Εισαγωγή στην Επιστήμη των Υπολογιστών

4ο εξάμηνο Σ.Η.Μ.Μ.Υ. & Σ.Ε.Μ.Φ.Ε.

http://www.corelab.ece.ntua.gr/courses/

2η ενότητα: Γράφοι, Αυτόματα, Τυπικές Γραμματικές

Στάθης Ζάχος

Επιμέλεια: Αρης Παγουρτζής, Πάνος Χείλαρης

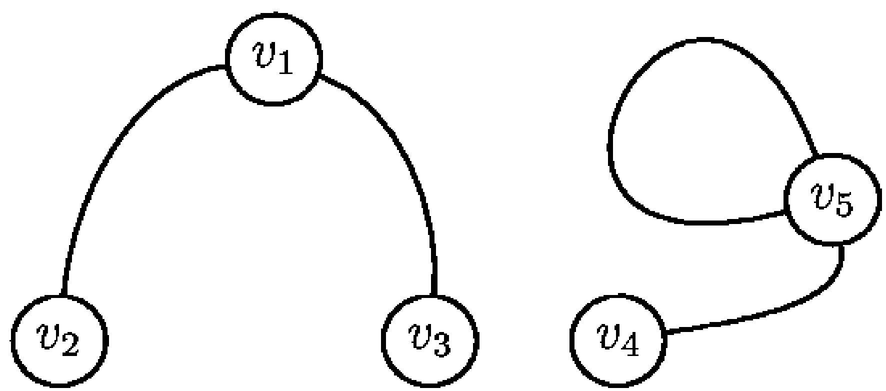
Γράφοι

Ορισμός 5.2.1. Γράφος (ή γράφημα) G, ονομάζεται ένα διατεταγμένο ζεύγος συνόλων (V, E), όπου V είναι μη κενό σύνολο στοιχείων και E ένα σύνολο μη διατεταγμένων ζευγών του V, δηλαδή

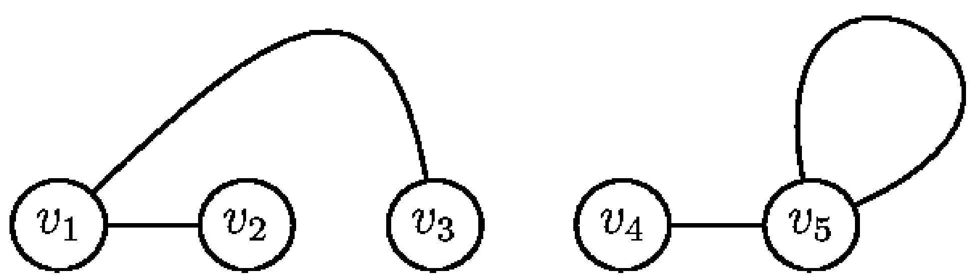
$$E \subseteq \binom{V}{2}$$

$$E = \{\{v_1, v_2\}, \{v_1, v_3\}, \{v_4, v_5\}, \{v_5, v_5\}\}$$

(α)



(β)



