



ΗΥ - 280 Θεωρία Υπολογισμού

Διάλεξη 4

Κωνσταντίνος Βάρσος

Τμήμα Επιστήμης Υπολογιστών – Πανεπιστήμιο Κρήτης

Χειμερινό εξάμηνο 2025 - 26

Γλώσσες και Αυτόματα

Τι είδαμε στην Διάλεξη 3

- Μη-ντετερμινιστικά πεπερασμένα αυτόματα DFA.
- ϵ -κλειστότητα.
- Αλγόριθμος διαγραφής ϵ -κλειστότητας.

Μετατροπή NFA σε DFA

Παρατηρήσεις

- Ένα DFA είναι και NFA. [δεν χρησιμοποιεί τον μη-ντετερμινισμό]
- Αν $\mathcal{N}$ είναι η κλάση όλων των γλωσσών που αναγνωρίζεται από κάποιο NFA, τότε οι κανονικές γλώσσες είναι υποσύνολο της $\mathcal{N}$.
- Ισχύει το αντίστροφο;

Μετατροπή NFA σε DFA

Θεώρημα. Για κάθε DFA M υπάρχει NFA N τ.ω. $L(M) = L(N)$.

Απόδειξη. Από τον ορισμό των NFA (N) και DFA (M) , για τις μεταβάσεις ισχύει

$$\delta_N = K \times \Sigma \cup \{\varepsilon\} \rightarrow 2^K, \quad \delta_M = K \times \Sigma \rightarrow K.$$

Για το DFA $M = (K_M, \Sigma, s_M, \delta_M, F_M)$ ορίζουμε το NFA $N = (K_N, \Sigma, s_N, \delta_N, F_N)$, να έχει ακριβώς το ίδιο K , s και F . Οι μεταβάσεις θα μετασχηματιστούν ως εξής:

$$\delta_N(q, a) = \{\delta_M(q, a)\}, \quad \text{για } a \in \Sigma,$$

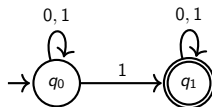
και $\delta_N(q, \varepsilon) = \emptyset, \forall q \in K$.



Μετατροπή NFA σε DFA

Αλγόριθμος Προσομοίωσης NFA. Σημειώνει την τρέχουσα κατάσταση και καθέναν από τους ενεργούς υπολογισμούς (ενεργά υπολογιστικά νήματα).

π.χ.,



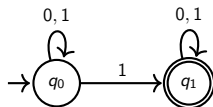
Για είσοδο $w = 111$, η προσομοίωση θα είναι

$$\langle q_0 \rangle \xrightarrow{1} \langle q_0, q_1 \rangle \xrightarrow{1} \langle q_0, q_1, q_1 \rangle \xrightarrow{1} \langle q_0, q_1, q_1, q_1 \rangle.$$

Μετατροπή NFA σε DFA

Αλγόριθμος Προσομοίωσης NFA. Σημειώνει την τρέχουσα κατάσταση και καθέναν από τους ενεργούς υπολογισμούς (ενεργά υπολογιστικά νήματα).

π.χ.,



Για είσοδο $w = 111$, η προσομοίωση θα είναι

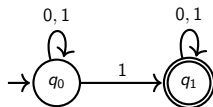
$$\langle q_0 \rangle \xrightarrow{1} \langle q_0, q_1 \rangle \xrightarrow{1} \langle q_0, q_1, q_1 \rangle \xrightarrow{1} \langle q_0, q_1, q_1, q_1 \rangle.$$

- Αύξηση στην μνήμη.
- Η σειρά των υπολογιστικών νημάτων δεν παίζει ρόλο.
- Ίδια υπολογιστικά νήματα, κρατάμε το ένα.

Μετατροπή NFA σε DFA

Αλγόριθμος Προσομοίωσης NFA. Σημειώνει την τρέχουσα κατάσταση και καθέναν από τους ενεργούς υπολογισμούς (ενεργά υπολογιστικά νήματα).

π.χ.,



Για είσοδο $w = 111$, η προσομοίωση θα είναι

$$\{q_0\} \xrightarrow{1} \{q_0, q_1\} \xrightarrow{1} \{q_0, q_1\} \xrightarrow{1} \{q_0, q_1\}$$

- Καμία σειρά (διάταξη), όχι διπλότυπα.
- Πρέπει να κρατήσουμε υποσύνολο του $K!$.
- $|K|$ bits μνήμης (δηλαδή, $2^{|K|}$ καταστάσεις), πεπερασμένη.

Μετατροπή NFA σε DFA

Αλγόριθμος, κατασκευή ισοδύναμου DFA.

