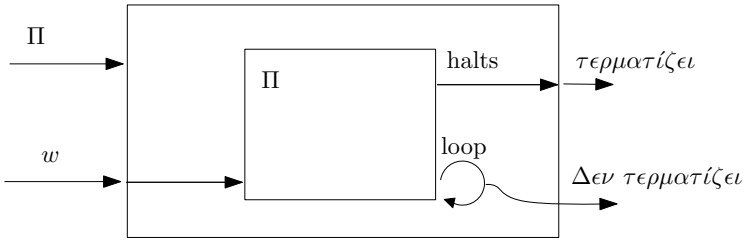
ΥΠΟΛΟΓΙΣΙΜΟΤΗΤΑ (Computability)

"Είναι μια πρόταση **υπολογιστή (computable)** ή όχι;"
"Είναι μια πρόταση **αποκρίσιμη (decidable)** ή όχι;"

Παράδειγμα: Το "halting problem" (HP)

Μπορούμε να φτιάξουμε ένα πρόγραμμα το οποίο όταν του δίνουμε ως είσοδο ένα πρόγραμμα Π και μια συμβολοσειρά w μας λέει εάν "το Π τερματίζει με είσοδο w";



Θεώρημα: Το halting problem είναι μη αποκρίσιμο

Παράδειγμα: Post-correspondence problem (PCP)

- Ντόμινο $\left[\frac{\text{string\_1}}{\text{string\_2}}\right]$ $\left[\frac{\text{a}}{\text{ab}}\right]$
- Συλλογή από Ντόμινο:

$$\left\{\left[\frac{\text{b}}{\text{ca}}\right], \left[\frac{\text{a}}{\text{ab}}\right], \left[\frac{\text{ca}}{\text{a}}\right], \left[\frac{\text{abc}}{\text{c}}\right]\right\}$$

- Ακολουθία-ταίριασμα (match): λίστα από ντόμινο τέτοια ώστε ο "αριθμητής"ισούται με τον "παρανομαστή"

$$\left|\begin{array}{cccc} \text{a} & \text{b} & & \\ \text{a} & \text{b} & & \end{array}\right| \quad \left|\begin{array}{cccc} \text{c} & \text{a} & & \\ \text{c} & \text{a} & & \end{array}\right| \quad \left|\begin{array}{cc} \text{a} & \\ \text{a} & \end{array}\right| \quad \left|\begin{array}{ccc} \text{a} & \text{b} & \text{c} \\ \text{a} & \text{b} & \text{c} \end{array}\right|$$

- **PCP:** Έχει μια δεδομένη συλλογή από ντόμινο μια ακολουθία-ταίριασμα;

Θεώρημα: Το Post-correspondence problem είναι μη αποκρίσιμο.

Αντικείμενο της θεωρίας πολυπλοκότητας:

- "Εάν δοθεί ότι μια συνάρτηση f είναι υπολογιστή (computable), ποιο το κόστος ή τα αγαθά (resources) που χρειάζονται για τον υπολογισμό της;"
- Μια συνάρτηση λέγεται υπολογιστή (computable) εάν υπάρχει ένα πρόγραμμα που υπολογίζει την τιμή της για κάθε όρισμα.

Θεώρημα: Υπάρχουν μη υπολογιστές συναρτήσεις.

Απόδειξη: Από ΛΗΜΜΑ-1 & ΛΗΜΜΑ-2

ΛΗΜΜΑ-1: Υπάρχουν άπειρα μεν, αλλά αριθμήσιμα διαφορετικά προγράμματα τα οποία (χρησιμοποιώντας κωδικοποίηση) μπορεί να απαριθμηθούν μηχανιστικά (effectively enumerable)

Απόδειξη:

- $\Sigma = \{Q_1, Q_2, \ldots, Q_m\}$ το αλφάβητο της γλώσσας προγραμματισμού
- Σ_n: σύνολο συμβολοσειρών του Σ μήκους n
- $\Sigma^* \equiv \bigcup_{n=0}^{\infty} \Sigma_n$
- Κάθε πρόγραμμα είναι στοιχείο του Σ^*

$$\begin{aligned} \Sigma_0 &= \{\epsilon\} \\ \Sigma_1 &= \{Q_1, Q_2, \ldots, Q_m\} \\ \Sigma_2 &= \{Q_1Q_1, Q_1Q_2, \ldots, Q_1Q_m, \ldots, Q_mQ_m\} \\ &\vdots \end{aligned}$$