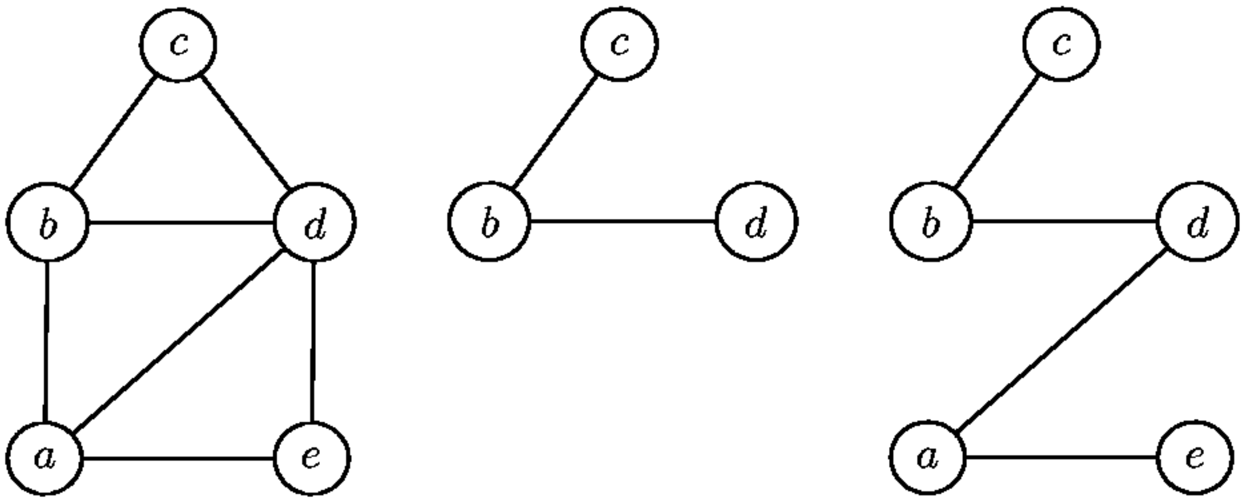
Γράφοι

Τα στοιχεία του μη κενού συνόλου V λέγονται **κορυφές** ή **κόμβοι** (*vertices, nodes*) του γράφου. Τα στοιχεία του συνόλου E λέγονται **ακμές** ή **πλευρές** (*edges*) και μπορούν να συμβολιστούν και με ένα γράμμα, π.χ. e, όπου $e = \{x, y\}, x, y \in V$, $x \neq y$. Καμιά φορά, καταχρηστικές θεωρούμε και ακμές-βρόχους, δηλαδή $e = \{x, x\}$.

Αν $e = \{v_1, v_2\}$ είναι πλευρά ενός γράφου G, αυτή ενώνει ή συνδέει τις κορυφές v_1, v_2 του G και μπορεί να συμβολιστεί επίσης ως v_1v_2 ή v_2v_1. Οι κορυφές v_1, v_2 λέγονται **άκρα** (*endpoints*) της πλευράς e επειδή δε η πλευρά e της συνδέει είναι **γειτονικές** (*adjacent*) κορυφές στο G.

Αν τώρα v_1, v_2 είναι γειτονικές κορυφές στο G, τότε η πλευρά v_1v_2 προσπίπτει (*incident*) στις v_1 και v_2. Δύο πλευρές που προσπίπτουν στην ίδια κορυφή είναι **γειτονικές** πλευρές στο G.

Υπογράφοι



Γράφος G Γράφος G' Γράφος G''

Σχήμα 5.3: Ο γράφος G και δύο υπογράφοι αυτού

Ορισμός 5.2.7. Ένας γράφος $G' = (V', E')$ είναι **υπογράφος** (*subgraph*) ενός άλλου γράφου $G = (V, E)$, αν ισχύει $V' \subseteq V$ και $E' \subseteq E$.

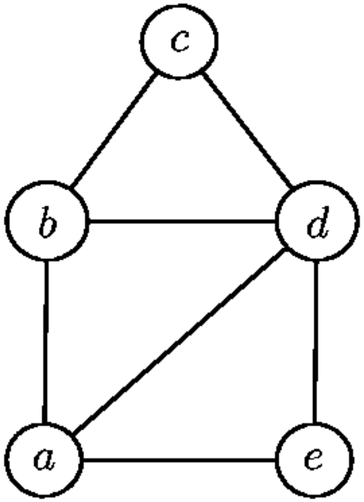
Δρόμος σε γράφο

Ορισμός 5.2.13. Σ'ένα γράφο G, μια πεπερασμένη ακολουθία, εναλλάξ κορυφών και πλευρών του G που αρχίζει και τελειώνει σε κορυφή και που κάθε πλευρά που περιέχεται στην ακολουθία προσπίπτει στην κορυφή που προηγείται και σ' αυτήν που έπεται, λέγεται **δρόμος ή διαδρομή** (*walk*) στο G.

Παράδειγμα 5.2.14. Στο γράφο G στο σχήμα 5.3 η ακολουθία κορυφών και πλευρών του γράφου

$$c\{c,d\}d\{d,b\}b\{b,a\}a\{a,d\}d\{d,b\}b$$

είναι δρόμος στο G.