- Το DFA εκτελεί τον αλγόριθμο προσομοίωσης.
- Το DFA θυμάται τις τρέχουσες καταστάσεις των ενεργών υπολογιστικών νημάτων, χωρίς διπλότυπα, [διατηρεί ένα υποσύνολο των καταστάσεων του NFA]
- Όταν διαβάζει ένα νέο σύμβολο, ενημερώνει τις καταστάσεις των ενεργών υπολογιστικών νημάτων.
- Αποδέχεται οποτεδήποτε ένα από τα υπολογιστικά νήματα είναι σε τερματική κατάσταση.

Μετατροπή NFA σε DFA

Θεώρημα. Για κάθε NFA N υπάρχει DFA M τ.ω. $L(N) = L(M)$.

Κατασκευή. Έστω NFA $N = (K, \Sigma, s, \delta, F)$, θα κατασκευάσουμε DFA $M = (K', \Sigma, s', \delta', F')$.
[Ξεκινάμε από την περίπτωση όπου το N δεν έχει ε -μεταβάσεις]

1. Νέο σύνολο αρχικών καταστάσεων $K' = 2^K$.
2. Μεταβάσεις: Για κάθε $R \in K'$ και σύμβολο $a \in \Sigma$ έχουμε

$$\tilde{\delta}(R, a) = \{q \in K \mid q \in \delta(r, a) \text{ για κάποιο } r \in R\},$$

Κάθε R κατάσταση του M , είναι σύνολο καταστάσεων του N . Κάθε τέτοια κατάσταση διαβάζοντας a πηγαίνει σε σύνολο καταστάσεων.

Εναλλακτικά,

$$\delta'(R, a) = \bigcup_{r \in R} \delta(r, a).$$

Μετατροπή NFA σε DFA

Κατασκευή (συνέχεια),

3. Αρχική κατάσταση, $q'_0 = \{q_0\}$ [Το M ξεκινάει από το υποσύνολο των καταστάσεων K' που περιέχουν μόνο την αρχική κορυφή του N]

4. Τελικές καταστάσεις,

$$F' = \{R \in K' \mid R \text{ περιέχουν τελική κορυφή του } N\},$$

Κατασκευάσαμε το $M = (K', \Sigma, s', \delta, F')$.

Μετατροπή NFA σε DFA

Κατασκευή (συνέχεια),

3. Αρχική κατάσταση, $q'_0 = \{q_0\}$ [Το M ξεκινάει από το υποσύνολο των καταστάσεων K' που περιέχουν μόνο την αρχική κορυφή του N]

4. Τελικές καταστάσεις,

$$F' = \{R \in K' \mid R \text{ περιέχουν τελική κορυφή του } N\},$$

Κατασκευάσαμε το $M = (K', \Sigma, s', \tilde{\delta}, F')$.

Αν έχουμε ε -μεταβάσεις χρησιμοποιούμε τον αλγόριθμο διαγραφής ε -μεταβάσεων, άρα από την $\tilde{\delta}$ πάμε στην δ' . □

Μετατροπή NFA σε DFA

Απόδειξη (συνέχεια), Τέλος, με μαθηματική επαγωγή στο μήκος της συμβολοσειράς εισόδου δείχνουμε ότι $L(N) = L(M)$.

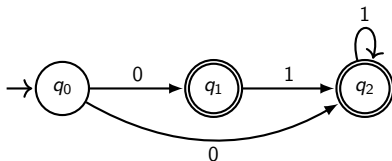
Χρησιμοποιούμε την επέκταση μεταβάσεων. Έστω μετάβαση $\delta : K \times \Sigma \rightarrow 2^K$

$$(\delta^*(q, \varepsilon) = \{q\}, \forall q \in K), \quad \left(\delta^*(q, wx) = \bigcup_{q' \in \delta^*(q, w)} \delta(q', x), \text{ για } x \in \Sigma, w \in \Sigma^* \right)$$

- $\delta'(p, a) = \bigcup_{r \in p \subseteq K} \delta(r, a)$
- Ισχύει: $\delta'^*(\{q\}, w) = \delta^*(q, w)$ (1). Απόδειξη της σχέσης με επαγωγή ως προς w
 - Αν $w = \varepsilon$: $\delta^*(q, w) = \{q\} = \delta'^*(\{q\}, w)$.
 - Έστω ότι ισχύει η (1) για το $w \in \Sigma^*$ και έστω $a \in \Sigma$. Τότε:
 - $\delta^*(q, wa) = \bigcup_{t \in \delta^*(q, w)} \delta(t, a) = \bigcup_{t \in \delta'^*(\{q\}, w)} \delta(t, a) = \delta'(\delta'^*(\{q\}, w), a) = \delta'^*(\{q\}, wa)$.
- Οπότε: $w \in L(M) \Leftrightarrow \delta^*(q_0, w) \cap F \neq \emptyset \Leftrightarrow \delta'^*(q_0, w) \in F' \Leftrightarrow w \in L(N)$. $\square$

Μη-ντετερμινιστικά Πεπερασμένα Αυτόματα

π.χ., Έστω το παρακάτω NFA, N , θα κατασκευάσουμε το ισοδύναμο DFA, M .



