Τι είναι υπολογιστής;

Γιατί μελετάμε τα Αυτόματα και τις Τυπικές Γλώσσες;

- Η Επιστήμη των Υπολογιστών (-ισμών) (Computer Science) έχει τόση σχέση με τους πραγματικούς υπολογιστές όση σχέση έχει η αστρονομία με τα τηλεσκόπια.
- Συμφωνούμε ότι ο υπολογιστής «λύνει προβλήματα». Λύνει όμως όλα τα προβλήματα; Τι μπορεί να «κάνει» και τι δεν μπορεί να «κάνει» ένας υπολογιστής;
- Πως σχετίζονται τα προβλήματα που λύνει ένας υπολογιστής με τα προβλήματα των Μαθηματικών;
- Τι μπορούν να «κάνουν» και τι δεν μπορούν να κάνουν τα Μαθηματικά;

Τι είναι πρόβλημα;

- Υπάρχουν ακέραιοι $x, y, z \geq 1$ και $n > 2$ τέτοιοι ώστε $x^n + y^n = z^n$;
- Γράψτε ένα πρόγραμμα JAVA που όταν η είσοδος είναι ένα ακέραιο πολυώνυμο p (πολλών μεταβλητών), στην έξοδο απαντά αν το πολυώνυμο p έχει ακέραιες ρίζες.

Βασική διαφορά των προβλημάτων:

Το πρώτο ζητά μια απόδειξη ύπαρξης ενώ το δεύτερο ζητά ένα πρόγραμμα JAVA (δηλαδή ένα αλγόριθμο).

Μας απασχολούν προβλήματα όπου το ζητούμενο είναι ένα αλγόριθμος και όχι μια απόδειξη ύπαρξης.

Δύο φημισμένα προβλήματα

Θεώρημα του Fermat (1637) Υπάρχουν ακέραιοι $x, y, z \geq 1$ και $n > 2$ τέτοιοι ώστε $x^n + y^n = z^n$; Η απάντηση είναι αρνητική (Andrew Wiles, 1994).

10ο πρόβλημα του Hilbert (1900) Υπάρχει πρόγραμμα που να λύνει ακέραιες εξισώσεις; Η απάντηση είναι αρνητική (Yuri Matijasevich, 1970)

Τι είναι πρόβλημα;

- Υπάρχουν ακέραιοι $x, y, z \geq 1$ και $n > 2$ τέτοιοι ώστε $x^n + y^n = z^n$;
- Γράψτε ένα πρόγραμμα JAVA που με είσοδο ακεραίους $x, y, z \geq 1$ να γράφει «ναι» στην έξοδο αν υπάρχει ακέραιος $n > 2$, τέτοιος ώστε $x^n + y^n = z^n$.
- Γράψτε ένα πρόγραμμα JAVA που με είσοδο την πρόταση: «Για οποιουσδήποτε ακεραίους $x, y, z \geq 1$ δεν υπάρχει ακέραιος $n > 2$, τέτοιος ώστε $x^n + y^n = z^n$», να γράφει «TRUE» στην έξοδο αν η πρόταση είναι αληθής.
- Γράψτε ένα πρόγραμμα JAVA που με είσοδο μια μαθηματική πρόταση P (σε κατάλληλο συμβολισμό), να γράφει «TRUE» στην έξοδο αν η P είναι αληθής (Entscheidungsproblem, David Hilbert, 1928)

Alan Turing (1912-1954)

Θεμελίωση της Θεωρίας Υπολογισμού

- Τυπικός ορισμός της έννοιας του Υπολογιστή (μηχανή Turing).
- Μελέτησε την έννοια της υπολογισιμότητας (1936).
- Έδειξε πως το Entscheidungsproblem (τα μαθηματικά μπορούν να αυτοματοποιηθούν;) είναι μη επιλύσιμο.

Τι είναι πρόβλημα;

Π1 Δίνεται γράφημα $G = (V, E)$, και κορυφές $s, t \in V$. Να βρεθεί στον G ένα μονοπάτι από το s στο t.

Π2 Δίνεται γράφημα $G = (V, E)$, και κορυφές $s, t \in V$. Υπάρχει στον G μονοπάτι από το s στο t;

Προβλήματα απόφασης

Το Π1 είναι πρόβλημα εύρεσης. Το Π2 είναι πρόβλημα απόφασης, δηλαδή απαντιέται με «ΝΑΙ» ή «ΟΧΙ»

Το Π2 ανάγεται στο πρόβλημα Π1: αν λυθεί το πρόβλημα εύρεσης έχει λυθεί και το πρόβλημα απόφασης. Το αντίστροφο δεν ισχύει πάντα.