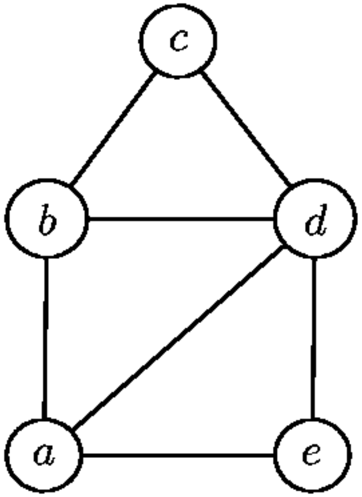
Μονοπάτι

Ορισμός 5.2.17. Ένας δρόμος στον οποίο κάθε κορυφή και κάθε πλευρά του εμφανίζονται ακριβώς μία φορά, λέγεται **απλό μονοπάτι** (*path*).

Παράδειγμα 5.2.18. Στο γράφο G στο σχήμα 5.3 ο δρόμος

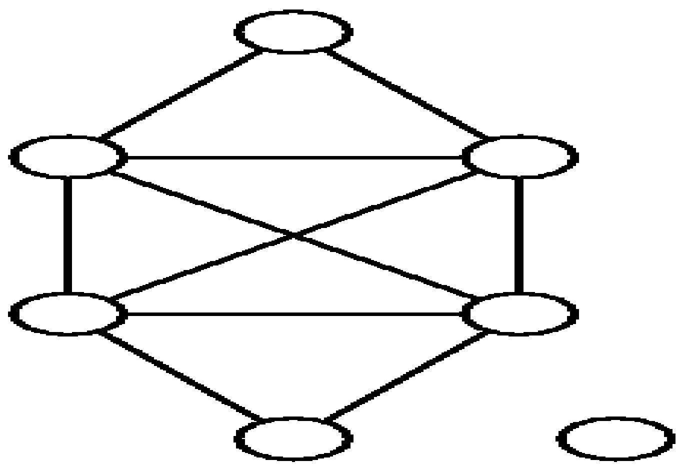
$$a\{a,b\}b\{b,c\}c\{c,d\}d\{d,e\}e$$

είναι απλό μονοπάτι.



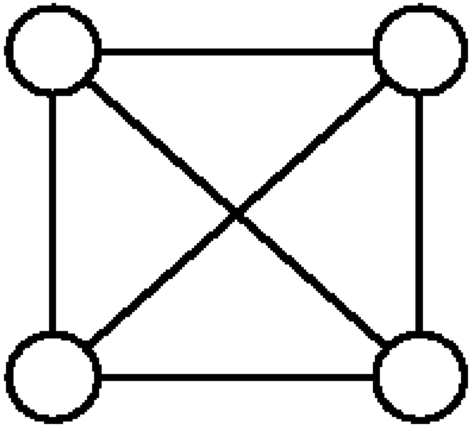
Γράφος Euler

Ορισμός 5.2.22. Ένας κύκλος που περνά ακριβώς μια φορά από κάθε πλευρά ενός γράφου G (χωρίς απαραίτητα να περνά ακριβώς μια φορά και από κάθε κορυφή) ονομάζεται **κύκλος Euler**. Ένας γράφος που έχει κύκλο Euler ονομάζεται **γράφος Euler**. Αποδεικνύεται εύκολα ότι ένας γράφος έχει κύκλο Euler ανν όλες οι κορυφές έχουν άρτιο βαθμό (σχήμα 5.5(α)).



Γράφος Hamilton

Ορισμός 5.2.23. Ένας κύκλος που περνά ακριβώς μια φορά από κάθε κορυφή ενός γράφου G (χωρίς απαραίτητα να περνά και από όλες τις πλευρές) ονομάζεται **κύκλος Hamilton**. Ένας γράφος που έχει κύκλο Hamilton ονομάζεται **γράφος Hamilton** (σχήμα 5.5(β)).