Σύνολο των εν δυνάμει καταστάσεων του M' : $K' = 2^K$, δηλαδή

$$K' = \{\emptyset, \{q_0\}, \{q_1\}, \{q_2\}, \{q_0, q_1\}, \{q_0, q_2\}, \{q_1, q_2\}, \{q_0, q_1, q_2\}\}.$$

Για ευκολία γράφουμε q_{ij} αντί για $q_i q_j$ και q_{ijk} αντί για $q_i q_j q_k$. Άρα το K' έχει τα στοιχεία

$$\{\emptyset, \{q_0\}, \{q_1\}, \{q_2\}, \{q_{01}\}, \{q_{02}\}, \{q_{12}\}, \{q_{012}\}\}.$$

Μη-ντετερμινιστικά Πεπερασμένα Αυτόματα

π.χ., (συνέχεια)

Μεταβάσεις: Για κάθε $R \in K'$ και σύμβολο $a \in \Sigma$ έχουμε

$$\delta'(R, a) = \{q \in K \mid q \in \delta(r, a) \text{ για κάποιο } r \in R\},$$

δηλαδή

δ'	$\emptyset$	$\{q_0\}$	$\{q_1\}$	$\{q_2\}$	$\{q_{01}\}$	$\{q_{02}\}$	$\{q_{12}\}$	$\{q_{012}\}$
0	$\emptyset$	$\{q_{12}\}$	$\emptyset$	$\emptyset$	$\{q_{12}\}$	$\{q_{12}\}$	$\emptyset$	$\{q_{12}\}$
1	$\emptyset$	$\emptyset$	$\{q_2\}$	$\{q_2\}$	$\{q_2\}$	$\{q_2\}$	$\{q_2\}$	$\{q_2\}$

Μη-ντετερμινιστικά Πεπερασμένα Αυτόματα

π.χ., (συνέχεια)

Μεταβάσεις: Για κάθε $R \in K'$ και σύμβολο $a \in \Sigma$ έχουμε

$$\delta'(R, a) = \{q \in K \mid q \in \delta(r, a) \text{ για κάποιο } r \in R\},$$

δηλαδή

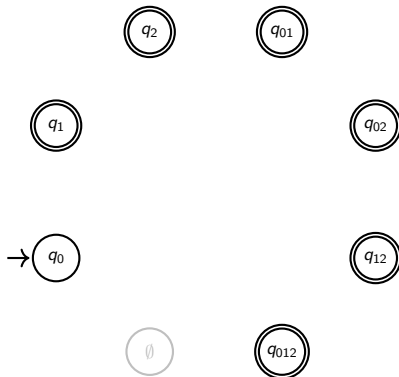
δ'	$\emptyset$	$\{q_0\}$	$\{q_1\}$	$\{q_2\}$	$\{q_{01}\}$	$\{q_{02}\}$	$\{q_{12}\}$	$\{q_{012}\}$
0	$\emptyset$	$\{q_{12}\}$	$\emptyset$	$\emptyset$	$\{q_{12}\}$	$\{q_{12}\}$	$\emptyset$	$\{q_{12}\}$
1	$\emptyset$	$\emptyset$	$\{q_2\}$	$\{q_2\}$	$\{q_2\}$	$\{q_2\}$	$\{q_2\}$	$\{q_2\}$

■ Αρχική κατάσταση: $q'_0 = q_0$

■ Τελικές καταστάσεις: $F' = \{q_1, q_2, q_{01}, q_{02}, q_{12}, q_{012}\}$.