Χάριν κομψότητας και ευκολίας ασχολούμαστε κυρίως με προβλήματα απόφασης.

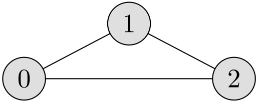
Προβλήματα απόφασης και γλώσσες

- Ορίζουμε ένα πεπερασμένο αλφάβητο, π.χ. $\Sigma = \{0, 1\}$.
- Κάθε πρόβλημα απόφασης μπορεί να κωδικοποιηθεί με μια γλώσσα, δηλαδή ένα υποσύνολο του συνόλου όλων των δυνατών συμβολοσειρών που παράγονται από το αλφάβητο Σ.
  - Αν θέλουμε να χρησιμοποιήσουμε το $\Sigma = \{0, 1\}$ βολεύει ο «εναδικός» συμβολισμός: $0 \to 1$, $1 \to 11$, $2 \to 111$, κτλ
  - Χρησιμοποιούμε τα 0 και 00 σαν διαχωριστικά
- Κάθε πρόβλημα απόφασης είναι ισοδύναμο με το να αποφασιστεί αν η κωδικοποίηση ενός στιγμιοτύπου ανήκει στο σύνολο που περιέχει τις κωδικοποιήσεις όλων των στιγμιοτύπων, τη γλώσσα δηλαδή, του προβλήματος.

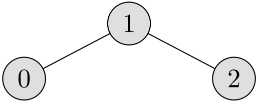
Κωδικοποίηση προβλήματος

Σε γράφο με 3 κόμβους υπάρχει μονοπάτι από τον κόμβο 0 στον κόμβο 2;

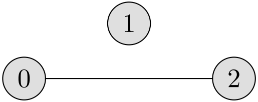
ΝΑΙ



1011011100110101110011101011

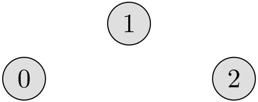


1011001101011100111011

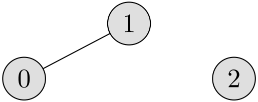


1011100110011101

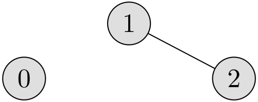
ΟΧΙ



1001100111



101100110100111



10011011100111011

Αλφάβητα και συμβολοσειρές

Ορισμοί

Σύμβολο είναι μια αφηρημένη οντότητα που δεν ορίζεται πιο συγκεκριμένα, π.χ. a, 1, void, class, $\clubsuit$, begin.

Αλφάβητο είναι ένα πεπερασμένο σύνολο συμβόλων Σ, π.χ. $\Sigma = \{\heartsuit, \diamondsuit, 1, \text{a}, \text{class}\}$, $\Sigma = \{0, 1\}$, $\Sigma = \{a, b, \ldots, z\}$

Συμβολοσειρά (string) ενός αλφαβήτου Σ είναι μια πεπερασμένη ακολουθία συμβόλων του Σ, π.χ. $\mathrm{abracadabra}$, $\mathrm{badgirl}$, ένα πρόγραμμα JAVA.

Μήκος συμβολοσειράς ονομάζεται ο αριθμός των συμβόλων που περιέχει, π.χ. $|\mathrm{abracadabra}| = 11$

Κενή ονομάζεται η μοναδική συμβολοσειρά με μήκος 0 και συμβολίζεται με ϵ.

Σ^* συμβολίζεται το σύνολο όλων των συμβολοσειρών που παράγονται από το αλφάβητο Σ, π.χ. αν $\Sigma = \{0, 1\}$ τότε $\Sigma^* = \{\epsilon, 0, 1, 00, 01, 11, 000, 001, 010, 100, 101, \ldots\}$

Πράξεις με συμβολοσειρές

Παράθεση (concatenation) των συμβολοσειρών x και y η συμβολοσειρά xy και συμβολίζεται με xy ή $x \circ y$, π.χ. $011 \circ 1001 = 0111001$.

