

Θεωρία Υπολογισμού  
(Άρτιοι AM)

Π. Ροντογιάννης

Χειμερινό Εξάμηνο 2011-2012

## Περιγραφή μαθήματος

---

Η ύλη του μαθήματος χωρίζεται σε τρεις μεγάλες ενότητες που προσπαθούν να απαντήσουν τα αντίστοιχα ερωτήματα:

- Θεωρία Αυτομάτων (automata theory).  
Τι είναι υπολογιστής;
- Θεωρία υπολογισιμότητας (computability theory).  
Τι είναι και τι δεν είναι υπολογίσιμο;
- Θεωρία υπολογιστικής πολυπλοκότητας (complexity theory). Τι μπορεί να υπολογιστεί γρήγορα και τι όχι;

Πιο γενικά, στο μάθημα αυτό εξετάζεται το ερώτημα: «Ποιές είναι οι δυνατότητες και ποιοί οι περιορισμοί της έννοιας του υπολογιστή;»

# Θεωρία Αυτομάτων

---

- Η Θεωρία Αυτομάτων ασχολείται με τις ιδιότητες μαθηματικών μοντέλων υπολογισμού.
- Ένα αυτόματο είναι μια στοιχειώδης υπολογιστική μηχανή η οποία έχει τη δυνατότητα να αναγνωρίζει τυπικές γλώσσες.
- Όπως θα δούμε, υπάρχουν πολλά διαφορετικά είδη αυτομάτων, με διαφορετικές υπολογιστικές δυνατότητες.
- Τα αυτόματα είναι πολύ χρήσιμα σε διάφορες περιοχές της Πληροφορικής (μεταγλωττιστές, hardware, γλώσσες προγραμματισμού, κλπ).

# Θεωρία Υπολογισιμότητας

---

- Η Θεωρία Υπολογισιμότητας εξετάζει ποια προβλήματα μπορούν να επιλυθούν από υπολογιστή και ποια όχι.
- Μας βοηθάει να καταλάβουμε ποια είναι τα όρια της επιστήμης μας (στα περισσότερα μαθήματα Πληροφορικής μαθαίνουμε τι μπορούμε να κάνουμε με τους υπολογιστές, και όχι τι δεν μπορούμε).
- Μας προσφέρει τα μαθηματικά εργαλεία για να αποδείξουμε ότι ένα πρόβλημα δεν είναι επιλύσιμο.
- Η Θεωρία Υπολογισιμότητας έχει πολύ στενές σχέσεις με τη Μαθηματική Λογική.

# Θεωρία Υπολογιστικής Πολυπλοκότητας

---

- Η Θεωρία Πολυπλοκότητας συγγενεύει στενά με τη Θεωρία Υπολογισιμότητας.
- Στη Θεωρία Πολυπλοκότητας, το ζητούμενο είναι να κατατάξουμε τα προβλήματα ως «εύκολα» ή «δύσκολα» (ενώ στη Θεωρία Υπολογισιμότητας ως «επιλύσιμα» ή «μη επιλύσιμα»).
- Υπάρχουν προβλήματα για τα οποία δεν έχει καταφέρει κανείς να βρει αποτελεσματικό αλγόριθμο επίλυσης (για τα προβλήματα αυτά υπάρχουν αλγόριθμοι οι οποίοι είναι απελπιστικά αναποτελεσματικοί).
- Ένα κεντρικό πρόβλημα της περιοχής είναι το γνωστό μας «P vs NP». Η περιοχή της Υπολογιστικής Πολυπλοκότητας έχει να επιδείξει πολλά ενδιαφέροντα, όμορφα αλλά δυστυχώς ανοιχτά προβλήματα.

## Κάποια πρόσωπα της ιστορίας μας

---

- Georg Cantor (Μελέτη της Θεωρίας Συνόλων)
- David Hilbert (Σχολή του «Φορμαλισμού»)
- Kurt Gödel (Θεώρημα της μη πληρότητας)
- Alonzo Church (Θεμελίωση Θεωρητικής Πληροφορικής)
- Alan Turing (Μηχανή Turing)
- Noam Chomsky (Ιεραρχία τυπικών γλωσσών)

## Συμβολοσειρές και γλώσσες

---

**Αλφάβητο:** Αλφάβητο είναι κάθε πεπερασμένο μη κενό σύνολο. Τα μέλη του τα ονομάζουμε σύμβολα ή γράμματα.

Π.χ.  $\Sigma_1 = \{0, 1\}$ ,  $\Sigma_2 = \{a, b, \dots, w\}$ .