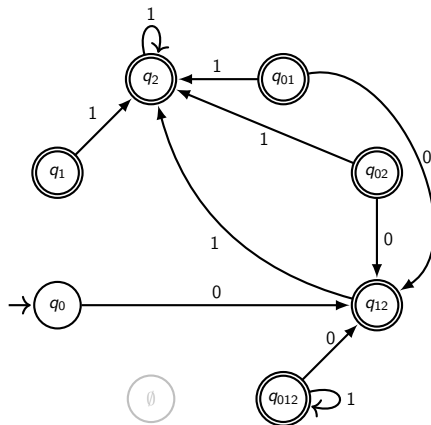
Μη-ντετερμινιστικά Πεπερασμένα Αυτόματα

π.χ., (συνέχεια) Αρχικά έχουμε τις καταστάσεις,



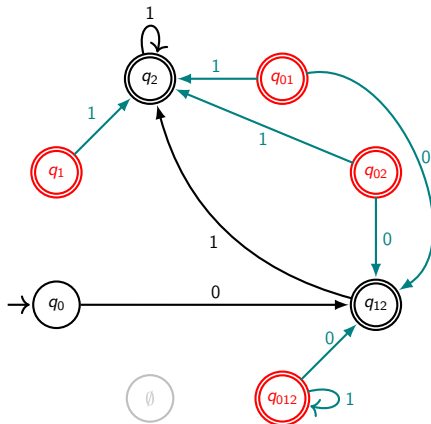
Μη-ντετερμινιστικά Πεπερασμένα Αυτόματα

π.χ., (συνέχεια) Από τον πίνακα μεταβάσεων γράφουμε τις ακμές



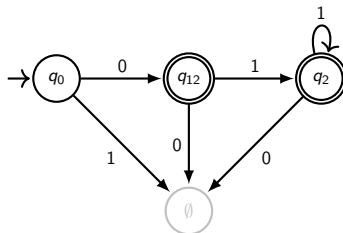
Μη-ντετερμινιστικά Πεπερασμένα Αυτόματα

π.χ., (συνέχεια) Διαγράφουμε τις καταστάσεις στις οποίες **δεν** μπορούμε να μεταβούμε αν ξεκινήσουμε από την αρχική κατάσταση, εδώ είναι η q_0 . [μη-προσβάσιμες από την q_0]



Μη-ντετερμινιστικά Πεπερασμένα Αυτόματα

π.χ., (συνέχεια) Το ισοδύναμο DFA έχει την μορφή



Μετατροπή NFA σε DFA

Μερικά σχόλια

- Όταν το M διαβάζει ένα σύμβολο a σε μια κατάσταση q , τότε 'παίρνουμε' όλες τις καταστάσεις που οδηγεί το a από το q .
- Καθώς κάθε κατάσταση μπορεί να είναι σύνολο καταστάσεων, λαμβάνουμε υπόψιν μας την ένωση όλων αυτών των συνόλων.
- Το M 'θυμάται' τα υποσύνολα των πιθανών καταστασεών του N .

Μη-ντετερμινιστικά Πεπερασμένα Αυτόματα

Συνέπεια του προηγούμενου θεωρήματος είναι το παρακάτω αποτέλεσμα.

Θεώρημα. Αν ένα NFA αναγνωρίζει μια γλώσσα A , τότε η A είναι κανονική.

Απόδειξη.

- Έστω το NFA $M = (K, \Sigma, \delta, s, F)$ που αναγνωρίζει την γλώσσα A .
- Κατασκευή DFA $M' = (K', \Sigma, \delta', s', F')$ ώστε να αναγνωρίζει την A .
- M' θυμάται όλα τα υποσύνολα των δυνατών συνόλων καταστάσεων της M . □

Μη-ντετερμινιστικά Πεπερασμένα Αυτόματα

Συνέπεια του προηγούμενου θεωρήματος είναι το παρακάτω αποτέλεσμα.

Θεώρημα. Αν ένα NFA αναγνωρίζει μια γλώσσα A , τότε η A είναι κανονική.

Απόδειξη.

- Έστω το NFA $M = (K, \Sigma, \delta, s, F)$ που αναγνωρίζει την γλώσσα A .
- Κατασκευή DFA $M' = (K', \Sigma, \delta', s', F')$ ώστε να αναγνωρίζει την A .
- M' θυμάται όλα τα υποσύνολα των δυνατών συνόλων καταστάσεων της M . □

■ Ισχύει το αντίστροφο; ✓

■ Αν $|K| = n$, τότε $|K'| = 2^n$; Αν το N έχει m το πλήθος μεταβάσεις, πόσες θα έχει το M ;

Κλειστότητα για Κανονικές Γλώσσες

Θεώρημα. Αν οι γλώσσες A , B είναι κανονικές, τότε και η γλώσσα $A \cup B$ είναι κανονική.

Κλειστότητα για Κανονικές Γλώσσες

Θεώρημα. Αν οι γλώσσες A , B είναι κανονικές, τότε και η γλώσσα $A \cup B$ είναι κανονική.

Κατασκευή. Έστω το NFA $N_A = (K_A, \Sigma, \delta_A, s_A, F_A)$ που αναγνωρίζει την A και το NFA $N_B = (K_B, \Sigma, \delta_B, s_B, F_B)$ που αναγνωρίζει την B .

■ Κατασκευάζουμε NFA $N = (K, \Sigma, \delta, s, F)$ το οποίο να αναγνωρίζει την $A \cup B$.

