Alonzo Church (1903-1995)

Alan Turing (1912-1954)

Θεωρία Υπολογισμού

Η Αρχή των Church-Turing

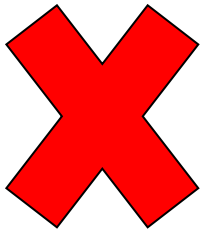
(Νόμος της Μηχανικής Υπολογισιμότητας)

Τι είναι αλγόριθμος;

- Άτυπα:
  - Μία συνταγή
  - Μία διαδικασία
  - Ένα πρόγραμμα στον Υπολογιστή
  - Και τι έγινε αν δεν ξέρεις; Όταν δω έναν τον καταλαβαίνω....

- Ιστορικά
  - Η ιδέα έχει μακρά ιστορία στα Μαθηματικά (ξεκινά με τον αλγόριθμο του Ευκλείδη) αλλά
    - Δεν ορίστηκε επακριβώς μέχρι τον 20ο αιώνα.
    - Αν και υπήρχαν άτυποι ορισμοί που όμως είναι ανεπαρκείς

Principia Mathematica (Whitehead, Russel)

1. Μπορούμε να θέσουμε αξιώματα ώστε κάθε αληθής πρόταση να είναι θεώρημα; Το σύνολο των αξιωμάτων μπορεί να είναι άπειρο αλλά θα είναι διαγνώσιμο (ένας αλγόριθμος που αποφασίζει αν η είσοδος είναι ή όχι αξίωμα). 

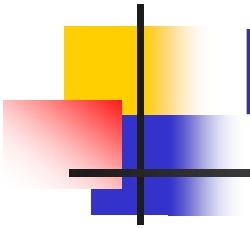
   Godel για Peano αριθμητική όχι όμως και για Presburger αριθμητική

2. Υπάρχει αλγόριθμος που να αποφασίζει δοθέντος ενός συνόλου αξιωμάτων αν ένα θεώρημα προκύπτει από αυτά ή όχι (Entscheidungsproblem); 

   Το πρόβλημα του τερματισμού

Γιατί ΤΜς;

Αν θέλουμε να ερευνήσουμε την έννοια του «αποδοτικού αλγόριθμου» γιατί χρησιμοποιούμε ένα απελπιστικά μη αποδοτικό μοντέλο όπως οι ΤΜς;;;;;;;;;;;!



Παρατηρήσεις

- Πολλά μοντέλα έχουν προταθεί για υπολογισμό γενικού σκοπού.
- Πάραυτα, όλα τα «λογικά» μοντέλα είναι ισοδύναμα με τις ΤΜ.
- Όλες οι «λογικές» γλώσσες προγραμματισμού (π.χ. Java, Pascal, C, Scheme, Mathematica, Maple, Cobol,. . .) είναι ισοδύναμες.
- Η ιδέα του αλγόριθμου πρέπει να είναι ανεξάρτητη του μοντέλου!
- Τελικά δεν ενδιαφερόμαστε για τις ΤΜ αλλά ενδιαφερόμαστε για την κατανόηση του υπολογισμού.

Διάφορα Μοντέλα

Τυπικοί ορισμοί εμφανίστηκαν το 1936:

- λ-άλγεβρα (Alonzo Church)
- Turing machines (Alan Turing)
- Αναδρομικές Συναρτήσεις (Godel και Kleene)
- Απαριθμητές
- Γραμματικές χωρίς περιορισμούς
- Αυτόματα δύο στοιβών
- Μηχανές Τυχαίας Προσπέλασης (RAMs)
- Το παιχνίδι της ζωής του Conway
- Υπολογισμός με DNA
- Κβαντικοί Υπολογιστές …

Αυτοί οι ορισμοί φαίνονται πολύ διαφορετικοί αλλά είναι αποδεδειγμένα ισοδύναμοι.

Το Δόγμα των Church-Turing

Κάθε πρόβλημα που μπορεί να λυθεί από μία μηχανιστική διαδικασία, μπορεί να λυθεί και από μία ΤΜ.