**Συμβολοσειρά (string):** Συμβολοσειρά ενός αλφάβητου  $\Sigma$  είναι μια πεπερασμένη ακολουθία συμβόλων του  $\Sigma$ .

Π.χ. 0101101 είναι συμβολοσειρά του αλφάβητου  $\Sigma = \{0, 1, 2\}$ .

Τη (μοναδική) συμβολοσειρά μήκους 0, την ονομάζουμε κενή και τη συμβολίζουμε με  $\varepsilon$ . Το σύνολο των συμβολοσειρών μήκους  $k$  το συμβολίζουμε με  $\Sigma^k$ , π.χ.  $\{0, 1\}^2 = \{00, 01, 10, 11\}$ .

Το σύνολο όλων των συμβολοσειρών του  $\Sigma$  το συμβολίζουμε με  $\Sigma^*$ . Π.χ.  $\{0, 1\}^* = \{\varepsilon, 0, 1, 00, 01, 10, 11, 000, \dots\}$ .

## Πράξεις με συμβολοσειρές

---

**Παράθεση (concatenation):** Η παράθεση δύο συμβολοσειρών  $x$  και  $y$  είναι η συμβολοσειρά  $xy$  που τη συμβολίζουμε  $xy$  ή  $x \circ y$ .

Π.χ. Η παράθεση του  $x = 011$  και του  $y = 1001$  είναι  $x \circ y = 0111001$ .

**Επανάληψη:** Αν  $w$  είναι μια συμβολοσειρά, τότε  $w^k$  αποτελείται από την παράθεση  $k$  αντιγράφων του  $w$ .

Π.χ.  $(01)^3 = 010101$ .

**Αντίστροφη:** Η αντίστροφη μιάς συμβολοσειράς  $w$  συμβολίζεται με  $w^R$  και προκύπτει αν διαβάσουμε το  $w$  από το τέλος προς την αρχή.

Π.χ.  $01011^R = 11010$ .

● **Άσκηση:** Δείξε ότι για οποιεσδήποτε συμβολοσειρές  $x$  και  $y$ :  $(x \circ y)^R = y^R \circ x^R$ .

# Γλώσσες (Languages)

---

**Γλώσσα:** Έστω  $\Sigma$  ένα αλφάβητο. Οποιοδήποτε υποσύνολο του  $\Sigma^*$  ονομάζεται γλώσσα του  $\Sigma$ .

Π.χ. Έστω  $\Sigma = \{0, 1, \dots, 9\}$ . Τα παρακάτω σύνολα αποτελούν γλώσσες του  $\Sigma$ :

- $L_1 = \{23, 044, 9999\}$  (πεπερασμένη γλώσσα).
- $L_2 = \{\varepsilon, 1, 11, 111, 1111, \dots\}$ .
- $L_3 = \{w : \text{η δεκαδική αναπαράσταση του } w \text{ είναι πρώτος αριθμός}\} = \{2, 3, 5, 7, 11, 13, \dots\}$ .
- $L_4 = \{w : \text{η δυαδική αναπαράσταση του } w \text{ είναι πρώτος αριθμός}\} = \{10, 11, 101, 111, 1011, 1101, \dots\}$ .
- $L_5 = \{\} = \emptyset$  (κενή γλώσσα).
- $L_6 = \{\varepsilon\}$ .
- $L_7 = \{w : w \text{ είναι πρόγραμμα της C++ χωρίς input που δεν τερματίζει ποτέ (κωδικοποιημένο στο δυαδικό σύστημα)}\}$ .