- 1 $K = \{q_0\} \cup K_A \cup K_B$
- 2 Αρχική κατάσταση για το N η q_0
- 3 Τελικές καταστάσεις $F = F_A \cup F_B$
- 4 Μεταβάσεις δ για κάθε $q \in K$ και $a \in \Sigma \cup \{\varepsilon\}$

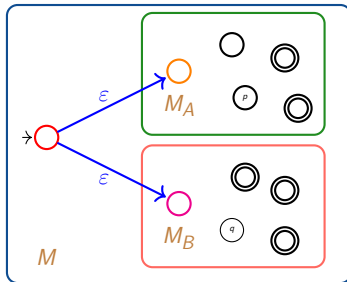
$$\delta(q, a) = \begin{cases} \delta_A(q, a) & , q \in K_A \\ \delta_B(q, a) & , q \in K_B \\ \{q_A, q_B\} & , q = q_0 \text{ και } a = \varepsilon \\ \emptyset & , q = q_0 \text{ και } a \neq \varepsilon \end{cases}$$

□

Κλειστότητα για Κανονικές Γλώσσες

Θεώρημα. Αν οι γλώσσες A , B είναι κανονικές, τότε και η γλώσσα $A \cup B$ είναι κανονική.

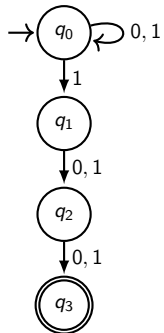
Σχηματικά,



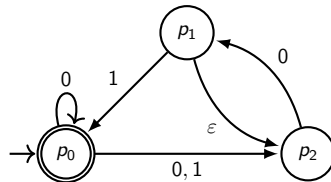
Κλειστότητα για Κανονικές Γλώσσες

π.χ., Έστω N_1 γλώσσα A και N_2 για γλώσσα B .

N_1

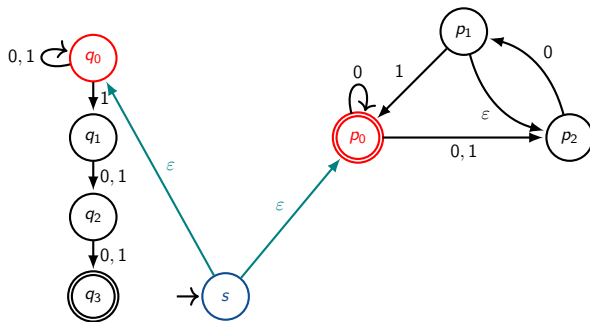


N_2



Κλειστότητα για Κανονικές Γλώσσες

π.χ., Έστω N_1 γλώσσα A και N_2 για γλώσσα B . N για την γλώσσα $A \cup B$,



Κλειστότητα για Κανονικές Γλώσσες

Θεώρημα. Αν οι γλώσσες A , B είναι κανονικές, τότε και η γλώσσα AB είναι κανονική.

Κλειστότητα για Κανονικές Γλώσσες

Θεώρημα. Αν οι γλώσσες A , B είναι κανονικές, τότε και η γλώσσα AB είναι κανονική.

Κατασκευή. Έστω το NFA $N_A = (K_A, \Sigma, \delta_A, s_A, F_A)$ που αναγνωρίζει την A και το NFA $N_B = (K_B, \Sigma, \delta_B, s_B, F_B)$ που αναγνωρίζει την B . Κατασκευάζουμε NFA $N = (K, \Sigma, \delta, s, F)$ το οποίο να αναγνωρίζει την AB

$$1 \quad K = K_A \cup K_B$$

2 Αρχική κατάσταση για το N η αρχική κατάσταση του N_A , q_A

$$3 \quad \text{Τελικές καταστάσεις } F = F_B$$

4 Μεταβάσεις δ για κάθε $q \in K$ και $a \in \Sigma \cup \{\varepsilon\}$

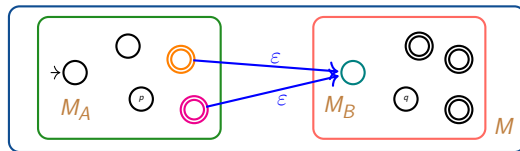
$$\delta(q, a) = \begin{cases} \delta_A(q, a) & , q \in K_A \text{ και } q \notin F_A \\ \delta_A(q, a) & , q \in F_A \text{ και } a \neq \varepsilon \\ \delta_A(q, a) \cup \{q_B\} & , q \in F_A \text{ και } a = \varepsilon \\ \delta_B(q, a) & , q \in K_B \end{cases}$$

□

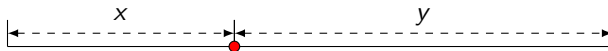
Κλειστότητα για Κανονικές Γλώσσες

Θεώρημα. Αν οι γλώσσες A , B είναι κανονικές, τότε και η γλώσσα AB είναι κανονική.

Σχηματικά,



Το M δέχεται την w αν χωρίζεται σε $w = xy$, όπου η M_A δέχεται την x και η M_B δέχεται την y .

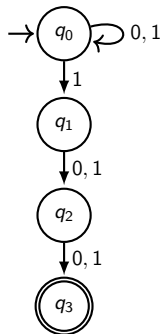


■ Στο NFA M περνάμε από το M_A στο M_B , όταν M_A δέχεται συμβολοσειρά.