Επανάληψη της συμβολοσειράς w είναι η συμβολοσειρά w^k που αποτελείται από την παράθεση k αντιγράφων του w, π.χ. $\mathrm{dobedo}^2 = \mathrm{dobedo} \circ \mathrm{dobedo} = \mathrm{dobedodobedo}$.
Επαγωγικός ορισμός:

$$
\begin{array}{lcl}
w^0 & = & \epsilon \\
w^{i+1} & = & w^i \circ w
\end{array}
$$

Αντίστροφη μιας συμβολοσειράς w είναι η συμβολοσειρά w^R και προκύπτει αν διαβάσουμε το w, από το τέλος προς την αρχή, π.χ. $01011^R = 11010$. Επαγωγικός ορισμός:

$$
\begin{array}{lcll}
w^R & = & \epsilon & \text{αν } |w| = 0 \\
w^R & = & au^R & \text{αν } |w| > 1, \text{ τότε } w = u \circ a
\end{array}
$$

Γλώσσες

Αν Σ είναι ένα αλφάβητο, οποιοδήποτε υποσύνολο του Σ^* ονομάζεται γλώσσα του Σ, π.χ. αν $\Sigma = \{0, 1, 2, 3, 4, 5, 6, 7, 8, 9\}$ τότε όλα τα παρακάτω σύνολα είναι γλώσσες του Σ:

- $L_1 = \{23, 044, 99999\}$ (πεπερασμένη γλώσσα),
- $L_2 = \{\epsilon, 1, 11, 111, 1111, \ldots\}$,
- $L_3 = \{w | w$ είναι δεδαδική αναπαράσταση πρώτου αριθμού$\} = \{2, 3, 5, 7, 11, 13, \ldots\}$,
- $L_4 = \{w | w$ είναι δυαδική αναπαράσταση πρώτου αριθμού$\} = \{10, 11, 101, 111, 1011, 1101, \ldots\}$,
- $L_5 = \emptyset$ (κενή γλώσσα),
- $L_6 = \{\epsilon\}$,
- $L_7 = \{w | w$ είναι πρόγραμμα της JAVA που δεν τελειώνει ποτέ$\}$ (το πρόγραμμα δεν έχει input και είναι κωδικοποιημένο στο δυαδικό σύστημα. Όμοια και για πρόγραμμα που κάποτε τελειώνει.)

Πράξεις με γλώσσες

Πράξεις συνόλων Οι γλώσσες είναι σύνολα, συνεπώς αν L_1, L_2 είναι γλώσσες ορίζονται η ένωση $L_1 \cup L_2$, η τομή $L1 \cap L_2$, το συμπλήρωμα $\Sigma^* - L$, κτλ.

Παράθεση δύο γλωσσών L_1, L_2 είναι η γλώσσα $L_1 \circ L_2$ ή $L_1 L_2$ που ορίζεται ως $L_1 \circ L_2 = \{w | w = x \circ y,\ x \in L_1,\ y \in L_2\}$, π.χ. αν $L_1 = \{\text{good}, \text{bad}\}$ και $L_2 = \{\text{boy}, \text{girl}\}$, τότε $L_1 L_2 = \{\text{goodboy}, \text{goodgirl}, \text{badboy}, \text{badgirl}\}$.

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

$$L^* = \{w | w = w_1 \circ w_2 \circ \cdots \circ w_n,\ n \geq 0,\ w_1, \ldots, w_n \in L\}$$

$L^+ = L \circ L^*$ η μικρότερη γλώσσα που περιέχει την L και την L^*.

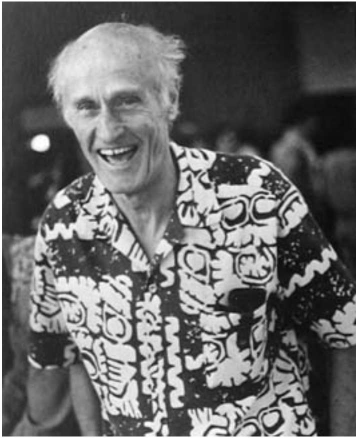
Kleene star

Παραδείγματα

- Αν $L = \{0, 11\}$ τότε
  $L^* = \{\epsilon, 0, 00, 11, 000, 011, 100, 0000, 0011, 0110, 1100, 1111, \ldots\}$,
- αν $L = \{\epsilon\}$, τότε $L^* = \{\epsilon\}$,
- αν $L = \emptyset$, τότε $L^* = \{\epsilon\}$ (άρα $\forall L : \epsilon \in L^*$),
- αν $L = \{01, 1, 100\}$ τότε $110001110011 \in L^*$ γιατί
  $110001110011 = 1 \circ 100 \circ 01 \circ 1 \circ 100 \circ 1 \circ 1$

Steven Cole Kleene

1909 - 1994



Πόσες συμβολοσειρές και πόσες γλώσσες υπάρχουν;

- Ένα αλφάβητο Σ είναι από τον ορισμό του πεπερασμένο.
- Υπάρχουν άπειρες συμβολοσειρές του Σ.
- Υπάρχουν άπειρες γλώσσες του Σ.
- Τα «άπειρα» όμως είναι διαφόρων ειδών.

