Περιγραφή μαθήματος

Σκοπός του μαθήματος είναι η εισαγωγή στη Θεωρία Υπολογισμού και στη Θεωρία Υπολογιστικής Πολυπλοκότητας (Θεωρία Αλγορίθμων).

Τι είναι πολυπλοκότητα;

Υπάρχουν πολλών ειδών πολυπλοκότητες, π.χ.

- Τα θηλαστικά είναι πιο πολύπλοκα από τα βακτήρια.
- Το σκάκι είναι πιο πολύπλοκο από την τρίλιζα.
- Το σύνολο των πρώτων αριθμών είναι πιο πολύπλοκο από το σύνολο των αρτίων αριθμών.

Η τελευταία περίπτωση είναι που μας ενδιαφέρει κυρίως εδώ.

Πρώτοι αριθμοί - Άρτιοι Αριθμοί

Ένα μέσο για να συλλάβουμε την πολυπλοκότητα των πρώτων αριθμών είναι η υπολογιστική πολυπλοκότητα (**computational complexity**). Κατά κάποιο τρόπο (αλλά όχι και μοναδικό) οι πρώτοι αριθμοί είναι πιο πολύπλοκοι από τους αρτιους γιατί το ερώτημα «Είναι ο ξ πρώτος;» είναι πιο δύσκολο από το ερώτημα «Είναι ο ξ άρτιος;».

Αλλά τι σημαίνει «πιο δύσκολο ερώτημα»; Ένα από τα μεγαλύτερα επιτεύγματα των Μαθηματικών και της Πληροφορικής τον τελευταίο αιώνα είναι πως κατάφερε να κάνει αυτό το ερώτημα αυστηρό και ακριβές και παράλληλα να το απαντήσει στη γενικότητα του.

Τι είναι πολυπλοκότητα;

Ταυτόχρονα, κατά ειρωνικό τρόπο, ένα από τα μεγαλύτερα άλυτα μαθηματικά προβλήματα σήμερα είναι το «P=NP»[1], ένα πρόβλημα πού γεννήθηκε μέσα από τις επιτυχίες της υπολογιστικής πολυπλοκότητας.

[1] www.claymath.org

Η έννοια «πρόβλημα»

Τι είναι πρόβλημα;

1η Περίπτωση: Υπάρχουν ακέραιοι $x, y, z \geq 1$ και $n > 2$ τέτοιοι ώστε $x^n + y^n = z^n$;

2η Περίπτωση: Γράψτε ένα πρόγραμμα **Pascal** που όταν η είσοδος είναι ένα ακέραιο πολυώνυμο p (πολλών μεταβλητών), στην έξοδο απαντά 'ναι' αν το πολυώνυμο p έχει ακέραιες ρίζες και 'οχι' διαφορετικά;

Υπάρχει μιά βασική διαφορά ανάμεσα στις δύο περιπτώσεις: Στην πρώτη περίπτωση η απάντηση είναι «ναι» ή «όχι», ενώ στη δεύτερη περίπτωση η απάντηση είναι ένα πρόγραμμα **Pascal** (αλγόριθμος).

Η έννοια «πρόβλημα»

Η πρώτη περίπτωση ($x^n + y^n = z^n$) είναι το περίφημο Θεώρημα του **Fermat** που προτάθηκε από τον **Fermat** και παρέμεινε άλυτο για τετρακόσια περίπου χρόνια, ώς το 1994, όταν ο **Andrew Wiles** κατάφερε να το λύσει (έδειξε ότι η σωστή απάντηση στην ερώτηση είναι αρνητική).

Το δεύτερο πρόβλημα (πρόγραμμα που να λύνει ακέραιες εξισώσεις) προτάθηκε από τον **David Hilbert** το 1900, είναι το περίφημο δέκατο πρόβλημα του **Hilbert**, και απαντήθηκε απο τον **Yuri Matiyasevich** το 1970, που έδειξε ότι δεν υπάρχει τέτοιο πρόγραμμα **Pascal.**

Μας ενδιαφέρουν κυρίως προβλήματα της δεύτερης κατηγορίας, όπου το ζητούμενο είναι ένας αλγόριθμος και όχι ένα απλό «ναι» ή «όχι».

Η έννοια «πρόβλημα» ΙΙ

Τι είναι πρόβλημα;

1η Περίπτωση: Αποδείξτε πως δεν υπάρχουν ακέραιοι $x, y, z \geq 1$ και $n > 2$ τέτοιοι ώστε $x^n + y^n = z^n$.

2η Περίπτωση: Γράψτε ένα πρόγραμμα Pascal που όταν η είσοδος είναι ένα ακέραιο πολυώνυμο p (πολλών μεταβλητών), στην έξοδο απαντά αν το πολυώνυμο p έχει ακέραιες ρίζες.

Υπάρχει μιά βασική διαφορά ανάμεσα στις δύο περιπτώσεις: Στην πρώτη περίπτωση η απάντηση είναι μια οποιαδήποτε φορμαλιστικά σωστή μαθηματική απόδειξη, ενώ στη δεύτερη το πρόγραμμα καθώς υπολογίζει αποδεικνύει την ύπαρξη ή μη ριζών για το p.

Το ίδιο το πρόγραμμα είναι μία «αυτοματοποιημένη» μορφή απόδειξης.

Κάποια πρόσωπα της ιστορίας μας

David Hilbert: Έθεσε (μεταξύ άλλων) τα ερωτήματα

1. αν μπορεί να αποδειχτεί πως τα Μαθηματικά είναι συνεπή (**consistent**) (1900).
2. αν τα Μαθηματικά μπορούν να «αυτοματοποιηθούν» (**Entscheidungsproblem**, 1928).

Kurt Gödel: Εδειξε ότι το (1) είναι αδύνατο με το περίφημο «Θεώρημα της μη πληρότητας» (1930).

Alonzo Church: Έδειξε ότι το (2) είναι αδύνατο και τυποποίησε την έννοια της υπολογισιμότητας.

Alan Turing: Έθεσε τις βάσεις της Θεωρίας Υπολογισμού. Όρισε την έννοια του υπολογιστή (μηχανές **Turing**) και μελέτησε την έννοια της υπολογισιμότητας (1936). Έδειξε (σχεδόν ταυτόχρονα με τον **Church**) πως ουσιαστικά το (2) είναι αδύνατο.

Noam Chomsky: Ανέλυσε τήν ιεραρχία των γλωσσών και γραμματικών (ιεραρχία **Chomsky**, 1957).

Ύλη του μαθήματος

Η ύλη του μαθήματος χωρίζεται σε 3 μεγάλες ενότητες που προσπαθούν να απαντήσουν τα αντίστοιχα ερωτήματα:

- **Μοντέλα υπολογισμού. Αυτόματα και γλώσσες.**
  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;

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

Π.χ. $\Sigma_1 = \{0, 1\}$, $\Sigma_2 = \{a, b, \ldots, w\}$. **Συμβολοσειρά (string):** Συμβολοσειρά ενός αλφάβητου Σ είναι μια πεπερασμένη ακολουθία συμβόλων του Σ.

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

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

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

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. $01011^R = 11010$.

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

Γλώσσες (Languages)

Γλώσσα: Έστω Σ ένα αλφάβητο. Οποιοδήποτε υποσύνολο του Σ^* ονομάζεται γλώσσα του Σ.

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

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

- $L_8 = \{w : w$ είναι πρόγραμμα της C++ χωρίς **input** που κάποτε τελειώνει (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 Σ συμβολίζεται με $\bar{L}$ και είναι η γλώσσα $\Sigma^* - L$ που αποτελείται από τις συμβολοσειρές του Σ που δεν ανήκουν στην L.

Επιπλέον μπορούμε να ορίσουμε τις παρακάτω πράξεις σε γλώσσες:

Παράθεση: Αν L_1 και L_2 είναι δυο γλώσσες του αλφαβήτου Σ, τότε η παράθεσή τους συμβολίζεται $L_1 \circ L_2$ ή $L_1 L_2$ και ορίζεται σαν $L_1 \circ L_2 = \{w : w = xy$ για κάποιο $x \in L_1$ 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 περισσοτέρων συμβολοσειρών της L:

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

Π.χ. Αν $L = \{0, 11\}$ τότε
$L^* = \{\varepsilon, 0, 00, 11, 000, 011, 110, 0000, 0011, 0110, 1100, 1111, \ldots\}$.

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 Σ (εξ ορισμού πεπερασμένο).

Πόσες συμβολοσειρές του Σ υπάρχουν; Άπειρες.

Πόσες γλώσσες του Σ υπάρχουν; Άπειρες.

Αλλά υπάρχουν πολλών ειδών «άπειρα».

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

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

Μετρήσιμα σύνολα: Ένα σύνολο λέγεται μετρήσιμα άπειρο αν είναι ισάριθμο του N, και μετρήσιμο αν είναι πεπερασμένο ή μετρήσιμα άπειρο.

Μετρώντας (συνέχεια)

Παραδείγματα μετρήσιμων συνόλων:

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

Θεώρημα: Το σύνολο Σ^* των συμβολοσειρών ενός αλφαβήτου Σ είναι μετρήσιμο.

Απόδειξη: Εξ΄ ορισμού το αλφάβητο Σ είναι πεπερασμένο. Μπορούμε τότε να δημιουργήσουμε μια ακολουθία που περιέχει όλες τις συμβολοσειρές του Σ 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 θα την ονομάζουμε λεξικογραφική ακολουθία των συμβολοσειρών του Σ. Π.χ. Αν $\Sigma = \{0, 1\}$, η ακολουθία είναι $\varepsilon, 0, 1, 00, 01, 10, 11, 000, \ldots$.

Θεώρημα: Το σύνολο των γλωσσών ενός αλφαβήτου Σ είναι μη μετρήσιμο.

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

Θεώρημα: Το σύνολο των γλωσσών ενός αλφαβήτου Σ είναι μη μετρήσιμο.

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

Έστω $w_1, w_2, \ldots$ η λεξικογραφική ακολουθία των συμβολοσειρών του Σ, όπου κάθε συμβολοσειρά του Σ 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.

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

Συνεπώς η ακολουθία $S_1, S_2, \ldots$ δεν περιέχει όλες τις γλώσσες του Σ. Άρα το σύνολο των γλωσσών του Σ είναι μη μετρήσιμο.

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 μετρήσιμο, 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 ειναι αν υπάρχει πρόγραμμα C++ για αυτές τις γλωσσες. 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 μετρήσιμο; Η μέθοδος που χρησιμοποιήσαμε λέγεται διαγωνοποίηση και είναι πολύ απλή.

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 $R_a = \{b : \ (a, b) \in R\}$

Π.χ. $A = \{1, 2, 3, 4, 5\}$ **και**
$R = \{(1, 3), (2, 2), (2, 4), (2, 6), (3, 3), (3, 5), (4, 2), (4, 5), (5, 1), (5, 2), (5, 3), (5, 5)\}$

	1	2	3	4	5
1			X		X
2		X		X	X
3			X		X
4		X			X
5	X	X	X		X

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

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

Την μέθοδο διαγωνοποίησης την εισήγαγε ο **Georg Cantor** στα 1891. Την χρησιμοποίησε για να δείξει ότι το σύνολο των πραγματικών αριθμών $\mathcal{R}$ είναι «μεγαλύτερο» από το σύνολο των φυσικών αριθμών $\mathcal{N}$.

Η απόδειξη είναι κάπως έτσι: Κάθε πραγματικός μπορεί να γραφτεί στο δεκαδικό σύστημα. Ας υποθέσουμε οτι μπορούμε να τους μετρήσουμε (δηλαδή να τους βάλουμε σε σειρά) π.χ.

$$
\begin{aligned}
&0.\mathbf{0}000000\ldots\\
&0.0\mathbf{0}10000\ldots\\
&0.01\mathbf{1}0030\ldots\\
&0.080\mathbf{5}000\ldots\\
&\vdots
\end{aligned}
$$

Ας κατασκευάσουμε τώρα έναν αριθμο που διαφέρει απο τον πρώτο στο πρώτο δεκαδικό ψηφίο, απο το δεύτερο στο δεύτερο δεκαδικό ψηφίο κοκ. Π.χ., για το παραπάνω παράδειγμα μπορούμε να

κατασκευάσουμε τον

$$0.1126\ldots$$

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 !!!