Πίνακας Γειτνίασης

Ορισμός 5.2.25 (Πίνακας γειτνίασης). Έστω ένας γράφος $G = (V, E)$ με $V = \{v_1, v_2, \ldots, v_n\}$. Τότε ο γράφος μπορεί να παρασταθεί με τη βοήθεια ενός $n \times n$ πίνακα $A(G)$, όπου

$$A(G) = [a_{ij}], \quad a_{ij} = \begin{cases} 1, & \text{αν } \{v_i, v_j\} \in E \\ 0, & \text{αλλιώς} \end{cases}$$

Ο πίνακας $A(G)$ λέγεται **πίνακας γειτνίασης** (*adjacency matrix*), και είναι συμμετρικός ($a_{i,j} = a_{j,i}$).

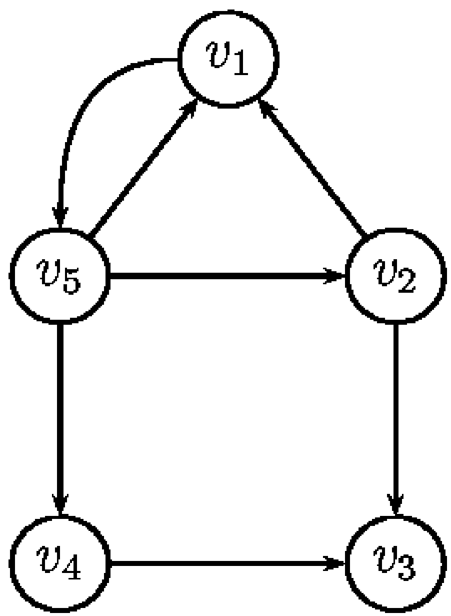
Πίνακας Πρόσπτωσης

Ορισμός 5.2.26. Έστω ένας γράφος $G = (V, E)$ με $V = \{v_1, v_2, ..., v_n\}$ και $E = \{l_1, l_2, ..., l_m\}$. Τότε ο γράφος μπορεί να παρασταθεί με τη βοήθεια ενός $n \times m$ πίνακα $B(G)$, που ονομάζεται **πίνακας πρόσπτωσης** (*incidence matrix*), όπου

$$B(G) = [b_{ij}], \quad b_{ij} = \begin{cases} 1, & \text{αν } l_j \text{ προσπίπτει στο } v_i \\ 0, & \text{αλλιώς} \end{cases}$$

Κατευθυνόμενος Γράφος

Αν στον ορισμό του γράφου αντικαταστήσουμε τα στοιχεία του E με διατεταγμένα ζεύγη στοιχείων του V, παίρνουμε ένα **προσανατολισμένο ή κατευθυνόμενο** γράφο (*directed graph, digraph*). Δηλαδή $E \subseteq V \times V$.



Συνεκτικός Γράφος

- Εάν υπάρχει δρόμος μεταξύ δύο οποιωνδήποτε κόμβων ο γράφος λέγεται συνεκτικός.
- Ένας κατευθυνόμενος γράφος είναι:
  - *ισχυρά συνεκτικός* αν υπάρχει δρόμος για κάθε ζεύγος κόμβων λαμβάνοντας υπ'όψη τις κατευθύνσεις.
  - *ασθενώς συνεκτικός* αν υπάρχει δρόμος για κάθε ζεύγος κόμβων αγνοώντας τις κατευθύνσεις.

Αυτόματα

Στο κεφάλαιο αυτό θα επιδιώξουμε να ταξινομήσουμε τα υπολογιστά σύνολα (γλώσσες) ανάλογα με το είδος του **αυτομάτου** (*αφηρημένης υπολογιστικής συσκευής*) που μπορεί να τα αναγνωρίσει.

Γενικά μπορούμε να κατατάξουμε τα αυτόματα ανάλογα με διάφορα κριτήρια όπως, π.χ., ως προς

- **αιτιότητα** (ντετερμινιστικά, μη ντετερμινιστικά, πιθανοτικά)
- **μέγεθος** (συγκεκριμένο, αυξανόμενο, άπειρο)
- **είσοδο/έξοδο** (διακριτές δηλ. ψηφιακά, συνεχείς δηλ. αναλογικά)
- **γενικό ρολόι** σε περίπτωση παραλληλίας (συγχρονισμένα, ασύγχρονα)
- **αριθμό καταστάσεων** (πεπερασμένο, άπειρο)
- **λειτουργία** (αναγνωριστικά (*acceptors*), γεννητικά (*generators*), υπολογιστικά (*transducers*)).