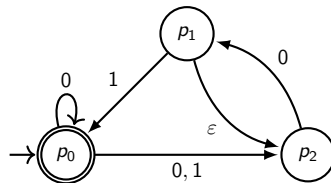
Κλειστότητα για Κανονικές Γλώσσες

π.χ., Έστω N_1 γλώσσα A και N_2 για γλώσσα B . N για την γλώσσα AB ,

N_1



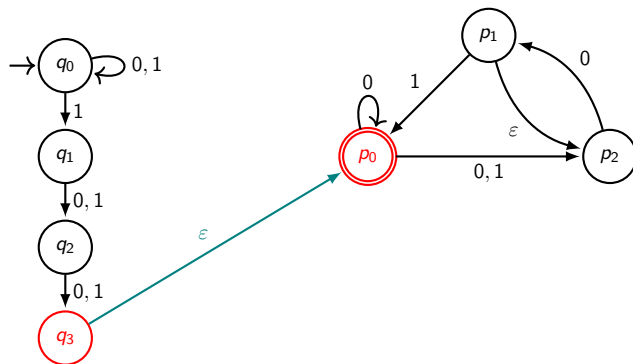
N_2



Κλειστότητα για Κανονικές Γλώσσες

π.χ., (συνέχεια) Έστω N_1 γλώσσα A και N_2 για γλώσσα B . N για την γλώσσα AB ,

N



Κλειστότητα για Κανονικές Γλώσσες

Θεώρημα. Αν η γλώσσα A είναι κανονική, τότε και η γλώσσα A^* είναι κανονική.

Κλειστότητα για Κανονικές Γλώσσες

Θεώρημα. Αν η γλώσσα A είναι κανονική, τότε και η γλώσσα A^* είναι κανονική.

Κατασκευή. Έστω το NFA $N_A = (K_A, \Sigma, \delta_A, s_A, F_A)$ που αναγνωρίζει την A . Κατασκευάζουμε NFA $N = (K, \Sigma, \delta, s, F)$ το οποίο να αναγνωρίζει την A^*

- 1 $K = \{q_0\} \cup K_A$
- 2 Αρχική κατάσταση για το N η q_0
- 3 Τελικές καταστάσεις $F = \{q_0\} \cup K_A$
- 4 Μεταβάσεις δ για κάθε $q \in K$ και $a \in \Sigma \cup \{\varepsilon\}$

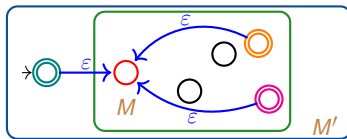
$$\delta(q, a) = \begin{cases} \delta_A(q, a) & , q \in K_A \text{ και } q \notin F_A \\ \delta_A(q, a) & , q \in F_A \text{ και } a \neq \varepsilon \\ \delta_A(q, a) \cup \{q_A\} & , q \in F_A \text{ και } a = \varepsilon \\ \{q_1\} & , q = q_0 \text{ και } a = \varepsilon \\ \emptyset & , q = q_0 \text{ και } a \neq \varepsilon \end{cases}$$

□

Κλειστότητα για Κανονικές Γλώσσες

Θεώρημα. Αν η γλώσσα A είναι κανονική, τότε και η γλώσσα A^* είναι κανονική.

Σχηματικά,



Το M' πρέπει να δέχεται

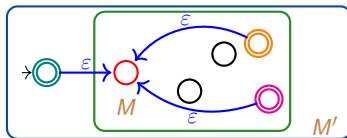
- ε και w , αν $w = x_1x_2 \dots x_k$, όπου $k \geq 0$ και το M δέχεται το x_i .



Κλειστότητα για Κανονικές Γλώσσες

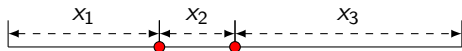
Θεώρημα. Αν η γλώσσα A είναι κανονική, τότε και η γλώσσα A^* είναι κανονική.

Σχηματικά,



Το M' πρέπει να δέχεται

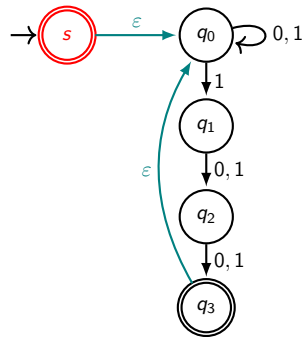
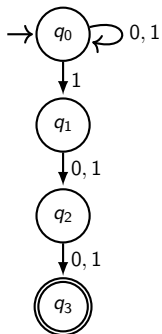
- ε και w , αν $w = x_1x_2 \dots x_k$, όπου $k \geq 0$ και το M δέχεται το x_i .