Νόμος της Μηχανικής Υπολογισιμότητας

Διαισθητική έννοια αλγορίθμου = Πρόγραμμα ΤΜ

Τι μπορούν να κάνουν οι πραγματικοί Η/Υ

Παραγωγή Θερμότητας

Σταμάτημα Πόρτας

Παροχή Περιβάλλοντος Διαβίωσης για Ψάρια

Υπολογισμός που μπορούν υποτίθεται να κάνουν αλλά η ΤΜ δεν μπορεί

Παραγωγή Τυχαιότητας

Εξωτικά Μοντέλα;;;;

- Τι γίνεται για τις περιπτώσεις εξωτικών μοντέλων;

- Πάρτε τον επεξεργαστή BVUi-Proton Multiple CORE (στην αγορά θα βγει τα Χριστούγεννα του 2024)

Λειτουργεί σαν μία TM μόνο που:

- Το πρώτο βήμα χρειάζεται 1 δευτερόλεπτο.
- Το δεύτερο βήμα χρειάζεται 1/2 δευτερόλεπτα.
- Το *i*-οστό βήμα απαιτεί 2^{-i} δευτερόλεπτα . . .

Έπειτα από 2 δευτερόλεπτα ο BVUi αναγνωρίζει κάθε αναγνωρίσιμη γλώσσα!

Ερώτηση: Η ύπαρξη του BVUi ακυρώνει το δόγμα των Church-Turing;

Το 10ο Πρόβλημα του Hilbert

Το 1900, ο David Hilbert έδωσε μία διάλεξη στο Διεθνές Συνέδριο των Μαθηματικών στο Παρίσι.

- Παρουσίασε 23 βασικά μαθηματικά προβλήματα
- Προκλήσεις για τον 20ο αιώνα
- Το 10ο πρόβλημα αφορούσε αλγόριθμους

- Ας ξεκινήσουμε με κάποιο υπόβαθρο για το συγκεκριμένο πρόβλημα . . .

Πολυώνυμα

- Ένας όρος είναι το γινόμενο μεταβλητών και σταθερών, π.χ. $6x^3yz^2$.
- Ένα πολυώνυμο είναι άθροισμα όρων π.χ.

$$6x^3yz^2 + 3xy^2 - x^3 - 10 .$$

- Η ρίζα ενός πολυωνύμου είναι μία ανάθεση τιμών ώστε το πολυώνυμο να είναι ίσο με μηδέν.
- Για παράδειγμα:, $x = 5$, $y = 3$, και $z = 0$ είναι μία ρίζα του παραπάνω πολυωνύμου.
- Ενδιαφερόμαστε για ακέραιες ρίζες.
- Μερικά πολυώνυμα έχουν ακέραιες ρίζες και μερικά όχι (π.χ. $x^2 - 2$).

Το 10ο Πρόβλημα του Hilbert

- Το πρόβλημα: Σχεδίασε έναν αλγόριθμο που να ελέγχει αν ένα πολυώνυμο έχει ακέραια ρίζα.

- Στην πραγματικότητα αυτό που είπε είναι:

  "*να σχεδιάσεις μία διαδικασία σύμφωνα με την οποία μπορεί να βρεθεί σε πεπερασμένο αριθμό πράξεων*".

Προσέξτε ότι ο Hilbert αναζητά την σχεδίαση ενός αλγόριθμου τον οποίο θεωρεί ότι υπάρχει και ότι απλά χρειάζεται κάποιος να ψάξει να τον βρει.

Το 10ο Πρόβλημα του Hilbert

- Γνωρίζουμε πια ότι δεν υπάρχει τέτοιος αλγόριθμος.
- Οι μαθηματικοί του 1900 δεν μπορούσαν να αποδείξουν κάτι τέτοιο μιας και δεν υπήρχε ένας τυπικός ορισμός της έννοιας του αλγόριθμου.
- Οι διαισθητικοί ορισμοί είναι εντάξει για την κατασκευή αλγορίθμων
- Για την απόδειξη ανυπαρξίας αλγόριθμου όμως απαιτείται ένας τυπικός ορισμός.

Τυπικός ορισμός του προβλήματος

Θεωρείστε τη γλώσσα:

$D = \{p \mid p \text{ είναι ένα πολυώνυμο με ακέραια ρίζα}\}$

Το 10ο πρόβλημα του Hilbert ρωτά αν αυτή η γλώσσα είναι διαγνώσιμη.