Αυτόματα

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 τώρα η συσκευή διαβάζει είσοδο, σύμβολο προς σύμβολο, από αριστερά προς τα δεξιά και ανάλογα αλλάζει καταστάσεις, τότε έχουμε ένα **αναγνωριστή πεπερασμένων καταστάσεων** (*FSA: finite state acceptor*) για το οποίον

η συνάρτηση μετάβασης είναι της μορφής

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

όπου $\alpha \in \Sigma$ και το Σ το αλφάβητο εισόδου.

Αυτόματα

Αν προσθέσουμε έξοδο σε ένα *FSA*, δηλαδή $\delta : (q_i, a) \to (q_j, b)$, 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 ικανότητες: το αυτόματο ονομάζεται τότε **αυτόματο στοίβας** (*PDA: push-down automaton*). Αν αντί στοίβας έχουμε απεριόριστη δυνατότητα μνήμης υπό μορφή ταινίας, τότε έχουμε τη γνωστή μας **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.

Τυπικές γλώσσες

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).

Μια λέξη ή πρόταση ή συμβολοσειρά ή string είναι μια πεπερασμένου μήκους ακολουθία συμβόλων. Ένα **αλφάβητο** Σ είναι ένα πεπερασμένο σύνολο 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. Άλλοι χρήσιμοι όροι είναι **πρόθεμα** (*prefix*), **υποσυμβολοσειρά** (*substring*), **κατάληξη** (*suffix*), **αντίστροφη** (*reversal*),**παλινδρομική** ή **καρκινική** (*palindrome*).

Τυπικές γλώσσες

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

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

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^*.

Τυπικές γλώσσες

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

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

Τυπικές γραμματικές

Ορισμός 3.2.1. Μια *τυπική γραμματική* G αποτελείται από:

- ένα αλφάβητο V από *μη τερματικά* σύμβολα (μεταβλητές),
- ένα αλφάβητο T από *τερματικά* σύμβολα (σταθερές), τ.ω. $V \cap T = \emptyset$,
- ένα πεπερασμένο σύνολο P από *κανόνες παραγωγής*, δηλαδή διατεταγμένα ζεύγη (α, β) όπου $\alpha, \beta \in (V \cup T)^*$ και $\alpha \neq \varepsilon$
(Σύμβαση: γράφουμε $\alpha \rightarrow \beta$ αντί για (α, β)),
- ένα *αρχικό σύμβολο* (ή αξίωμα) $S \in V$.

Σύμβαση για τη χρήση γραμμάτων:

$$
\begin{aligned}
a, b, c, d, \ldots &\in T \\
A, B, C, D, \ldots &\in V \\
z, y, x, w, v, u, \ldots &\in T^* \\
\alpha, \beta, \gamma, \delta, \ldots &\in (V \cup T)^*
\end{aligned}
$$

Τυπικές γραμματικές

Σύντμηση: Γράφουμε $\alpha \to \beta \mid \gamma \mid \delta$ ως ένα κανόνα στο P αντί για τους τρεις κανόνες $\alpha \to \beta$, $\alpha \to \gamma$, $\alpha \to \delta$ στο P.

Ορισμός 3.2.2.

- Λέμε ότι το $\gamma_1 \alpha \gamma_2$ *παράγει* το $\gamma_1 \beta \gamma_2$ και το συμβολίζουμε με $\gamma_1 \alpha \gamma_2 \Rightarrow \gamma_1 \beta \gamma_2$, αν ο $\alpha \to \beta$ είναι κανόνας παραγωγής (δηλαδή $(\alpha, \beta) \in P$).

- Συμβολίζουμε με $\stackrel{*}{\Rightarrow}$ το ανακλαστικό, μεταβατικό κλείσιμο του $\Rightarrow$, δηλαδή $\alpha \stackrel{*}{\Rightarrow} \beta$ (με λόγια: «το α παράγει το β») σημαίνει ότι υπάρχει μια ακολουθία: $\alpha \Rightarrow \alpha_1 \Rightarrow \alpha_2 \Rightarrow \cdots \alpha_k \Rightarrow \beta$.