- Αν το DFA M έχει $|K| = n$, τότε $|K'| =$;

Κλειστότητα για Κανονικές Γλώσσες

π.χ., (συνέχεια) Έστω N_1 γλώσσα A , N για την γλώσσα A^* ,
 N_1 N



Κανονικές Γλώσσες

- Αν A κανονική γλώσσα, τότε και η $\bar{A}$ είναι κανονική.
- Η $\bar{A}$ περιέχει όλες τις συμβολοσειρές που δεν αναγνωρίζει η A .

Κανονικές Γλώσσες

- Αν A κανονική γλώσσα, τότε και η $\bar{A}$ είναι κανονική.
- Η $\bar{A}$ περιέχει όλες τις συμβολοσειρές που δεν αναγνωρίζει η A .
- Στο ντετερμινιστικό αυτόματο που αναγνωρίζει την A , άλλαξε τις τελικές σε μη-τελικές και το αντίστροφο.

Κανονικές Γλώσσες

- Αν A κανονική γλώσσα, τότε και η $\bar{A}$ είναι κανονική.
- Η $\bar{A}$ περιέχει όλες τις συμβολοσειρές που δεν αναγνωρίζει η A .
- Στο ντετερμινιστικό αυτόματο που αναγνωρίζει την A , άλλαξε τις τελικές σε μη-τελικές και το αντίστροφο.
- Αν A, B κανονικές γλώσσες, τότε και η $A \cap B$ είναι κανονική.

Κανονικές Γλώσσες

- Αν A κανονική γλώσσα, τότε και η $\bar{A}$ είναι κανονική.
- Η $\bar{A}$ περιέχει όλες τις συμβολοσειρές που δεν αναγνωρίζει η A .
- Στο ντετερμινιστικό αυτόματο που αναγνωρίζει την A , άλλαξε τις τελικές σε μη-τελικές και το αντίστροφο.
- Αν A, B κανονικές γλώσσες, τότε και η $A \cap B$ είναι κανονική.
- Κανόνας de Morgan: $A \cap B = \overline{(\bar{A} \cup \bar{B})}$.

Κανονικές Γλώσσες

- Μια γλώσσα σε ένα αλφάβητο Σ είναι κανονική εάν υπάρχει **κανονική έκφραση** στο Σ που την **περιγράφει**.
- Είναι ένας αλγεβρικός τρόπος για να αναπαραστήσουμε κανονικές γλώσσες.
- Έχουν αρκετές πρακτικές εφαρμογές, όπως ταιριάσματα, αναζήτηση, text, σχεδιασμός μεταγλωττιστών κτλ.
- Για συντομία θα αναφερόμαστε στις κανονικές εκφράσεις ως *ΚΕ*.

Κανονικές Εκφράσεις

Ορισμός. Λέμε ότι η R είναι Regex αν η R αναπαρίσταται με έναν από τους παρακάτω τρόπους

Regex	$\mathcal{L}(\text{Regex})$	
$\emptyset$	$\{\}$	κενό σύνολο
ε	$\{\varepsilon\}$	σύνολο που περιέχει το $\{\varepsilon\}$
a	$\{a\}$	$a \in \Sigma$
$R_1 \cup R_2$	$\mathcal{L}(R_1) \cup \mathcal{L}(R_2)$	για Regex R_1 και R_2
$R_1 R_2$	$\mathcal{L}(R_1)\mathcal{L}(R_2)$	για Regex R_1 και R_2
R_1^*	$(\mathcal{L}(R_1))^*$	για Regex R_1
(R_1)	$\mathcal{L}(R_1)$	για Regex R_1

Κανονικές Εκφράσεις

Ορισμός. Λέμε ότι η R είναι Regex αν η R αναπαρίσταται με έναν από τους παρακάτω τρόπους

Regex	$\mathcal{L}(\text{Regex})$	
$\emptyset$	$\{\}$	κενό σύνολο
ε	$\{\varepsilon\}$	σύνολο που περιέχει το $\{\varepsilon\}$
a	$\{a\}$	$a \in \Sigma$
$R_1 \cup R_2$	$\mathcal{L}(R_1) \cup \mathcal{L}(R_2)$	για Regex R_1 και R_2
$R_1 R_2$	$\mathcal{L}(R_1)\mathcal{L}(R_2)$	για Regex R_1 και R_2
R_1^*	$(\mathcal{L}(R_1))^*$	για Regex R_1
(R_1)	$\mathcal{L}(R_1)$	για Regex R_1

■ Κάθε Regex αντιστοιχεί σε μοναδική γλώσσα. Το αντίστροφο ισχύει ;

■ Στην βιβλιογραφία το $\cup$ γράφεται και ως $+$.

■ Προτεραιότητα πράξεων (από την μεγαλύτερη προς την μικρότερη): $()$, $*$, $\circ$, $\cup$. [Ως συνήθως, οι $()$ αλλάζουν την προτεραιότητα των πράξεων.]