Γνωρίζουμε πια ότι δεν είναι διαγνώσιμη αλλά είναι αναγνωρίσιμη (1970, Matijasevic – 23 χρονών)!

Πολυώνυμα

Θεωρείστε την πιο απλή γλώσσα:

$$D_1 = \{p \mid p \text{ είναι ένα πολυώνυμο στο } x \text{ με ακέραια ρίζα}\}$$

Μία TM που αναγνωρίζει την D_1.

Σε είσοδο p,

- Αποτίμησε το p με το x να τίθεται διαδοχικά στις τιμές 0, 1,−1, 2,−2, ...
- Αν το p παίρνει την τιμή μηδέν τότε αποδεχόμαστε.

Πολυώνυμο

$D_1 = \{p \mid p$ *είναι ένα πολυώνυμο στο* x *με ακέραια ρίζα*$\}$

Προσέξτε ότι

- Αν το p έχει ακέραια ρίζα, η μηχανή αποδέχεται.
- Αν όχι, η M_1 εγκλωβίζεται.
- *Η* M_1 αναγνωρίζει αλλά δεν διαγιγνώσκει.

Πολυώνυμο

Στην πραγματικότητα η D_1 είναι διαγνώσιμη.

Μπορεί να αποδειχτεί ότι όλες οι ρίζες του p[x] βρίσκονται στο διάστημα

$$(-|kc_{max}/c_1|, |kc_{max}/c_1|)$$

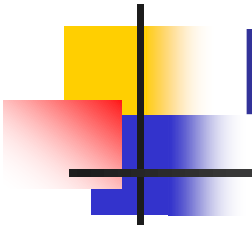
όπου k είναι το πλήθος των όρων, c_{max} είναι η μέγιστη πολλαπλασιαστική σταθερά και c_1 είναι η σταθερά του όρου υψηλότερης τάξης.

Από το θεώρημα του Matijasevic, τέτοια όρια στις ρίζες δεν μπορούν να υπολογιστούν για πολυώνυμα πολλαπλών μεταβλητών.

ΚΩΔΙΚΟΠΟΙΗΣΗ

Κωδικοποίηση

- Η είσοδος σε μία ΤΜ είναι μία λέξη συμβόλων.
- Όμως εμείς θέλουμε αλγόριθμους που λειτουργούν σε γραφήματα, πίνακες, πολυώνυμα κοκ.
- Πρέπει να κωδικοποιήσουμε αυτά τα αντικείμενα.
- Συνήθως μπορούμε να το κάνουμε με διάφορους τρόπους.

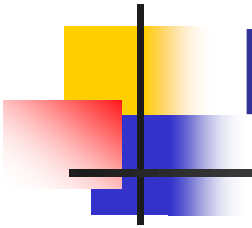


Κωδικοποίηση

- Θεωρείστε λέξεις που αναπαριστούν ακατεύθυντα γραφήματα.
- Ένα γράφημα είναι συνδεδεμένο αν κάθε κόμβος μπορεί να προσπελάσει οποιονδήποτε άλλο κόμβο του γραφήματος χρησιμοποιώντας κάποιες ακμές.
- Ορίζουμε τη γλώσσα:

A = {G | G είναι ένα συνδεδεμένο ακατεύθυντο γράφημα}

Η γλώσσα πρέπει να είναι διαγνώσιμη.



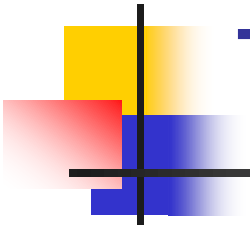
Περιγραφή

Περιγραφή υψηλού επιπέδου για την TM που διαγνώσκει:

$$A = \{G \mid G \text{ είναι ένα συνδεδεμένο ακατεύθυντο γράφημα}\}$$

Σε είσοδο G, που κωδικοποιεί το γράφημα G:

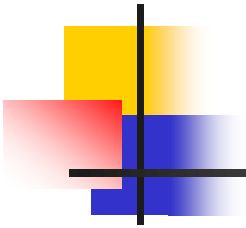
1. Επέλεξε τον πρώτο κόμβο του G και σημάδεψέ τον.
2. Επανέλαβε έως ότου κανένας καινούργιος κόμβος δεν σημαδεύεται:
   - Για κάθε μη-σημαδεμένο κόμβο του G, σημάδεψέ τον αν έχει ακμή της οποίας ο άλλος κόμβος είναι σημαδεμένος
3. Διάτρεξε τους κόμβους του G για έλεγχο αν είναι όλοι σημαδεμένοι. Αν είναι αποδεχόμαστε αλλιώς απορρίπτουμε.