## Πράξεις με γλώσσες

---

Αφού οι γλώσσες είναι σύνολα, ορίζεται η **ένωση**  $L_1 \cup L_2$  και η **τομή** τους  $L_1 \cap L_2$  όπως και το συμπλήρωμα:

**Συμπλήρωμα:** Το συμπλήρωμα μιας γλώσσας  $L$  του αλφαβήτου  $\Sigma$  συμβολίζεται με  $\bar{L}$  και είναι η γλώσσα  $\Sigma^* - L$  που αποτελείται από τις συμβολοσειρές του  $\Sigma^*$  που δεν ανήκουν στην  $L$ .

Επιπλέον μπορούμε να ορίσουμε τις ακόλουθες πράξεις σε γλώσσες:

**Παράθεση:** Αν  $L_1$  και  $L_2$  είναι δυο γλώσσες του αλφαβήτου  $\Sigma$ , τότε η παράθεσή τους συμβολίζεται  $L_1 \circ L_2$  ή  $L_1 L_2$  και ορίζεται σαν  $L_1 \circ L_2 = \{w : w = xy \text{ για κάποιο } x \in L_1 \text{ και κάποιο } y \in L_2 \}$ .

Π.χ. αν  $L_1 = \{0, 1, 00\}$  και  $L_2 = \{\varepsilon, 00\}$  τότε  $L_1 \circ L_2 = \{0, 1, 00, 000, 100, 0000\}$ .

**Kleene star:** Η Kleene star  $L^*$  μια γλώσσας  $L$  είναι η γλώσσα των συμβολοσειρών που προκύπτουν από παράθεση μηδέν ή περισσότερων συμβολοσειρών της  $L$ :

$$L^* = \{w : w = w_1 \circ w_2 \circ \dots \circ w_n \text{ για } n \geq 0 \text{ και } w_1, \dots, w_n \in L\}.$$

Π.χ. Αν  $L = \{0, 11\}$  τότε

$$L^* = \{\varepsilon, 0, 00, 11, 000, 011, 110, 0000, 0011, 0110, 1100, 1111, \dots\}.$$

Π.χ. Αν  $L = \{\varepsilon\}$  τότε  $L^* = \{\varepsilon\}$ .

Π.χ. Αν  $L = \{\}$  τότε  $L^* = \{\varepsilon\}$  (άρα για κάθε  $L$ :  $\varepsilon \in L^*$ ).

**$L^+$ :** Ορίζουμε επίσης  $L^+ = LL^*$ .

## Πόσες συμβολοσειρές και γλώσσες υπάρχουν;

---

Θεώρησε ένα αλφάβητο  $\Sigma$  (εξ ορισμού πεπερασμένο).

Πόσες συμβολοσειρές του  $\Sigma$  υπάρχουν; Άπειρες.

Πόσες γλώσσες του  $\Sigma$  υπάρχουν; Άπειρες.

Αλλά υπάρχουν πολλών ειδών «άπειρα».

**Ισάριθμα σύνολα:** Δύο σύνολα  $A$  και  $B$  λέγονται ισάριθμα αν υπάρχει αμφιμονοσήμαντη αντιστοιχία  $f : A \rightarrow B$ .

**Πεπερασμένα σύνολα:** Ένα σύνολο είναι πεπερασμένο αν είναι ισάριθμο με το  $\{1, 2, \dots, n\}$ , όπου  $n$  κάποιος φυσικός αριθμός.

**Αριθμήσιμα σύνολα:** Ένα σύνολο λέγεται αριθμήσιμα άπειρο αν είναι ισάριθμο του  $\mathbb{N}$ , και αριθμήσιμο αν είναι πεπερασμένο ή αριθμήσιμα άπειρο.

## Μετρώντας (συνέχεια)

---

Παραδείγματα αριθμήσιμων συνόλων:

- Το σύνολο των ζυγών αριθμών (αντιστοιχία:  $f(n) = 2n$ . Ακολουθία:  $0, 2, 4, \dots$ ).
- Το σύνολο των ακεραίων (Ακολουθία:  $0, 1, -1, 2, -2, 3, -3, \dots$ ).
- Το σύνολο  $N \times N$   
(Ακολουθία:  $(0, 0), (0, 1), (1, 0), (0, 2), (1, 1), (2, 1), \dots$ ).