$\Longrightarrow$ Με χρήση compiler για έλεγχο ορθότητας μπορούμε να κατασκευάσουμε μηχανιστική αρίθμιση των συντακτικά ορθών προγραμμάτων □

ΛΗΜΜΑ-2: Υπάρχουν μη-αριθμήσιμες άπειρες διαφορετικές συναρτήσεις

Απόδειξη:

- Έστω το σύνολο των *ολικών συναρτήσεων* $\Phi : \mathcal{N} \to \mathcal{N}$
  (ολικές συναρτήσεις: ορίζονται για κάθε $x \in \mathcal{N}$)
- Έστω $\Phi_0, \Phi_1, \ldots$ μια αρίθμησή τους
- Ορίζουμε την f : $f(x) = \Phi_x(x) + 1$

  $f(0) = \Phi_0(0) + 1$

  $f(1) = \Phi_1(1) + 1$

  $\vdots$

  $\to$ Η f είναι *ολική*

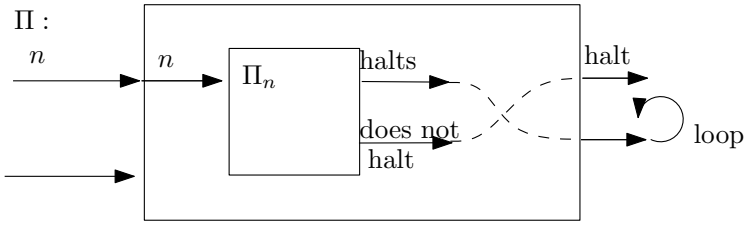
  $\to$ Έστω ότι η f έχει δείκτη y στην αρίθμηση, δηλαδή $f \equiv \Phi_y$
- $f \equiv \Phi_y \Longrightarrow \Phi_y(y) = \Phi_y(y) + 1$ $\to\leftarrow$ΑΤΟΠΟ

□

Θεώρημα: Το halting problem δεν είναι αποκρίσιμο

Απόδειξη: Έστω ότι είναι αποκρίσιμο

- Έστω $\Pi_0, \Pi_1, \ldots$ μια μηχανιστική αρίθμηση όλων των προγραμμάτων
- Φτιάχνω πρόγραμμα Π:



- Έστω $\Pi \equiv \Pi_i$ (στην αρίθμηση)

  $\Pi(n)$ σταματάει $\Longleftrightarrow \Pi_n(n)$ δεν σταματάει

  $\Pi_i(n)$ σταματάει $\Longleftrightarrow \Pi_n(n)$ δεν σταματάει

  $\Pi_i(i)$ σταματάει $\Longleftrightarrow \Pi_i(i)$ δεν σταματάει $\to\leftarrow$ΑΤΟΠΟ

□

Ορισμός Ένα σύνολο S λέγεται *αποκρίσιμο (decidable)* εάν και μόνο εάν υπάρχει αλγόριθμος που σταματάει (ή μια υπολογιστική μηχανή που δίνει έξοδο "ναι"για κάθε είσοδο $a \in S$ και "όχι" για κάθε είσοδο $a \notin S$)

Ορισμός Ένα σύνολο S λέγεται *καταγράψιμο (enumerable, listable, effectivelly generatable)* εάν και μόνο εάν υπάρχει μια γεννήτρια διαδικασία ή μια μηχανή που καταγράφει το S

Ιδιότητες

S αποκρίσιμο $\Longrightarrow \bar{S}$ αποκρίσιμο

S αποκρίσιμο $\Longrightarrow S$ καταγράψιμο

$S, \bar{(}S)$ καταγράψιμο $\Longrightarrow S$ αποκρίσιμο

S καταγράψιμο με γνησίως αύξουσα διάταξη $\Longrightarrow S$ αποκρίσιμο

Θέση (thesis) των Church-Turing

Όλα τα γνωστά και "άγνωστα"μοντέλα της έννοιας *υπολογιστός* είναι μηχανιστικά ισοδύναμα

Δοθέντος ενός αλγορίθμου σε ένα μοντέλο για μια συνάρτηση f τότε μηχανιστικά μπορούμε να κατασκευάσουμε αλγόριθμο σε άλλο μοντέλο για την ίδια συνάρτηση

{Η έννοια του αλγορίθμου} είναι ισοδύναμη με {την έννοια του αλγορίθμου για μηχανές **Turing**}

Τι είναι ένας "Αλγόριθμος"

- Συνταγή
- Διαδικασία
- Ένα πρόγραμμα υπολογιστή
- ...