ΤΜ και Υπολογιστές

- Οποιοδήποτε σύγχρονο υπολογιστικό σύστημα μπορεί να μοντελοποιηθεί από μία ΤΜ.
- Αν υπάρχει ένας αλγόριθμος για κάποιο πρόβλημα τότε η ΤΜ μπορεί να το λύσει.
- Προσέξτε ότι δεν ενδιαφερόμαστε για το χρόνο υπολογισμού αλλά για το αν μπορούμε να υπολογίσουμε

ΚΑΘΟΛΙΚΕΣ ΤΜ



Καθολικές ΤΜς

Θα ορίσουμε την Καθολική ΤΜ U.

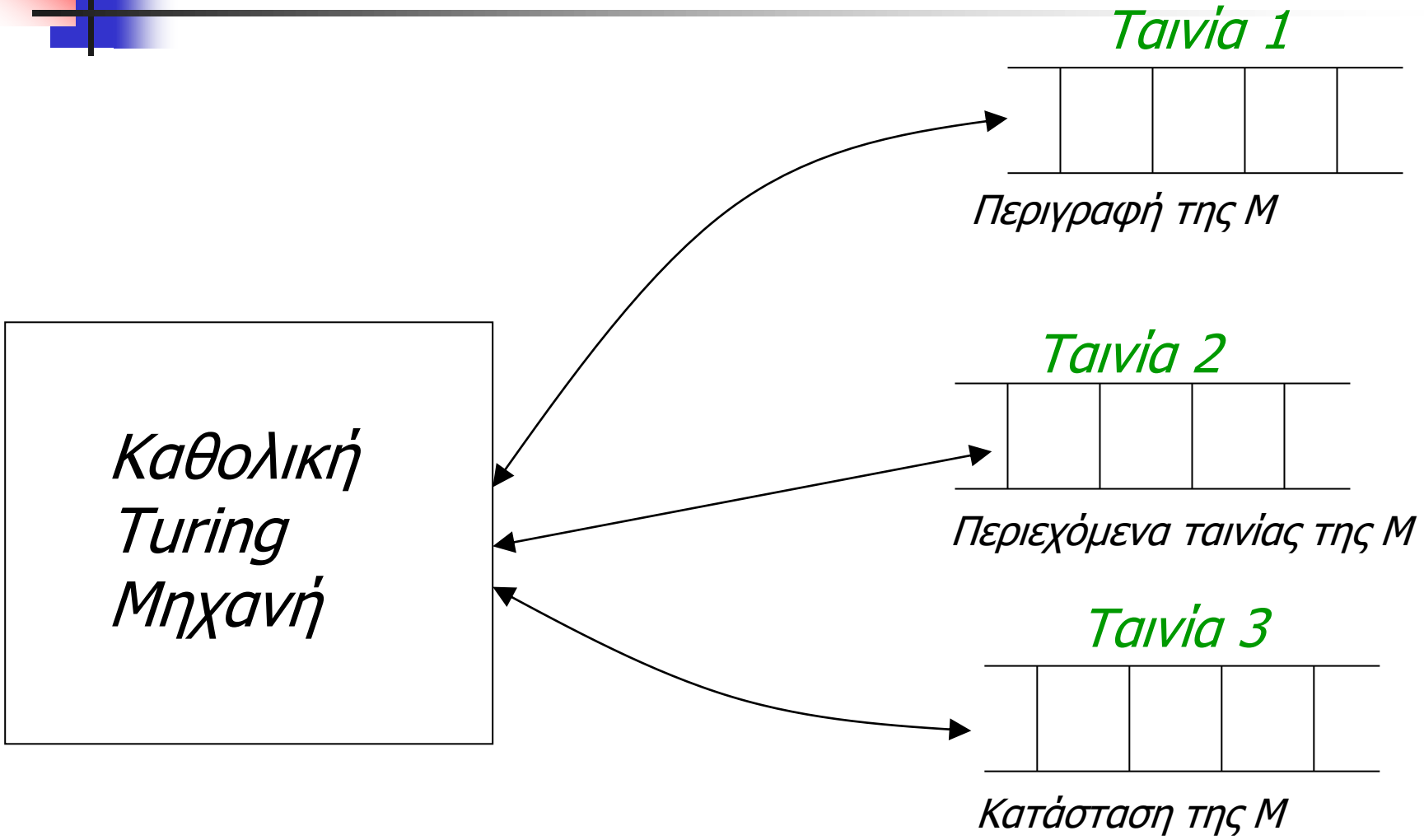
Σε είσοδο $\langle M,w \rangle$, όπου M είναι μία ΤΜ και w μία λέξη