Τελικά υπάρχουν «περισσότερες» γλώσσες από συμβολοσειρές.

Πεπερασμένα και άπειρα σύνολα

Ισάριθμα ονομάζονται δύο σύνολα A και B αν υπάρχει 1-1 και επί (αμφιμονοσήμαντη) αντισοιχία $f : A \longrightarrow B$.

Πεπερασμένο ονομάζεται ένα σύνολο A αν υπάρχει $n \in \mathbb{N}$ έτσι ώστε τα A και $\{1, 2, \ldots, n\}$ να είναι ισάριθμα. Ο πληθικός αριθμός του A είναι ο αριθμός n ($|A| = n$).

Άπειρο λέγεται ένα σύνολο αν δεν είναι πεπερασμένο, π.χ. το $\mathbb{N}$ (γιατί;).

Μετρήσιμα σύνολα

Ισάριθμα ονομάζονται δύο σύνολα A και B αν υπάρχει 1-1 και επί (αμφιμονοσήμαντη) αντιστοιχία $f : A \longrightarrow B$.

Μετρήσιμα άπειρο λέγεται ένα σύνολο που είναι ισάριθμο με το $\mathbb{N}$.

Μετρήσιμο λέγεται ένα σύνολο αν είναι πεπερασμένο ή μετρήσιμα άπειρο. Ένα σύνολο που δεν είναι μετρήσιμο, λέγεται μη μετρήσιμο

Διαισθητική περιγραφή

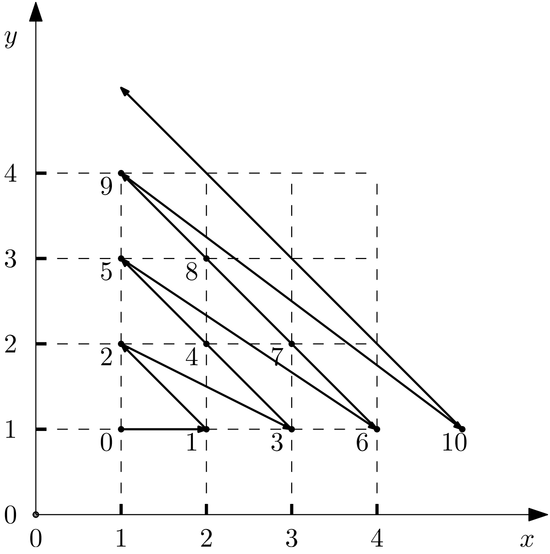
Ένα σύνολο είναι μετρήσιμο αν υπάρχει μηχανιστική διαδικασία που μπορεί να καταγράψει ακολουθιακά τα στοιχεία του (με οποιαδήποτε σειρά). Αν η ακολουθία των στοιχείων τελιώσει κάποτε, το σύνολο είναι πεπερασμένο. Αν η ακολουθία των στοιχείων είναι άπειρη, το σύνολο είναι μετρήσιμα άπειρο.

Μετρήσιμα άπειρα σύνολα

Παραδείγματα

$\mathbb{N}$	Ζυγοί	Περιττοί	$\mathbb{Z}$	$\mathbb{Q}$
0	0	1	0	1/1
1	2	3	1	2/1
2	3	5	-1	1/2
3	6	7	2	3/1
4	8	9	-2	2/2
5	10	11	3	1/3
6	12	13	-3	4/1
7	14	15	4	3/2
8	16	17	-4	2/3
9	18	19	5	1/4
10	20	21	-5	5/1
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$

Μηχανιστική απαρίθμηση των ρητών αριθμών



Το Σ^* είναι μετρήσιμα άπειρο σύνολο

Πρέπει να δώσουμε ένα τρόπο μηχανιστικής απαρίθμησης των στοιχείων του Σ^*:

- Ξεκινάμε διατάσοντας το αλφάβητο: $\Sigma = \{a_1, a_2, \ldots, a_n\}$, θεωρούμε δηλαδή ότι $a_1 < a_2 < \cdots < a_n$.
- Για κάθε $k \geq 0$ όλες οι συμβολοσειρές μήκους k απαριθμούνται πριν από τις συμβολοσειρές με μήκος $k+1$.
- Οι n^k συμβολοσειρές μήκους k απαριθμούνται λεξικογραφικά: αν οι συμβολοσειρές διαφέρουν στο πρώτο σύμβολο, ή αν αρχίζουν με τα ίδια σύμβολα και διαφέρουν στο i-οστό σύμβολο, τότε απαριθμείται πρώτη, η συμβολοσειρά που έχει στη θέση της διαφοράς το μικρότερο στη διάταξη του αλφαβήτου σύμβολο.