- δεν είχε οριστεί με ακρίβεια πριν τον 20 αιώνα

Το $10^{\text{ο}}$ πρόβλημα του Hilbert

- Το 1900, ο David Hilbert σε μια διάσημη διάλεξή του
  - όρισε 23 μαθηματικά προβλήματα
  - προκλήσεις για τον $20^{\text{ο}}$ αιώνα
  - το $10^{\text{ο}}$ πρόβλημα αφορούσε τους αλγορίθμους

Πολυώνυμα:

όρος	γινόμενο μεταβλητών και σταθερών συντελεστών
	$6x^3yz^2$
πολυώνυμο	άθροισμα από όρους
	$6x^3yz^2 + 3xy^2 - x^3 - 10$
ρίζα	τιμές των μεταβλητών έτσι ώστε η
	τιμή του πολυωνύμου είναι ίση με μηδέν (0)
	$x = 5$, $y = 3$, $z = 0$

Το πρόβλημα: Να βρεθεί ένας αλγόριθμος που ελέγχει εάν ένα πολυώνυμο έχει ακέραιες ρίζες

Ο Hilbert είπε

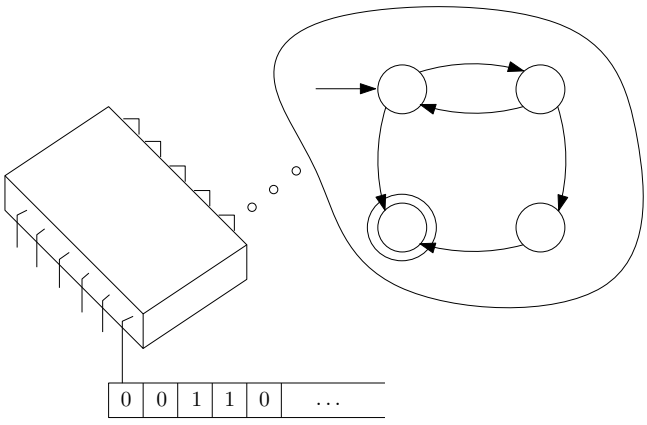
> "... to device a process according to which it can be determined by a finite number of operations."

Παρατηρήστε ότι

- Ο Hilbert ζητάει να κατασκευαστεί ένας αλγόριθμος
- υποθέτει ότι ο αλγόριθμος υπάρχει

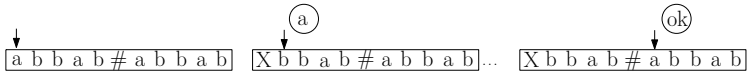
$\Longrightarrow$ Σήμερα γνωρίζουμε ότι δεν υπάρχει τέτοιος αλγόριθμος
[Yuri Matijasevic, 1970]

ΜΗΧΑΝΕΣ TURING

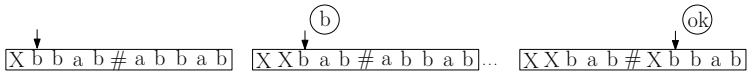


- Χρησιμοποιεί άπειρη ταινία ως μνήμη
- Η ταινία αρχικά περιέχει την συμβολοσειρά-είσοδο ακολουθούμενη από κενό
- Η ταινία χρησιμοποιείται ως "πρόχειρη μνήμη"
- Η μηχανή έχει *accept* και *reject* καταστάσεις
- μπορεί να "τρέχει"για πάντα

- Μια ΤΜ που ελέγχει εάν η είσοδος ανήκει στη γλώσσα $\left\{w\#w | w \in \{a,b\}^+\right\}$



  - Η κεφαλή βρίσκεται στο αριστερότερο σύμβολο
  - θυμάται το σύμβολο (σε κατάσταση) και το διαγράφει X
  - ψάχνει στα δεξιά: **REJECT** εάν βρει ΚΕΝΟ πριν από #
  - όταν βρει #, πηγαίνει μια θέση δεξιά
  - εάν τα σύμβολα δεν ταιριάζουν **REJECT**



  - διαγράφει το σύμβολο, γράφει X
  - ψάχνει αριστερά, προσπερνά το #, πηγαίνει έως το X
  - πηγαίνει μια θέση δεξιά
  - θυμάται το σύμβολο (σε κατάσταση), το αντικαθιστά με X
  - ψάχνει δεξιά, μετά το # έως το τελευταίο X ...