Κανονικές Εκφράσεις

■ Αν α κανονική έκφραση, τότε $\mathcal{L}(\alpha)$ είναι η αντίστοιχη γλώσσα,

1 $\mathcal{L}(\emptyset) = \emptyset, \mathcal{L}(w) = w.$

2 $\mathcal{L}((\alpha \cup \beta)) = \mathcal{L}(\alpha) \cup \mathcal{L}(\beta).$

3 $\mathcal{L}((\alpha\beta)) = \mathcal{L}(\alpha)\mathcal{L}(\beta).$

4 $\mathcal{L}((\alpha^*)) = \mathcal{L}(\alpha)^*.$

■ Οι Regex R_1 και R_2 είναι ισοδύναμες ($R_1 \equiv R_2$) αν $\mathcal{L}(R_1) = \mathcal{L}(R_2).$

Κανονικές Εκφράσεις

■ Αν α κανονική έκφραση, τότε $\mathcal{L}(\alpha)$ είναι η αντίστοιχη γλώσσα,

$$1 \quad \mathcal{L}(\emptyset) = \emptyset, \mathcal{L}(w) = w.$$

$$2 \quad \mathcal{L}((\alpha \cup \beta)) = \mathcal{L}(\alpha) \cup \mathcal{L}(\beta).$$

$$3 \quad \mathcal{L}((\alpha\beta)) = \mathcal{L}(\alpha)\mathcal{L}(\beta).$$

$$4 \quad \mathcal{L}((\alpha^*)) = \mathcal{L}(\alpha)^*.$$

■ Οι Regex R_1 και R_2 είναι ισοδύναμες ($R_1 \equiv R_2$) αν $\mathcal{L}(R_1) = \mathcal{L}(R_2)$.

π.χ., Ποια η γλώσσα της $(0 \cup 1)^*$;

$$\mathcal{L}((0 \cup 1)^*) = \mathcal{L}((0 \cup 1))^* = (\mathcal{L}(0) \cup \mathcal{L}(1))^* = (\{0\} \cup \{1\})^* = \{0, 1\}^*.$$

Κανονικές Εκφράσεις

π.χ., Βρείτε κανονική έκφραση για την γλώσσα ...

$$A = \{w \in \{0, 1\}^* \mid w \text{ περιέχει } 10 \text{ και τελειώνει σε } 1\}$$

Κανονικές Εκφράσεις

π.χ., Βρείτε κανονική έκφραση για την γλώσσα ...

$$A = \{w \in \{0, 1\}^* \mid w \text{ περιέχει } 10 \text{ και τελειώνει σε } 1\}$$

$$(0 \cup 1)^* 10 (0 \cup 1)^* 1$$

Κανονικές Εκφράσεις

π.χ., Βρείτε κανονική έκφραση για την γλώσσα ...

$$A = \{w \in \{0, 1\}^* \mid w \text{ περιέχει } 10 \text{ και τελειώνει σε } 1\}$$

$$(0 \cup 1)^* 10 (0 \cup 1)^* 1$$

$$B = \{w \in \{\omega, \theta\}^* \mid w \text{ περιέχει ζυγό αριθμό } \omega\}$$

Κανονικές Εκφράσεις

π.χ., Βρείτε κανονική έκφραση για την γλώσσα ...

$$A = \{w \in \{0, 1\}^* \mid w \text{ περιέχει } 10 \text{ και τελειώνει σε } 1\}$$

$$(0 \cup 1)^* 10 (0 \cup 1)^* 1$$

$$B = \{w \in \{\omega, \theta\}^* \mid w \text{ περιέχει ζυγό αριθμό } \omega\}$$

$$(\theta^* \omega \theta^* \omega \theta^*)^* \text{ ή } (\theta^* \omega \theta^* \omega)^* \theta^* \text{ ή } (\theta \cup \omega \theta^* \omega)^*$$

Κανονικές Εκφράσεις

π.χ., Βρείτε κανονική έκφραση για την γλώσσα ...

$$A = \{w \in \{0, 1\}^* \mid w \text{ περιέχει } 10 \text{ και τελειώνει σε } 1\}$$

$$(0 \cup 1)^* 10 (0 \cup 1)^* 1$$

$$B = \{w \in \{\omega, \theta\}^* \mid w \text{ περιέχει ζυγό αριθμό } \omega\}$$

$$(\theta^* \omega \theta^* \omega \theta^*)^* \text{ ή } (\theta^* \omega \theta^* \omega)^* \theta^* \text{ ή } (\theta \cup \omega \theta^* \omega)^*$$

$$C = \{w \in \{a, b\}^* \mid w \text{ τελειώνει με } b \text{ και δεν περιέχει } aa\}$$

$$(b \cup ab) \cup (b \cup ab)^* \text{ ή } (b \cup ab)^+$$