Παράδειγμα μηχανιστικής απαρίθμησης του $\{0,1\}^*$

$$
\{0,1\}^* = \{ \begin{array}{r} \epsilon \\ 0, \\ 1, \\ 00, \\ 01, \\ 10, \\ 11, \\ 000, \\ 001, \\ 010, \\ 011, \\ 100, \\ 101, \\ \vdots \end{array} \}
$$

Πεπερασμένη περιγραφή γλωσσών

Οι γλώσσες μπορούν να περιγραφούν με διάφορους τρόπους:

- $L = \{w | w \in \{0,1\}^*$ και περιέχει ίσο αριθμό από 0 και 1$\}$
- Γενικότερα έχουμε περιγραφές της μορφής $L = \{w | \omega$ ικανοποιεί την ιδότητα $T\}$, για κάποια ιδιότητα T.
- Η ιδιότητα T πρέπει να έχει πεπερασμένη περιγραφή Π_T. Έστω Π το σύνολο αυτών των περιγραφών: $\Pi = \{\Pi_T | T$ είναι ιδιότητα$\}$.
  Άρα το Π είναι μια γλώσσα!

Υπάρχουν πολλά σύνολα-είδη περιγραφών. Ξεκινάμε με μια απλή και φυσική περιγραφή.

Κανονικές εκφράσεις (regular expressions)

Περιγράφουν μια γλώσσα χρησιμοποιώντας σύμβολα του Σ, το $\emptyset$, σε συνδυασμό με $\cup$, * και τη βοήθεια παρενθέσεων:

1. Το $\emptyset$ είναι ΚΕ, όπως και κάθε $\sigma \in \Sigma$.
2. Αν a, b είναι ΚΕ, τότε το $(a \cup b)$ είναι ΚΕ.
3. Αν a, b είναι ΚΕ, τότε το (ab) είναι ΚΕ.
4. Αν a είναι ΚΕ, τότε το (a^*) είναι ΚΕ.
5. Τίποτα άλλο δεν είναι ΚΕ.

Προτεραιότητα τελεστών

Η προτεραιότητα *, παράθεση, ένωση επιτρέπει να περιορίσουμε τις παρενθέσεις, π.χ. γράφουμε $b \cup ab^*$ αντί για $(b \cup (a(b^*)))$. Αν δεν δημιουργείται σύγχυση γράφουμε a^* αντί (a^*).

Παραδείγματα κανονικών εκφράσεων

Από το αλφάβητο $\Sigma = \{a, b\}$ μπορούν να προκύψουν οι ΚΕ:

- $((a \cup b)(a^*)) = (a \cup b)a^*$
- $(a \cup (a^*))^*) = (a \cup a^*)^*$
- $((a \cup b))^* aa((a \cup b))^* = (a \cup b)^* aa(a \cup b)^*$
- $(a^*)(b^*)(a^*)(b^*) = a^* b^* a^* b^*$

Γλώσσες και κανονικές εκφράσεις

Σε κάθε ΚΕ αντιτοιχούμε μια γλώσσα ως εξής (αν a είναι μια ΚΕ τότε $\mathcal{L}(a)$ θα συμβολίζει τη γλώσσα):

1. $\mathcal{L}(\emptyset) = \emptyset$, $\mathcal{L}(\sigma) = \{\sigma\}$
2. $\mathcal{L}((a \cup b)) = \mathcal{L}(a) \cup \mathcal{L}(b)$
3. $\mathcal{L}((ab)) = \mathcal{L}(a)\mathcal{L}(b)$
4. $\mathcal{L}(a^*) = (\mathcal{L}(a))^*$

Μια γλώσσα L λέγεται κανονική αν και μόνο αν υπάρχει αντίστοιχη κανονική έκφραση p, δηλαδή $L = \mathcal{L}(p)$.