  - πηγαίνει μια θέση δεξιά
  - εάν βρει a ή b **REJECT**
  - εάν βρει ΚΕΝΟ **ACCEPT**

ΟΡΙΣΜΟΣ (ΜΗΧΑΝΗ TURING)

- Βασίζεται στην **συνάρτηση μετάβασης (transition function)**

$$\delta : Q \times \Gamma \to Q \times \Gamma \times \{L, R\}$$

$\delta(q, a) = (r, b, L)$ σημαίνει:

  - Η ΤΜ στην κατάσταση q με την κεφαλή να διαβάζει a
  - αντικαθιστά το a με b
  - η κατάσταση γίνεται r
  - η κεφαλή κινείται *αριστερά*

Μια Μηχανή Turing (ΤΜ) είναι μια 7-άδα $(Q, \Sigma, \Gamma, \delta, q_0, q_a, q_r)$ όπου:

  - Q ένα πεπερασμένο σύνολο καταστάσεων (states)
  - Σ το *αλφάβητο-εισόδου*, δεν περιέχει το *κενό* σύμβολο $\textvisiblespace$
  - Γ το *αλφάβητο-ταινίας*, περιέχει το Σ και το $\textvisiblespace$
  - $\delta : Q \times \Gamma \to Q \times \Gamma \times \{L, R\}$ η συνάρτηση μετάβασης
  - $q_0 \in Q$ η *αρχική κατάσταση*
  - $q_a \in Q$ η *accept*-*κατάσταση*
  - $q_r \in Q$ η *reject*-*κατάσταση*

- Μια TM M **αποδέχεται (accepts)** την είσοδο x ανν σταματά σε κατάσταση αποδοχής q_a για το x

  – Η γλώσσα L_M που αναγνωρίζει/αποδέχεται η TM M είναι:

$$L_M = \{x \in \Sigma^* : M \text{ αποδέχεται το } x\}$$

- Μια TM M **αποφασίζει (decides)** την είσοδο x ανν σταματά σε μια κατάσταση αποδοχής q_a ή απόρριψης q_r

  – Η γλώσσα L_M για την οποία αποκρίνεται/αποφασίζει (decides) η TM M είναι

$$L_M = \{x \in \Sigma^* : M \text{ αποφασίζει το } x\}$$

ΠΑΡΑΛΛΑΓΕΣ της Μηχανής Turing

1. TM όπου η κεφαλή μπορεί να μείνει ακίνητη

$$\delta : Q \times \Gamma \to Q \times \Gamma \times \{L, R, S\}$$

- Είναι ισοδύναμη με την TM

  Αντικατάσταση της $\delta(q, a) = (q', b, S)$ με

  $\delta(q, a) = (q_1, b, R)$ q_1 είναι νέα κατάσταση

  $\delta(q_1, x) = (q', x, L)$ για κάθε $x \in \Gamma$

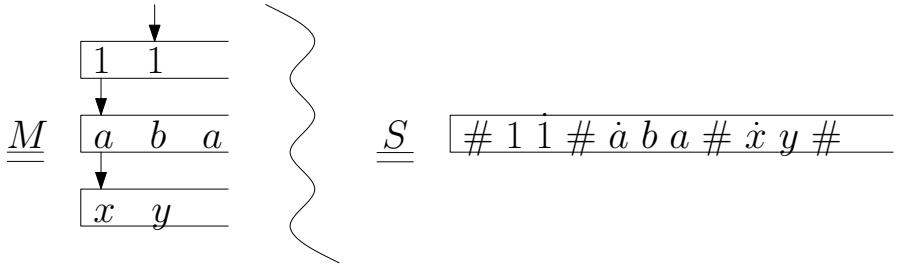
Η έννοια της **προσομοίωσης** είναι σημαντική

2. TM με πολλαπλές ταινίες

- κάθε ταινία έχει την δική της κεφαλή
- αρχικά η είσοδος είναι στην ταινία #1 και οι υπόλοιπες ταινίες είναι κενές