- *Ως γλώσσα που παράγεται από τη γραμματική* G ορίζουμε την $L(G) := \{w \in T^* \mid S \stackrel{*}{\Rightarrow} w\}$.

- Δύο γραμματικές G_1, G_2 ονομάζονται ισοδύναμες αν $L(G_1) = L(G_2)$.

Τυπικές γραμματικές

Παράδειγμα 3.2.3. Έστω η γραμματική

$$G\colon V = \{S\}, T = \{a, b\}, P = \{S \to \varepsilon | aSb\}.$$

Μια πιθανή ακολουθία παραγωγής είναι η:

$S \Rightarrow aSb \Rightarrow aaSbb \Rightarrow aaaSbbb \Rightarrow aaabbb$

Η γλώσσα που παράγεται από τη γραμματική είναι η $L(G) = \{a^n b^n \mid n \in N\}$

Ιεραρχία Γραμματικών Chomsky

Ο Noam Chomsky (1956) ταξινόμησε τις τυπικές γραμματικές σε μια ιεραρχία σύμφωνα με τον τύπο των κανόνων παραγωγής τους:

τύπου 0: *γενικές* γραμματικές (general, phrase structure, semi-Thue). Μορφή κανόνων παραγωγής: $\alpha \to \beta$, $\alpha \neq \varepsilon$.

τύπου 1: γραμματικές *με συμφραζόμενα* ή μονοτονικές (context sensitive, monotonic). Μορφή: $\alpha \to \beta$, όπου $|\alpha| < |\beta|$ (μπορεί επιπλέον να επιτρέπεται $S \to \varepsilon$)

τύπου 2: γραμματικές *χωρίς συμφραζόμενα* (context free). Μορφή: $A \to \alpha$, όπου $A \in V$

τύπου 3: *κανονικές* γραμματικές (regular). Η μορφή των κανόνων παραγωγής τους είναι *δεξιογραμμική*: $A \to w$, $A \to wB$ ή *αριστερογραμμική*: $A \to w$, $A \to Bw$, όπου $w \in T^*$, $A, B \in V$.

Ιεραρχία Γραμματικών Chomsky

Όπως θα δούμε στη συνέχεια, αυτή είναι μια γνήσια ιεράρχηση, δηλαδή ισχύει

$$\text{τύπου } 3 \subset \text{τύπου } 2 \subset \text{τύπου } 1 \subset \text{τύπου } 0$$

Οι γλώσσες τύπου 0, 1, 2, 3 μπορούν να αναγνωριστούν από αυτόματα που έχουμε ήδη συζητήσει: από μηχανές Turing (Turing Machines - TM), γραμμικά περιορισμένα αυτόματα (linearly bounded automata - LBA), αυτόματα στοίβας (push down automata - PDA) και αναγνωριστές πεπερασμένων καταστάσεων (finite state acceptors - FSA), αντίστοιχα.

Ορισμός DFA

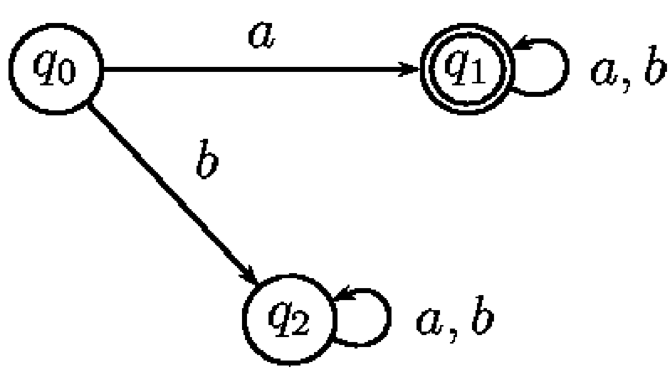
Ορισμός 3.3.1. Τυπικά ένα DFA είναι μία πεντάδα $M = (Q, \Sigma, \delta, q_0, F)$, όπου:

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

Παράδειγμα DFA

Παράδειγμα 3.3.2. $L_1 = \{w \in \Sigma^* \mid w \text{ αρχίζει από } a\}$

Διάγραμμα:



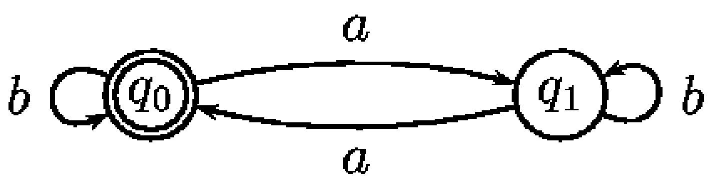
Πίνακας:

	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.

Παράδειγμα γλώσσας με DFA και γλώσσας χωρίς DFA

Παράδειγμα 3.3.3. $L_2 = \{w \in \Sigma^* \mid w$ περιέχει άρτιο πλήθος από $a\}$



	a	b
q_0	q_1	q_0
q_1	q_0	q_1

Παράδειγμα 3.3.4. $L_3 = \{w \in \Sigma^* \mid w$ 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.

Επέκταση ορισμού DFA

Ορισμός 3.3.5. $\tilde{\delta} : Q \times \Sigma^* \to Q$ όπου

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

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

$$\tilde{\delta}(q, a) = \tilde{\delta}(q, \varepsilon a) = \delta(\tilde{\delta}(q, \varepsilon), a) = \delta(q, a)$$

Δηλαδή, για σύμβολα $a \in \Sigma$ η $\tilde{\delta}$ ταυτίζεται με τη δ. Αυτό συμβολίζεται ως

$$\tilde{\delta} \restriction_{Q \times \Sigma} = \delta$$

Γλώσσα αποδεκτή από DFA

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

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

Μη ντετερμινιστικά πεπερασμένα αυτόματα (NFA)

- NFA: *μη ντετερμινιστικό πεπερασμένο αυτόματο.* Σε κάθε μετάβαση υπάρχει επιλογή της επόμενης κατάστασης από ένα σύνολο πιθανών νομίμων καταστάσεων.

- NFA_ε: *μη ντετερμινιστικό πεπερασμένο αυτόματο με ε-κινήσεις.* **Το** πεπερασμένο αυτόματο ενδέχεται να αλλάζει την κατάστασή του χωρίς να μετακινείται η κεφαλή στην ταινία εισόδου.

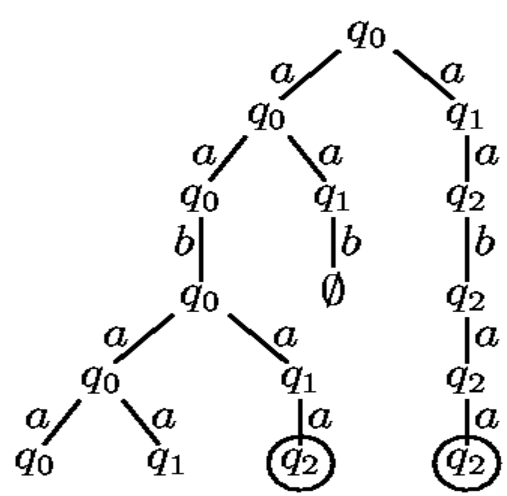
Παράδειγμα NFA

Παράδειγμα 3.3.7. NFA για $L_4 := \{w \in \Sigma^* \mid w \text{ περιέχει δύο συνεχόμενα } a\}$

	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.

Τυπικός ορισμός NFA

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

- 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.

Γλώσσα αποδεκτή από NFA

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

Ισοδυναμία DFA και NFA

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 3.3.13. *(Rabin - Scott)*

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

Ισοδυναμία DFA και NFA

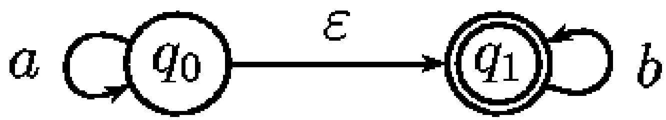
Απόδειξη. Ορίζουμε το DFA $M' = (Q', \Sigma', q_0', F', \delta')$ όπου $Q' = \mathrm{Pow}(Q)$, $\Sigma' = \Sigma$, $q_0' = \{q_0\}$, $F' = \{R \in Q' | R \cap F \neq \emptyset\} = \bigcup_{R \cap F \neq \emptyset} (R \in Q')$ και τέλος $\delta'(R, a) = \delta^{\times}(R, a)$, όπου $R \in Q'$.