Γλώσσες κανονικών εκφράσεων

- Ποιά είναι η γλώσσα της κανονικής έκφρασης: $((a \cup b)^*)$

$$
\begin{aligned}
\mathcal{L}(((a \cup b)^*)) &= (\mathcal{L}(a \cup b))^* && \text{(κανόνας 4)} \\
&= (\mathcal{L}(a) \cup \mathcal{L}(b))^* && \text{(κανόνας 3)} \\
&= (\{a\} \cup \{b\})^* && \text{(κανόνας 1)} \\
&= \{a, b\}^*
\end{aligned}
$$

- Ποιά είναι η γλώσσα της κανονικής έκφρασης: $((a \cup ba)^*)$

$$
\begin{aligned}
\mathcal{L}(((a \cup (ba))^*) &= (\mathcal{L}(a \cup (ba)))^* && \text{(κανόνας 4)} \\
&= (\mathcal{L}(a) \cup \mathcal{L}(ba))^* && \text{(κανόνας 2)} \\
&= (\mathcal{L}(a) \cup \mathcal{L}(a)\mathcal{L}(a))^* && \text{(κανόνας 3)} \\
&= (\{a\} \cup \{ba\})^* && \text{(κανόνας 1)} \\
&= \{a, ba\}^*
\end{aligned}
$$

Το δυναμοσύνολο του Σ^* δεν είναι μετρήσιμο

Έστω $\mathcal{P}(\Sigma^*)$ μετρήσιμο, τότε καταγράφουμε όλες τις γλώσσες του Σ:

	w_1	w_2	w_3	w_4	w_5	w_6	w_7	w_8	w_9	$\cdots$
	0	1	00	01	10	11	000	001	010	$\cdots$
$L_1 = \{00, 10, 000\}$	$\boxed{0}$	0	1	0	1	0	1	0	0	$\cdots$
$L_2 = \{01, 11, 010\}$	0	$\boxed{0}$	0	1	0	1	0	0	1	$\cdots$
$L_3 = \{0, 1, 00, 01, 10, 11\}$	1	1	$\boxed{1}$	1	1	1	0	0	0	$\cdots$
$L_4 = \{1\}$	0	1	0	$\boxed{0}$	0	0	0	0	0	$\cdots$
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\ddots$					
Συμπλήρωμα διαγωνίου	1	1	0	1	$\cdots$					
Γλώσσα T	0	1		01	$\cdots$					

Έστω $w_1, w_2, \ldots$ η λεξικογραφική ακολουθία όλων των συμβολοσειρών του Σ^*. Κατασκευάζουμε την γλώσσα που περιέχει τη συμβολοσειρά w_i ανν η L_i δεν περιέχει την w_i:

$$T = \{w_i : w_i \notin L_i,\ i = 0, 1, 2, \ldots\}$$

Η T διαφέρει με όλες τις L_i σε τουλάχιστο ένα στοιχείο. Άρα η T δεν είναι κάποια L_i, συνεπώς το $\mathcal{P}(\Sigma^*)$ δεν είναι μετρήσιμο.

Πεπερασμένες αναπαραστάσεις γλωσσών

- Οι πεπεραμένες γλώσσες έχουν πεπερασμένη αναπαράσταση: την εξαντλητική απαρίθμηση των συμβολοσειρών τους.
- Οι άπειρες γλώσσες έχουν ενδιαφέρον αν έχουν πεπερασμένες αναπαραστάσεις, π.χ. κανονικές εκφράσεις.
- Οι πεπεραμένες αναπαραστάσεις είναι κι αυτές συμβολοσειρές από κάποιο αλφάβητο Σ. Υπάρχουν μετρήσιμα άπειρες συμβολοσειρές (π.χ. $\{0, 1\}^*$) άρα μετρήσιμα άπειρες αναπαραστάσεις γλωσσών.
- Όλες οι γλώσσες όμως είναι μη μετρήσιμα άπειρες. Δηλαδή όλες οι πιθανές γλώσσες είναι περισσότερες από όλες τις διαθέσιμες συμβολοσειρές. Άρα υπάρχουν γλώσσες που δεν περιγράφονται!

Δυνατοί και αδύνατοι υπολογισμοί

Υπάρχουν αδύνατοι υπολογισμοί!

Υπάρχει γλώσσα που ένα πρόγραμμα JAVA δεν μπορεί να τυπώσει όλες τις συμβολοσειρές της, ακόμη κι αν το πρόγραμμα μπορούσε να τρέχει για πάντα!

Κεντρικό πρόβλημα στη Θεωρία Υπολογισμού

Υπάρχει πρόγραμμα JAVA που για οποιαδήποτε γλώσσα που έχει πεπερασμένη περιγραφή, τυπώνει στην έξοδό του όλες τις συμβολοσειρές της γλώσσας;