συνάρτηση μετάβασης: $Q \times \Gamma^k \to Q \times \Gamma^k \times \{L, R\}^k$
$\delta(q_i, a_1, a_2, \ldots, a_k) = (q_j, b_1, b_2, \ldots, b_k, L, R, \ldots, L)$

ΠΡΟΣΟΜΟΙΩΣΗ



- Η M έχει k ταινίες
- Προσομοιώνει τις k ταινίες σε μια ταινία όπου αποθηκεύει τα δεδομένα τους
- Η S "σημειώνει"την θέση των κεφαλών

Με είσοδο $w = w_1 w_2 \ldots w_n$, η S:

- στην ταινία: $\#\dot{w}_1 w_2 \ldots w_n \# \dot{\_} \# \dot{\_} \# \ldots \# \dot{\_} \#$
- ψάχνει από το 1^o # έως και το $k+1$ # για να διαβάσει τα σύμβολα στις ταινίες
- διασχίζει και γράφει τα νέα σύμβολα
- εάν χρειαστεί η S μετατοπίζει προς τα δεξιά τα περιεχόμενά της για να δημιουργήσει κενό χώρο

3. Μη Ντετερμινιστική TM (NTM)

Συνάρτηση μετάβασης $\delta : Q \times \Gamma \to \mathcal{P}(Q \times \Gamma \times \{L, R\})$

- ο "υπολογισμός" είναι ένα δένδρο
- **"accepts"** εάν ένα οποιοδήποτε κλαδί accepts

Βασική ιδέα προσομοίωσης:

- Μετατροπή της NTM M σε ντετερμινιστική TM D
  - Η D δοκιμάζει όλα τα κλαδιά υπολογισμού
  - Εάν η D εντοπίσει accept-κατάσταση, **accept**
  - Εάν όλα τα κλαδιά οδηγούν σε reject-κατάσταση, **reject**
  - Εάν όλα τα κλαδιά οδηγούν σε reject-κατάσταση ή loop, **loop**
- Ο υπολογισμός της M είναι ένα δένδρο
  - κάθε κλαδί αντιστοιχεί σε έναν μη-ντετερμινιστικό υπολογισμό
  - κάθε κόμβος είναι ένα **"configuration"** της M
  - η ρίζα είναι το αρχικό configuration
  - χρήση BFS

ΠΡΟΣΟΜΟΙΩΣΗ...

D έχει 3 ταινίες

ταινία εισόδου 1 1 0 1 ␣␣␣...

ταινία προσομοίωσης a b 1 0 1 c d ...

ταινία-διευθύνσεων 1 2 3 1 4 ...

- Η ταινία-εισόδου δεν μεταβάλεται ποτέ
- Η ταινία-προσομοίωσης είναι αντίγραφο της ταινίας της M
- Η ταινία-διευθύνσεων αποθηκεύει την "θέση" της M στο "δένδρο υπολογισμού" της M

Η ταινία-διευθύνσεων...

- Κάθε κόμβος του δένδρου έχει το πολύ b παιδιά
- Κάθε κόμβος έχεια μια **διεύθυνση** που είναι μία λέξη με αλφάβητο το $\Sigma_b = \{1, 2, \ldots, b\}$
- Για να πάμε στον κόμβο 1-2-3
  - ξεκινά από την ρίζα
  - ακολουθεί 1^{η} επιλογή
  - ακολουθεί 2^{η} επιλογή
  - ακολουθεί 3^{η} επιλογή
- Αγνοεί επιλογές που δεν έχουν νόημα

Η προσομοίωση...

- Αρχικά η ταινία-εισόδου περιέχει το w, οι άλλες είναι άδειες
- Αντιγράφει την ταινία εισόδου στην ταινία προσομοίωσης
- Στην ταινία-προσομοίωσης, προσομοιώνει την M με είσοδο w, για ένα κλαδί υπολογισμού. Για κάθε βήμα συμβουλεύεται την ταινία διευθύνσεων.
  - **ACCEPT** εάν φτάσει σε "κατάσταση"αποδοχής
  - Πηγαίνει στο επόμενο βήμα εάν
    - Τα σύμβολα της ταινίας - διευθύνσεων εξαντλήθηκαν
    - μη έγκυρη επιλογή
    - έχει φτάσει σε "κατάσταση"reject
- Αντικαθιστά τη λέξη στην ταινία-διευθύνσεων με την λεξικογραφικά επόμενή της