**Θεώρημα:** Το σύνολο  $\Sigma^*$  των συμβολοσειρών ενός αλφαβήτου  $\Sigma$  είναι αριθμήσιμο.

**Απόδειξη:** Εξ' ορισμού το αλφάβητο  $\Sigma$  είναι πεπερασμένο. Μπορούμε τότε να δημιουργήσουμε μια ακολουθία που περιέχει όλες τις συμβολοσειρές του  $\Sigma$  σε λεξικογραφική σειρά ως εξής:

Πρώτη η συμβολοσειρά μήκους 0, μετά όλες οι συμβολοσειρές μήκους 1 (ταξινομημένες), μετά όλες οι συμβολοσειρές μήκους 2 (ταξινομημένες), κοκ. Την ακολουθία αυτή την ονομάζουμε **λεξικογραφική ακολουθία** των συμβολοσειρών του  $\Sigma$ . Για παράδειγμα, αν  $\Sigma = \{0, 1\}$ , τότε η ακολουθία είναι  $\varepsilon, 0, 1, 00, 01, 10, 11, 000, \dots$

Η παραπάνω ακολουθία ορίζει μια αμφιμονοσήμαντη αντιστοιχία.

**Θεώρημα:** Το σύνολο των γλωσσών ενός αλφαβήτου  $\Sigma$  είναι μη αριθμήσιμο.

**Απόδειξη:** Με εις άτοπο απαγωγή. Έστω ότι το σύνολο των γλωσσών του  $\Sigma$  είναι αριθμήσιμο. Τότε θα μπορούμε να απαριθμήσουμε τις γλώσσες του  $\Sigma$  ως μια ακολουθία  $S_1, S_2, \dots$ . Θα κατασκευάσουμε γλώσσα  $T$  του  $\Sigma$  που διαφέρει από κάθε  $S_i$ ,  $i = 1, 2, \dots$ .

Έστω  $w_1, w_2, \dots$  η λεξικογραφική ακολουθία των συμβολοσειρών του  $\Sigma$ , όπου κάθε συμβολοσειρά του  $\Sigma$  εμφανίζεται ακριβώς μια φορά σε αυτή. Θα φροντίσουμε ώστε η  $T$  να διαφέρει από την  $S_i$  στη θέση του  $w_i$ . Η  $T$  **περιέχει** τη συμβολοσειρά  $w_i$  αν και μόνο αν η  $S_i$  **δεν περιέχει** τη συμβολοσειρά  $w_i$ .

$$T = \{w_i : w_i \notin S_i, i = 1, 2, \dots\}.$$

Συνεπώς η ακολουθία  $S_1, S_2, \dots$  δεν περιέχει όλες τις γλώσσες του  $\Sigma$ . Άρα το σύνολο των γλωσσών του  $\Sigma$  είναι μη αριθμήσιμο.

Συμπεράσματα για την αναπαράσταση των γλωσσών με πεπερασμένες περιγραφές:

- Οποιαδήποτε αναπαράσταση και αν διαλέξουμε, θα υπάρχουν γλώσσες που δεν περιγράφονται.
- Ειδικότερα, υπάρχει κάποια γλώσσα  $L$  τέτοια ώστε κανένα πρόγραμμα C++ δεν μπορεί να τυπώσει όλες τις συμβολοσειρές της (ακόμα και αν το αφήσουμε να τρέχει για πάντα). Γιατί; Απάντηση: Το σύνολο των προγραμμάτων είναι αριθμήσιμο, ενώ το σύνολο των γλωσσών δεν είναι.

Οι γλώσσες που μας ενδιαφέρουν είναι αυτές που μπορεί να περιγραφτούν (αυτές που έχουν πεπερασμένη περιγραφή). Το ερώτημα είναι αν υπάρχει πρόγραμμα C++ για αυτές τις γλώσσες. Αυτό είναι το κεντρικό θέμα του μαθήματος. Για να το μελετήσουμε πρέπει πρώτα να συμφωνήσουμε για το τι εννοούμε με τον όρο «πεπερασμένη περιγραφή».

## Μέθοδος Διαγωνοποίησης

---

Πώς αποδείξαμε ότι το σύνολο των γλωσσών δεν είναι αριθμήσιμο; Η μέθοδος που χρησιμοποιήσαμε λέγεται διαγωνοποίηση και προτάθηκε από τον G. Cantor (1891).

**Μέθοδος Διαγωνοποίησης:** Έστω  $R$  μια διμελής σχέση ενός συνόλου  $A$ , δηλαδή  $R$  είναι ένα υποσύνολο του Καρτεσιανού γινομένου  $A \times A$ :  $R \subseteq \{(a_1, a_2) : a_1, a_2 \in A\}$ . Τότε το διαγώνιο σύνολο  $\Delta = \{a : (a, a) \notin R\}$  διαφέρει από κάθε γραμμή  $R_a = \{b : (a, b) \in R\}$

## Παράδειγμα:

Έστω  $A = \{1, 2, 3, 4, 5\}$  και έστω η σχέση

$$R = \{(1, 3), (2, 2), (2, 4), (2, 5), (3, 3), (3, 5), (4, 2), (4, 5), (5, 1), (5, 2), (5, 3), (5, 5)\}$$

	1	2	3	4	5
1	$O$		$X$		
2		$X$		$X$	$X$
3			$X$		$X$
4		$X$		$O$	$X$
5	$X$	$X$	$X$		$X$

Οι γραμμές της  $R$  είναι  $R_1 = \{3\}$ ,  $R_2 = \{2, 4, 5\}$ ,  $R_3 = \{3, 5\}$ ,  $R_4 = \{2, 5\}$ , και  $R_5 = \{1, 2, 3, 5\}$ .

Η διαγώνιος είναι το σύνολο  $\Delta = \{1, 4\}$  και είναι διαφορετικό απο κάθε γραμμή της  $R$ .

## Πραγματικοί αριθμοί

---

Ο G. Cantor χρησιμοποίησε τη διαγωνιοποίηση για να δείξει ότι το σύνολο των πραγματικών αριθμών  $\mathcal{R}$  είναι «μεγαλύτερο» από το σύνολο των φυσικών αριθμών  $\mathcal{N}$ . Η απόδειξη μπορεί να σκιαγραφηθεί ως εξής:

Κάθε πραγματικός μπορεί να γραφτεί στο δεκαδικό σύστημα. Ας υποθέσουμε ότι μπορούμε να τους μετρήσουμε (δηλαδή να τους βάλουμε σε σειρά) π.χ.

0.**0**000000...

0.0**0**10000...

0.01**1**0030...

0.080**5**000...

⋮

Ας κατασκευάσουμε τώρα έναν αριθμο που διαφέρει απο τον πρώτο στο πρώτο δεκαδικό ψηφίο, απο το δεύτερο στο δεύτερο δεκαδικό ψηφίο κοκ. Ένας τέτοιος αριθμός είναι για παράδειγμα ο  $0.1126\dots$ , που διαφέρει από όλους τους αριθμούς.

## Η Υπόθεση του Συνεχούς

---

Παρεμπιπτόντως, ένα από τα μεγαλύτερα προβλήματα της Λογικής στον 20ο αιώνα είναι η «Υπόθεση του Συνεχούς»:

**Υπόθεση Συνεχούς:** Δεν υπάρχει κανένα σύνολο «μεγαλύτερο» από τους φυσικούς  $\mathcal{N}$  και «μικρότερο» από τους πραγματικούς  $\mathcal{R}$ . Με άλλα λόγια, κάθε μη αριθμήσιμο σύνολο περιέχει ένα υποσύνολο ισοδύναμο με το  $\mathcal{R}$ .

- Ο Gödel έδειξε το 1937 ότι η Υπόθεση του Συνεχούς είναι συμβατή με τα αξιώματα της συνολοθεωρίας (άρα δεν υπάρχει απόδειξη ότι η υπόθεση δεν ισχύει).
- Ο Cohen έδειξε το 1963 ότι η άρνηση της Υπόθεσης του Συνεχούς είναι επίσης συμβατή με τα αξιώματα της συνολοθεωρίας (άρα δεν υπάρχει απόδειξη ότι η υπόθεση ισχύει).

Συνεπώς η υπόθεση του συνεχούς δεν μπορεί να αποδειχτεί !