και η L^R.
- Λέμε ότι η κλάση των κανονικών γλωσσών είναι κλειστή ως προς την αντιστροφή.

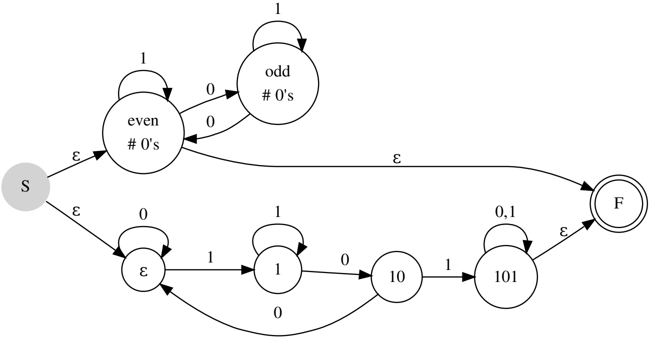
Ελαχιστοποίηση καταστάσεων DFA (μέθοδος Beckman):

DFA για μια $L \Rightarrow$ κατασκευή NFA για την $L^R \Rightarrow$ κατασκευή ισοδύναμου DFA για την $L^R \Rightarrow$ κατασκευή NFA για την $(L^R)^R \Rightarrow$ κατασκευή ισοδύναμου DFA για την $(L^R)^R \Rightarrow$ ελάχιστο DFA για την L (εκθετικό κόστος).

Υπάρχει πολυωνυμικός αλγόριθμος για την ελαχιστοποίηση των DFA.

Κλειστότητες κανονικών γλωσσών

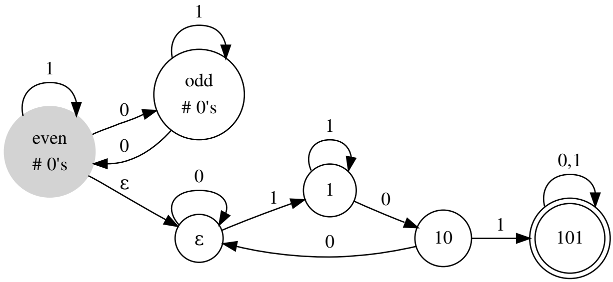
- Τα NFA με τις ϵ-κινήσεις μαζί με τον αλγόριθμο μετατροπής NFA σε DFA είναι ισχυρότατο εργαλείο για αποδείξεις κλειστότητας.
- Κλειστότητα ως προς την ένωση: ϵ-κινήσεις απο την αρχική του NFA προς τις αρχικές καταστάσεις των DFA και ϵ-κινήσεις απο τις τελικές των DFA προς την τελική του NFA. Μετά NFA$\Rightarrow$DFA.
- $L_1 = \{w : w \text{ περιέχει ζυγό \# } 0\}$, $L_2 = \{w : w \text{ περιέχει } 101\}$. NFA για $L_1 \cup L_2$:



Κλειστότητες κανονικών γλωσσών

Κλειστότητα ως προς την παράθεση: ϵ-κινήσεις από τις τελικές του «prefix» DFA προς την αρχική κατάσταση του «suffix» DFA. Αρχική κατάσταση του NFA η αρχική του «prefix» DFA, τελική κατάσταση του NFA η τελική του «suffix» DFA.

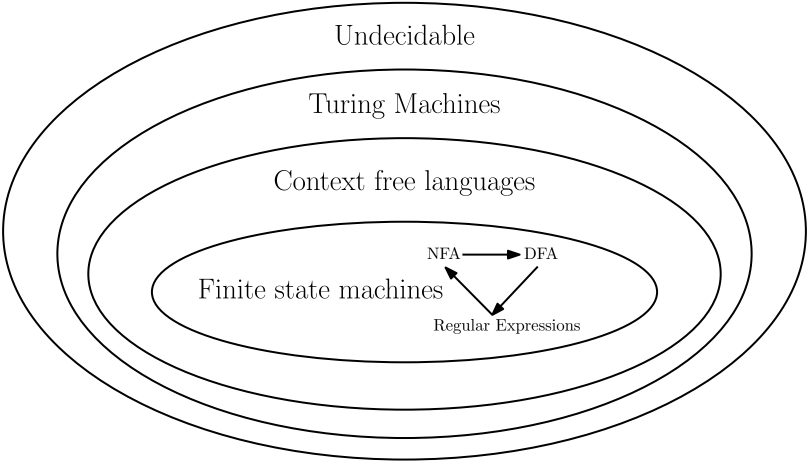
$L_1 = \{w : w \text{ περιέχει ζυγό \# } 0\}$, $L_2 = \{w : w \text{ περιέχει } 101\}$. NFA για $L_1 L_2$:



Κλειστότητες κανονικών γλωσσών

- Κλειστότητα ως προς την τομή ($L_1 \cap L_2 = \overline{\overline{L_1} \cup \overline{L_2}}$): DFA για $L_1, L_2 \Rightarrow$ εναλλαγή αρχικών–τελικών καταστάσεων $\Rightarrow$ DFA για $\overline{L_1}, \overline{L_2} \Rightarrow$ κατασκευή NFA για $\overline{L_1} \cup \overline{L_2} \Rightarrow$ κατασκευή ισοδύναμου DFA $\Rightarrow$ εναλλαγή αρχικών–τελικών καταστάσεων $\Rightarrow$ DFA για $L_1 \cap L_2$.
- Υπάρχει πιο αποδοτικός τρόπος: κατασκευή του γινομένου των DFA:
  - Ένα DFA προσομοιώνει την ταυτόχρονη λειτουργία και των δύο DFA.
  - Αποφεύγουμε τη δημιουργία NFA (εκθετικό κόστος).
  - Οι καταστάσεις του νέου DFA είναι το γινόμενο των καταστάσεων των L_1, L_2.

Μια ματιά στο σύμπαν



Τι δεν υπολογίζεται με FSMs;

- Οτιδήποτε απαιτεί μη πεπερασμένη μνήμη: π.χ. αν χρειάζεται να μετρήσουμε.
- Παραδείγμα γλωσσών που δεν είναι κανονικές: $L_1 = \{w : w \in 0^n 1^n\}$, $L_2 = \{w : w \in 0^{n^2}\}$, κτλ.
- Γιατί ένα FSM δεν μπορεί να αναγνωρίσει την L_1;
  - Διαισθητικά: χρειάζονται άπειρες καταστάσεις.
  - Έστω ότι κάποιο FSM με k καταστάσεις αναγνωρίζει την L_1.
  - Αν θέσουμε σαν είσοδο το $0^k 1^k$ στο FSM τότε αφού η είσοδος έχει μήκος $2k$ κάποιες καταστάσεις (τουλάχιστο μία) στον υπολογισμό του FSM θα επαναλαμβάνονται (αρχή του περιστεριώνα).
  - Η ύπαρξη αυτής της επανάληψης κάνει το FSM να αποδέχεται και άλλες συμβολοσειρές εκτός των $0^k 1^k$.
- Χρειαζόμαστε «ισχυρότερο» μοντέλο υπολογισμού.