1. Ελέγχει αν η $\langle M,w \rangle$ είναι μία σωστή κωδικοποίηση της ΤΜ ακολουθούμενης από μία λέξη στο Σ^*.

2. Εξομοιώνει την M στο w *(Πώς;;;;)*

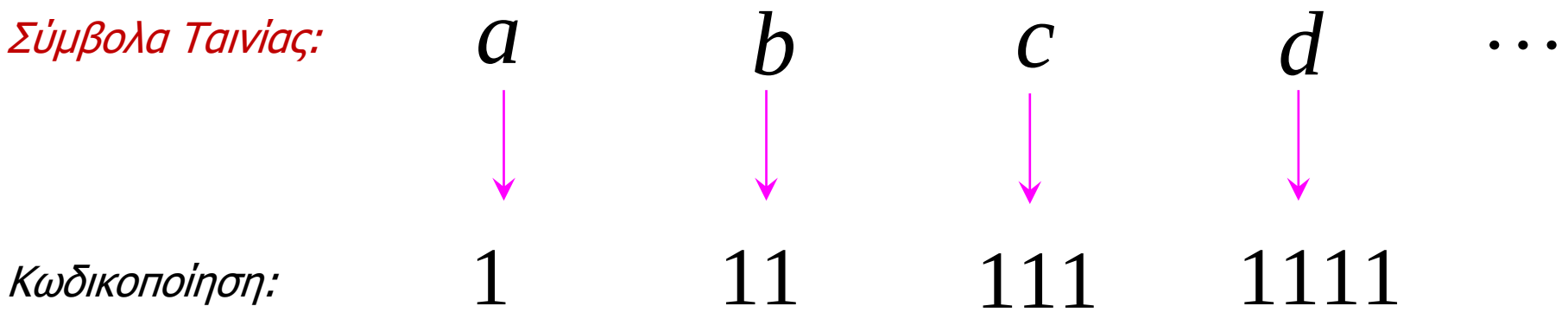
3. Αν η M σε είσοδο w αποδέχεται, τότε και η U αποδέχεται και αν η M απορρίψει τότε και η U απορρίπτει. Άρα όταν η M δεν τερματίζει το ίδιο συμβαίνει και με την U.

Καθολικές ΤΜ



Περιγραφή ΤΜ

Μία ακολουθία από σύμβολα.