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

NFA$_\varepsilon$

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

Παράδειγμα 3.3.14. NFA$_\varepsilon$ για $L_5 := \{a^*b^*\} = \{a^n b^m \mid n, m \in \mathbb{N}\}$



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

Τυπικός ορισμός NFA$_\varepsilon$

- 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.

ε-κλείσιμο

Ορισμός 3.3.15. 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 3.3.16. 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$(\varepsilon$-κλείσιμο$(P)) = \varepsilon$-κλείσιμο(P).

Επέκταση ορισμού NFA$_\varepsilon$

Ορισμός 3.3.17. $\ddot{\delta} : Q \times \Sigma^* \to \mathrm{Pow}(Q)$ όπου

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

Επιπλέον, επεκτείνουμε την $\tilde{\delta}$ στην $\delta^\times$, συμπεριλαμβάνοντας στο πεδίο ορισμού το δυναμοσύνολο του Q.

Ορισμός 3.3.18. $\delta^\times : \mathrm{Pow}(Q) \times \Sigma^* \to \mathrm{Pow}(Q)$ όπου

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

Ισχύει: $\delta^\times \restriction_{Q \times \Sigma^*} = \tilde{\delta}$.

Γλώσσα αποδεκτή από NFA$_\varepsilon$

Ορισμός 3.3.19. Έχουμε:

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

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.

Ισοδυναμία NFA και NFA$_\varepsilon$

Θεώρημα 3.3.20. *Έστω M ένα NFA$_\varepsilon$ τότε* $\exists$ NFA $M' : L(M) = L(M')$

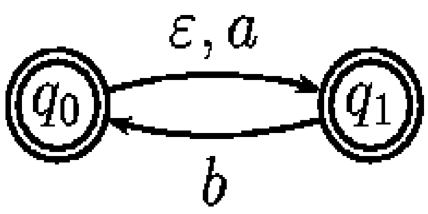
Απόδειξη. Ορίζουμε το NFA $M' = (Q, \Sigma, q_0, F', \delta')$ όπου

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

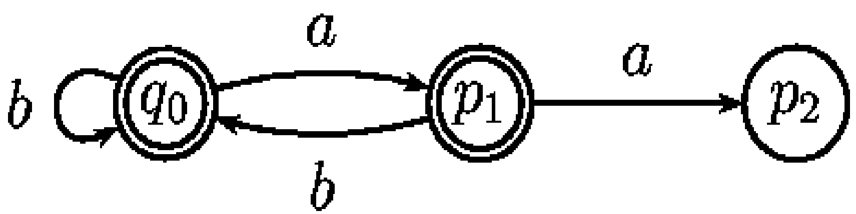
Πλέον, προκειμένου να ισχύει $L(M) = L(M')$, αρκεί να αποδειχθεί ο ισχυρισμός: $\forall w \in \Sigma^* - \{\varepsilon\}$: $\tilde{\delta}'(q_0, w) = \tilde{\delta}(q_0, w)$. (Άσκηση.) $\square$

Παράδειγμα ισοδυναμίας NFA_{ε} και DFA

Παράδειγμα 3.3.22. $\mathrm{NFA}_{\varepsilon}$ για $\overline{L_4}$ (δηλαδή «όχι δύο συνεχόμενα a»):

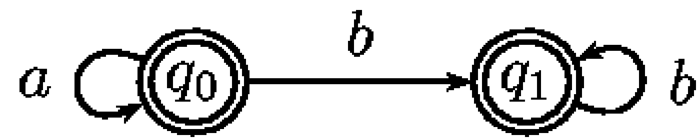


DFA για $\overline{L_4}$:



Παράδειγμα ισοδυναμίας NFA και DFA

Παράδειγμα 3.3.23. NFA για L_5:



DFA για L_5: