# Εισαγωγή στην Υπολογιστική Πολυπλοκότητα

**Δημήτρης Φωτάκης**

Τμήμα Μηχανικών Πληροφοριακών και Επικοινωνιακών Συστημάτων

Πανεπιστήμιο Αιγαίου

## Θεωρία Υπολογισμού

- Γιατί κάποια προβλήματα είναι **αδύνατο** να λυθούν από υπολογιστές;
- **Hilbert** (1900): **πληρότητα** και **αυτοματοποίηση** των μαθηματικών.
- 10ο πρόβλημα : **Αλγόριθμος** για λύση Διοφαντικών εξισώσεων: Έχει $x^2 - 2y^2 + 3 = 0$ **ακέραιες** ρίζες;
- **Αλγόριθμος**: Διατύπωση και απόδειξη.
- **Όχι αλγόριθμος**: Ορισμός "αλγόριθμου" μέσω **υπολογιστικού μοντέλου**. Απόδειξη ότι "αλγόριθμος $\Rightarrow$ αντίφαση στο μοντέλο".
- **Gödel**: Μαθηματικά **δεν είναι πλήρη**! Υπάρχουν "αλήθειες" που δεν αποδεικνύονται.
- **Turing**: Μαθηματικά **δεν αυτοματοποιούνται**! Μη-επιλύσιμα προβλήματα: επίλυσή τους δεν αυτοματοποιείται (γενική περίπτωση).
- **Matijasevic** (1970): Δεν υπάρχει αλγόριθμος για Διοφαντικές εξισώσεις! Για κάθε αλγόριθμο $\mathcal{A}$, υπάρχει εξίσωση που ο $\mathcal{A}$ δίνει **λάθος απάντηση**!



## Υπολογιστική Πολυπλοκότητα

- Γιατί κάποια προβλήματα είναι **δύσκολο** να λυθούν από υπολογιστές; Ανάπτυξη τελευταία 30 χρόνια (Papadimitriou, Computational Complexity, 1994).
- Ποια **επιλύσιμα** προβλήματα είναι **εύκολα** και ποια **δύσκολα**.
- **Επιλύσιμα** προβλήματα: Υπολογιστικοί πόροι;
  - Εύλογοι υπολογιστικοί πόροι $\Rightarrow$ **ευεπίλυτα** (tractable) προβλήματα. Πολλαπλασιασμός Πινάκων, Κλασματικό Σακίδιο, Ελάχιστο Επικαλύπτον Δέντρο, Συντομότερα Μονοπάτια.
  - Διαφορετικά, **δυσεπίλυτα** (intractable). Ακέραιο Σακίδιο, Περιοδεύων Πωλητής, Κάλυψη Συνόλων, Δρομολόγηση, Χρωματισμός, Συντομότερα Μονοπάτια με Περιορισμούς, κλπ.
  - Επίδραση **υπολογιστικού μοντέλου** στους υπολογιστικούς πόρους.



## Προβλήματα και Αλγόριθμοι

- **Αλγόριθμος** είναι λεπτομερής περιγραφή μεθόδου επίλυσης προβλήματος. υπολογιστική μηχανή (Turing) που τερματίζει.
- **Υπολογιστικό πρόβλημα** αποτελείται από άπειρο σύνολο στιγμιοτύπων. αποτελεί αντικείμενο μελέτης.
- **Στιγμιότυπο** είναι μαθηματικό αντικείμενο για το οποίο ρωτάμε **ερώτηση** και περιμένουμε **απάντηση**.
- Δύο είδη προβλημάτων:
  - **Απόφασης**: απαντήσεις **ΝΑΙ** ή **ΟΧΙ**.
  - **Βελτιστοποίησης**: καλύτερη εφικτή **λύση**.

## Παραδείγματα Προβλημάτων

- **Πρόβλημα Προσπελασιμότητας**:
  - **Στιγμιότυπο**: Κατευθυνόμενο γράφημα $G(V, E)$ και διακεκριμένες κορυφές $s$ και $t$.
  - **Ερώτηση**: Υπάρχει μονοπάτι από $s$ στο $t$;
- **Πρόβλημα Συντομότερου Μονοπατιού**:
  - **Στιγμιότυπο**: Κατευθυνόμενο γράφημα με μήκη στις ακμές $G(V, E, w)$ και διακεκριμένες κορυφές $s$ και $t$.
  - **Ερώτηση**: Ποιο είναι το συντομότερο $s - t$ μονοπάτι;



## Παραδείγματα Προβλημάτων

- **Πρόβλημα κύκλου Hamilton**:
  - **Στιγμιότυπο**: Γράφημα $G(V, E)$.
  - **Ερώτηση**: Υπάρχει κύκλος Hamilton στο $G$ (κύκλος που διέρχεται από κάθε κορυφή ακριβώς μία φορά);
- **Πρόβλημα Περιοδεύοντος Πωλητή**:
  - **Στιγμιότυπο**: Σύνολο $= \{1, \dots, n\}$ σημείων και αποστάσεις $d(i, j)$ μεταξύ κάθε ζεύγους διαφορετικών σημείων.
  - **Ερώτηση**: Ποια μετάθεση $\pi$ του $N$ ελαχιστοποιεί

$$d(\pi(n), \pi(1)) + \sum_{i=1}^{n-1} d(\pi(i), \pi(i+1))$$



## Προσέγγιση

- **Κλάσεις προβλημάτων** (complexity classes) με παρόμοια "δυσκολία" (υπολογιστική πολυπλοκότητα).
- **Αναγωγή σε πλήρη** (complete) προβλήματα κάθε κλάσης: Συνοψίζουν **δυσκολία** της κλάσης.
- **Πλήρες** πρόβλημα "εύκολο" $\Rightarrow$ **´Ολη** η κλάση "εύκολη".
- **Αρνητικά** αποτελέσματα $\Rightarrow$ **´Ολη** η κλάση "δύσκολη".
- Προσδιορισμός **υπολογιστικού έργου** για λύση προβλημάτων στην κλάση (με παραδειγματικά υπολογιστικά προβλήματα)!
- Διαλεκτική σχέση **αλγόριθμων** και **πολυπλοκότητας**.



## Κωδικοποίηση Προβλημάτων σε Τυπικές Γλώσσες

- Πρόβλημα βελτιστοποίησης $\rightarrow$ πρόβλημα απόφασης με φράγμα $B$.
  - Ελαχιστοποίηση: Υπάρχει εφικτή λύση με κόστος $\leq B$;
  - Μεγιστοποίηση: Υπάρχει εφικτή λύση με κέρδος $\geq B$;
- Πρόβλημα απόφασης $\rightarrow$ τυπική γλώσσα με κωδικοποίηση.
  - Στιγμιότυπο $\rightarrow$ συμβολοσειρά σε αλφάβητο $\Sigma$.
  - Πρόβλημα $\rightarrow$ γλώσσα, υποσύνολο του $\Sigma^*$.
- Πρόβλημα $\Pi$ και κωδικοποίηση $e$: Γλώσσα $\mathcal{L}(\Pi, e)$ αποτελείται από $x \in \Sigma^*$ που προκύπτουν από την $e$-κωδικοποίηση των ΝΑΙ-στιγμιοτύπων του $\Pi$.

$$\mathcal{L}(\Pi, e) = \{e(x) \in \Sigma^* : x \in \Pi\}$$

- Π.χ. προβλήματα προσπελασιμότητας και κύκλου Hamilton.

# Ντετερμινιστικές Μηχανές Turing

- **Ταινίες** διαβάζονται και γράφονται από **κεφαλές**:
  - Πρώτη ταινία: **ταινία εισόδου**.
  - Τελευταία ταινία: **ταινία εξόδου**.
  - Άλλες ταινίες: **ταινίες εργασίας** - μνήμη.
- **Μηχανή Turing** $M = (Q, \Sigma, \delta, q_0, F)$ με $k \geq 1$ ταινίες:
  - $Q$ **σύνολο καταστάσεων**.
  - $\Sigma$ **αλφάβητο εισόδου**. $\Gamma = \Sigma \cup \{\sqcup\}$ **αλφάβητο ταινίας**.
    $\triangleright$ **αρχή ταινίας**.
  - $q_0 \in Q$ **αρχική κατάσταση**.
  - $F \subseteq Q$ **τελικές καταστάσεις**.
  - $\delta : (Q \setminus F) \times \Gamma^k \mapsto Q \times \Gamma^k \times \{L, R, S\}^k$ **συνάρτηση μετάβασης**.
    (κατάσταση $q$, κεφαλές διαβάζουν $\alpha_1 \ldots \alpha_k$) $\to$
    ($q'$, κεφαλές γράφουν $\alpha'_1 \ldots \alpha'_k$, κεφαλές μετακινούνται $L$, $R$, ή $S$)



# Ντετερμινιστικές Μηχανές Turing

- **Ντετερμινισμός**: $M(x)$ εξελίσσεται με **προδιαγεγραμμένο** τρόπο.
- $M$ τερματίζει σε τελική κατάσταση: ΝΑΙ, ΟΧΙ, ΤΕΛΟΣ ή δεν τερματίζει.
  - $M(x) =$ ΝΑΙ, $M$ **αποδέχεται** $x$.
  - $M(x) =$ ΟΧΙ, $M$ **απορρίπτει** $x$.
  - $\mathcal{L}(M) = \{x \in \Sigma^* : M(x) = \text{ΝΑΙ}\}$.
  - $M(x) =$ ΤΕΛΟΣ και έξοδος $y$, $M$ **υπολογίζει** $y = f(x)$.
- **Καθολική Μηχανή Turing**: $U(M; x) = M(x)$.
  Προσομοιώνει τη λειτουργία της μηχανής $M$ με είσοδο $x$.



# Παράδειγμα Μηχανής Turing

| $p \in Q \setminus F$ | $\sigma \in \Gamma$ | $\delta(p, \sigma)$ |
|---|---|---|
| $q_0$ | $\triangleright$ | $(q_0, \triangleright, R)$ |
| $q_0$ | $0$ | $(q_0, 0, R)$ |
| $q_0$ | $1$ | $(q_0, 1, R)$ |
| $q_0$ | $\sqcup$ | $(s, \sqcup, L)$ |
| $s$ | $\triangleright$ | (ΤΕΛΟΣ, $\triangleright, R$) |
| $s$ | $0$ | $(s_0, \sqcup, R)$ |
| $s$ | $1$ | $(s_1, \sqcup, R)$ |
| $s$ | $\sqcup$ | $(s, \sqcup, S)$ |
| $s_0$ | $\triangleright$ | (ΤΕΛΟΣ, $\triangleright, R$) |
| $s_0$ | $0$ | $(q_0, 0, L)$ |
| $s_0$ | $1$ | $(q_0, 0, L)$ |
| $s_0$ | $\sqcup$ | $(q_0, 0, L)$ |
| $s_1$ | $\triangleright$ | (ΤΕΛΟΣ, $\triangleright, R$) |
| $s_1$ | $0$ | $(q_0, 1, L)$ |
| $s_1$ | $1$ | $(q_0, 1, L)$ |
| $s_1$ | $\sqcup$ | $(q_0, 1, L)$ |

- $M = (Q, \Sigma, \delta, q_0, F)$.
- $\Sigma = \{0, 1\}$, $\Gamma = \{0, 1, \sqcup\}$.
- $Q = \{q_0, s, s_0, s_1\} \cup \{\text{ΤΕΛΟΣ}\}$.

$$
\begin{array}{rcl}
(q_0, \underline{\triangleright}1101) & \to & (q_0, \triangleright\underline{1}101) \to \\
(q_0, \triangleright 1\underline{1}01) & \to & (q_0, \triangleright 11\underline{0}1) \to \\
(q_0, \triangleright 110\underline{1}) & \to & (q_0, \triangleright 1101\underline{\sqcup}) \to \\
(s, \triangleright 110\underline{1}\sqcup) & \to & (s_1, \triangleright 110\sqcup\underline{\sqcup}) \to \\
(q_0, \triangleright 110\underline{\sqcup}1) & \to & (s, \triangleright 11\underline{0}\sqcup 1) \to \\
(s_0, \triangleright 11\sqcup\underline{\sqcup}1) & \to & (q_0, \triangleright 11\underline{\sqcup}01) \to \\
(s, \triangleright 1\underline{1}\sqcup 01) & \to & (s_1, \triangleright 1\sqcup\underline{\sqcup}01) \to \\
(q_0, \triangleright 1\underline{\sqcup}101) & \to & (s, \triangleright\underline{1}\sqcup 101) \to \\
(s_1, \triangleright\sqcup\underline{\sqcup}101) & \to & (q_0, \triangleright\underline{\sqcup}1101) \to \\
(s, \underline{\triangleright}\sqcup 1101) & \to & (\text{ΤΕΛΟΣ}, \triangleright\underline{\sqcup}1101)
\end{array}
$$



# Υπολογισιμότητα

- **Ημιαποφασίσιμη** $\mathcal{L}$: $\forall x \in \mathcal{L}$, $M(x) =$ ΝΑΙ.
  $\forall x \notin \mathcal{L}$, $M(x) \neq$ ΝΑΙ (μπορεί να μην τερματίζει).
- **Αποφασίσιμη** $\mathcal{L}$: $\forall x \in \mathcal{L}$, $M(x) =$ ΝΑΙ.
  $\forall x \notin \mathcal{L}$, $M(x) =$ ΟΧΙ.
- **Υπολογίσιμη** $f$: $\forall x \in \Sigma^*$, $f(x) = y \Rightarrow M(x) = y$.
  $f(x)$ δεν ορίζεται $\Rightarrow M(x)$ δεν τερματίζει.
- **Αξίωμα Church - Turing**: Υπολογίσιμο $\Leftrightarrow$ Turing αποφασίσιμο / υπολογίσιμο!
- Να αποδείξετε τις ακόλουθες προτάσεις:
  1. Κάθε αποφασίσιμη γλώσσα είναι και ημιαποφασίσιμη.
  2. $\mathcal{L}$ αποφασίσιμη $\Rightarrow$ συμπλήρωμα $\overline{\mathcal{L}}$ αποφασίσιμο.
  3. $\mathcal{L}$ αποφασίσιμη $\Leftrightarrow$ $\mathcal{L}$ και $\overline{\mathcal{L}}$ ημιαποφασίσιμες.

# Μη-Υπολογισιμότητα

- Μη-αποφασίσιμες γλώσσες (προβλήματα που **δεν λύνονται**) γιατί γλώσσες **πάρα πολλές** και μηχανές Turing **πολλές**.
- **Πρόβλημα Τερματισμού**: $M(x)$ τερματίζει;
- Το πρόβλημα Τερματισμού είναι **μη-αποφασίσιμο**!
- **Απόδειξη**: $H(M;x) = $ ΝΑΙ αν $M(x)$ τερματίζει.
  $H(M;x) = $ ΟΧΙ αν $M(x)$ δεν τερματίζει.
- Ορίζω $J(M)$ τερματίζει $\Leftrightarrow$ $H(M;M) = $ ΟΧΙ $\Leftrightarrow$ $M(M)$ δεν τερματίζει.
- **Άτοπο**: $J(J)$ τερματίζει ανν $J(J)$ δεν τερματίζει!
- Πολλά άλλα προβλήματα δεν λύνονται!!!



# Χρονική Πολυπλοκότητα

- **Χρονική Πολυπλοκότητα** $M \equiv$ αύξουσα $t : \mathbb{N} \mapsto \mathbb{N}$: $\forall x, |x| = n$, $M(x)$ τερματίζει $\leq t(n)$ βήματα.
- **Χρονική Πολυπλοκότητα** $\Pi \equiv$ Πολυπλοκότητα γρηγορότερης $M$ που λύνει $\Pi$.
- $\mathbf{DTIME}[t(n)] \equiv \{\Pi : \Pi$ λύνεται σε χρόνο $\mathrm{O}(t(n))\}$
- **Ιεραρχία Κλάσεων** Χρονικής Πολυπλοκότητας:
  - $\mathbf{DTIME}[t(n)] \subset \mathbf{DTIME}[\omega(t(n) \log t(n))]$
  - $\mathbf{DTIME}[n] \subset \mathbf{DTIME}[n^2] \subset \mathbf{DTIME}[n^3] \subset \cdots$
- **Πολυωνυμικός Χρόνος**: $\mathbf{P} \equiv \cup_{k \geq 0} \mathbf{DTIME}[n^k]$.
- **Εκθετικός Χρόνος**: $\mathbf{EXP} \equiv \cup_{k \geq 0} \mathbf{DTIME}[2^{n^k}]$.

$$\mathbf{P} \subset \mathbf{EXP}$$



# Ευεπίλυτα και Δυσεπίλυτα Προβλήματα

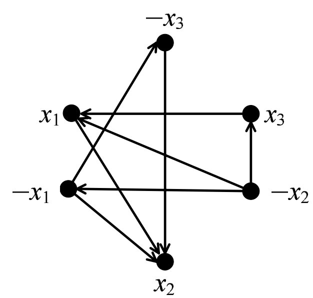
- **Κλάση P**: προβλήματα που λύνονται σε πολυωνυμικό χρόνο.
- **Αξίωμα Cook - Karp**: **P** ταυτίζεται με ευεπίλυτα προβλήματα.
- **Υπέρ**:
  - Δεν εξαρτάται από υπολογιστικό μοντέλο!
  - Συνήθως **μικρά** πολυώνυμα (π.χ. $n, n^2, n^3, \ldots$).
  - Διπλασιασμός υπολογιστικής ισχύος $\Rightarrow$ **σημαντική αύξηση** (π.χ. $2, \sqrt{2}, 2^{1/3}, \ldots$) μεγέθους στιγμιοτύπων.
- **Κατά**:
  - **Ακραίες περιπτώσεις**: Αλγόριθμος με χρόνο $n^{100}$ δεν είναι πρακτικός ενώ αλγόριθμος με χρόνο $2^{n/100}$ είναι!
  - **Γραμμικός Προγραμματισμός**: Simplex εκθετικός στη θεωρία αλλά ταχύτερος στην πράξη!



# 2-Ικανοποιησιμότητα $\in$ P

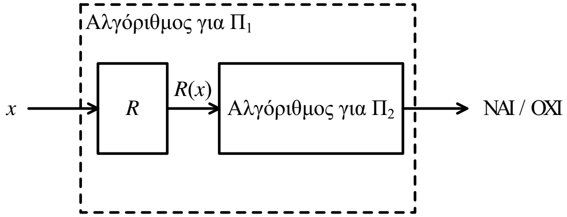
- Λογική πρόταση $\phi$ σε **$k$-Συζευκτική Κανονική Μορφή** ($k$-ΣΚΜ): $\phi = c_1 \wedge \ldots \wedge c_m$, $c_i = \ell_{i_1} \vee \ldots \vee \ell_{i_k}$, $\ell \in \{x_i, \neg x_i\}$.

  Π.χ. $k = 2$: $(x_1 \vee x_2) \wedge (x_1 \vee \neg x_3) \wedge (\neg x_1 \vee x_2) \wedge (x_2 \vee x_3)$
- **$k$-Ικανοποιησιμότητα**: $\phi$ σε $k$-ΣΚΜ ικανοποιήσιμη;
- $\ell_i \vee \ell_j \equiv (\neg \ell_i \to \ell_j) \wedge (\neg \ell_j \to \ell_i)$.
- Γράφημα $G_\phi$ με κορυφές $\{x_1, \ldots, x_n\} \cup \{\neg x_1, \ldots, \neg x_n\}$.
  Όρος $\ell_i \vee \ell_j$: ακμές $(\neg \ell_i, \ell_j)$ και $(\neg \ell_j, \ell_i)$.
- $G_\phi$ συμμετρικό: $(x, x') \Leftrightarrow (\neg x', \neg x)$.
- $\ell_i \to \ell_j$ ψευδής $\Leftrightarrow$ $\ell_i = 1$ και $\ell_j = 0$.
- $\phi$ μη ικανοποιήσιμη ανν μονοπάτι $x - \neg x$ και μονοπάτι $\neg x - x$.



# Αναγωγή και Πληρότητα

- $\Pi_1$ **ανάγεται πολυωνυμικά** σε $\Pi_2$: $\exists$**πολυωνυμικά υπολογίσιμη** συνάρτηση $R : \Sigma^* \mapsto \Sigma^*$, ώστε $\forall x \in \Sigma^*$, $x \in \Pi_1 \Leftrightarrow R(x) \in \Pi_2$ $(\Pi_1 \leq_P \Pi_2)$.
  $R$ ονομάζεται **πολυωνυμική αναγωγή**.
- $\Pi_1 \leq_P \Pi_2$ "δηλώνει" ότι το $\Pi_2$ είναι τουλ. **τόσο δύσκολο** όσο το $\Pi_1$.
- $\mathbf{C}$ κλάση προβλημάτων. $\Pi$ είναι **$\mathbf{C}$-δύσκολο** αν $\forall \Pi' \in \mathbf{C}$ ανάγεται στο $\Pi$.
  Αν $\Pi$ είναι $\mathbf{C}$-δύσκολο και $\Pi \in \mathbf{C} \Rightarrow \Pi$ είναι **$\mathbf{C}$-πλήρες**.
- Τα **$\mathbf{C}$-πλήρη** προβλήματα **"συνοψίζουν"** τη δυσκολία της κλάσης $\mathbf{C}$.
- Κλάση $\mathbf{C}$ είναι **κλειστή** ως προς (πολυωνυμική) αναγωγή αν $\forall \Pi_1, \Pi_2$, $\Pi_1 \leq_P \Pi_2$ και $\Pi_2 \in \mathbf{C} \Rightarrow \Pi_1 \in \mathbf{C}$.






# Μερικές Ασκήσεις

- Πολυωνυμική αναγωγή είναι μεταβατική (σύνθεση αναγωγών).
- $\mathbf{P}$ κλειστό ως προς πολυωνυμική αναγωγή.
- $\Pi_1 \leq_P \Pi_2$ και $\Pi_2 \in \mathbf{P}$ $(\mathbf{NP}) \Rightarrow \Pi_1 \in \mathbf{P}$ $(\mathbf{NP})$.
- (Κλειστές) κλάσεις με **κοινό πλήρες πρόβλημα ταυτίζονται**!
- Κλάσεις $\mathbf{C}, \mathbf{C}'$ κλειστές ως προς αναγωγή.
  Αν έχουν κοινό πλήρες πρόβλημα, $\mathbf{C} = \mathbf{C}'$.