Περιγραφή ΤΜ

Καταστάσεις: q_1 q_2 q_3 q_4 $\cdots$

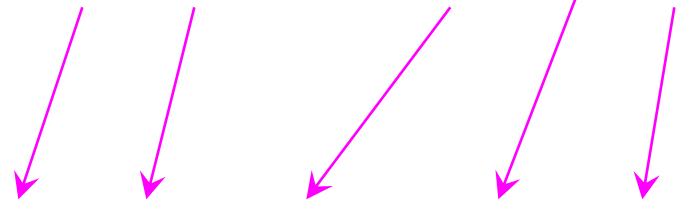
Κωδικοποίηση: 1 11 111 1111

Κίνηση: L R

Κωδικοποίηση: 1 11

Κωδικοποίηση Μεταβάσεων

Μετάβαση : $\delta(q_1, a) = (q_2, b, L)$

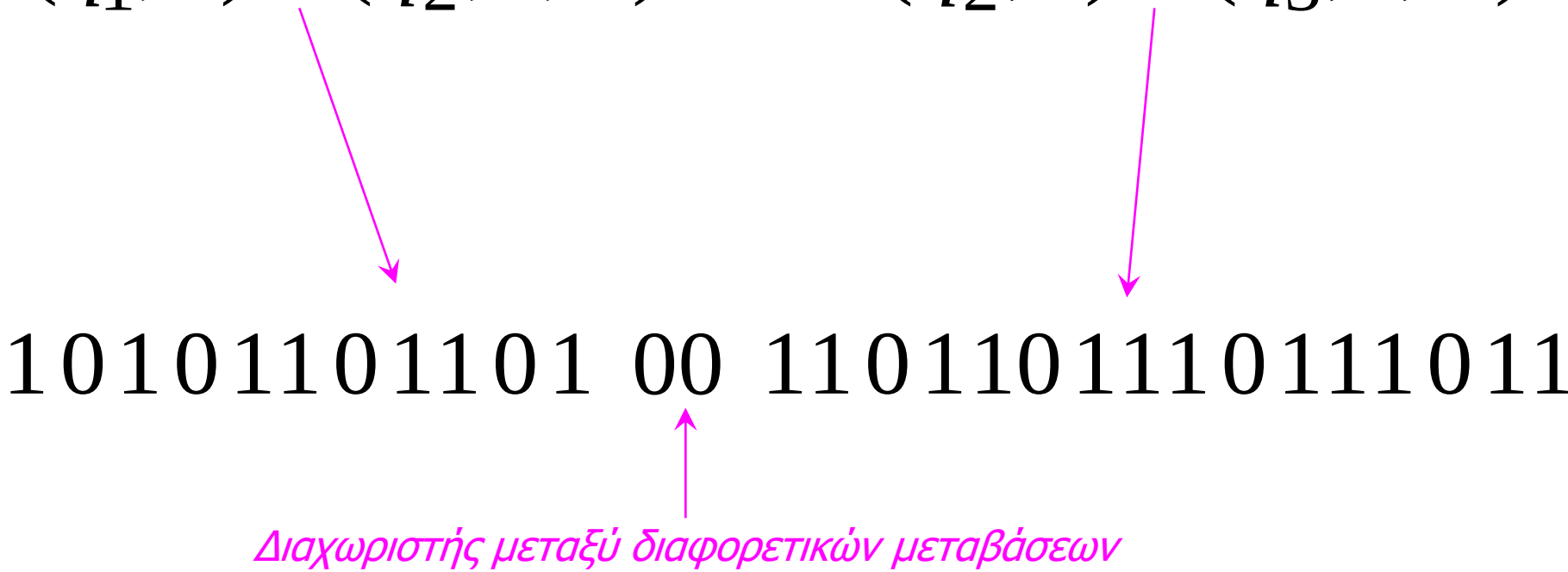


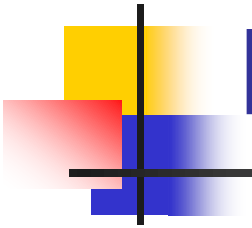
Κωδικοποίηση: 1 0 1 0 1 1 0 1 1 0 1

διαχωριστής

Κωδικοποίηση Συνάρτησης Μετάβασης

$\delta(q_1, a) = (q_2, b, L)$ $\delta(q_2, b) = (q_3, c, R)$





Γλώσσα των ΤΜ

- Στην ταινία 1 τοποθετούμε την κωδικοποίηση της ΤΜ ως ακολουθία 0 και 1.

- Όλες αυτές οι λέξεις δημιουργούν μία γλώσσα: τη γλώσσα που αποτελείται από κωδικοποιήσεις ΤΜ

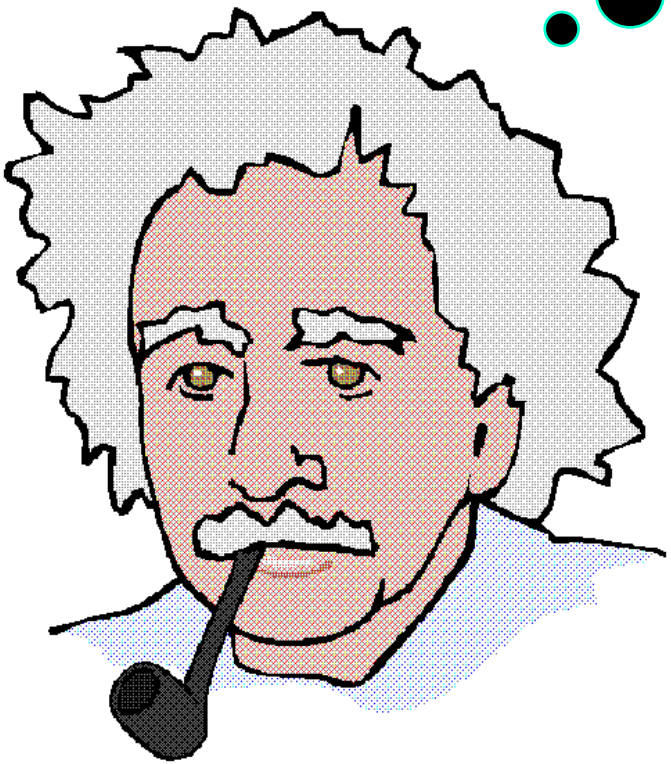
L = { 010100101, (TM 1)

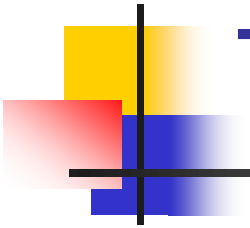
00100100101111, (TM 2)

111010011110010101, ...

..... }

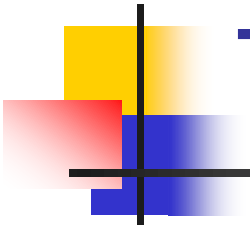
Φιλοσοφικές Σκέψεις





Το Αξίωμα των Church-Turing

Κάθε υπολογιστική μέθοδος που μπορεί να σχεδιαστεί και να εκτελεστεί από το ανθρώπινο μυαλό, μπορεί να εκτελεστεί και σε έναν Υπολογιστή



Το Αξίωμα των Church-Turing

- Η θέση αυτή δεν είναι Θεώρημα (δεν έχει αποδειχθεί) αλλά είναι μία εικασία σχετικά με το σύμπαν στο οποίο ζούμε.

- Η άποψή σας σχετικά με αυτό το αξίωμα μπορεί να επηρεαστεί από τις θρησκευτικές, επιστημονικές και φιλοσοφικές σας αντιλήψεις.

Εμπειρική Διαίσθηση

- Ποτέ κανένας μέχρι τώρα δεν έχει δώσει ένα αντιπαράδειγμα:

  Κανένας δεν έχει σχεδιάσει ένα παράδειγμα όπου οι άνθρωποι μπορούν να υπολογίσουν με έναν λογικό και καλώς ορισμένο τρόπο αλλά να μην μπορεί να προγραμματισθεί σε έναν Η/Υ. Άρα φαίνεται ότι το αξίωμα είναι σωστό. (Διαίσθηση;;;;;;;)

Μηχανική Διαίσθηση

- Ο εγκέφαλος είναι μία μηχανή. Τα επιμέρους κομμάτια αυτής της μηχανής ακολουθούν κάποιους φυσικούς νόμους. Επί της αρχής, ο ανθρώπινος εγκέφαλος μπορεί να εξομοιωθεί από έναν υπολογιστή. Άρα, κάθε σκέψη αυτού του εγκεφάλου μπορεί να υπολογισθεί από έναν Η/Υ. Άρα το αξίωμα είναι σωστό.

Πνευματική Διαίσθηση

- Ο νους του ανθρώπου αποτελείται από το φυσικό και το πνευματικό του κομμάτι (ψυχή). Η ψυχή, λόγω της φύσης της, δεν υπόκειται σε φυσικούς νόμους. Άρα, οι σκέψεις του νου δεν μπορούν να εξομοιωθούν και να αναχθούν σε απλά επιμέρους κομμάτια και κανόνες. Άρα το αξίωμα είναι ψευδές.

Κβαντική Διαίσθηση

- Ο εγκέφαλος είναι μία μηχανή που δεν ακολουθεί τους κλασσικούς νόμους. Οι νόμοι της κβαντικής μηχανικής περιγράφουν την λειτουργία του και επομένως, δεν μπορεί να αναχθεί σε απλά κομμάτια. Άρα, υπάρχουν εμπόδια στο να εξομοιωθεί από έναν ψηφιακό υπολογιστή. Το αξίωμα είναι ψευδές. Αν όμως επιτρέψουμε κβαντικούς υπολογιστές τότε το αξίωμα είναι αληθές.